

**LEVERAGING RELATIONAL REPRESENTATIONS  
FOR  
CAUSAL DISCOVERY**

A Dissertation Presented

by

MATTHEW J. H. RATTIGAN

Submitted to the Graduate School of the  
University of Massachusetts Amherst in partial fulfillment  
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2012

Computer Science

© Copyright by Matthew J. H. Rattigan 2012

All Rights Reserved

**LEVERAGING RELATIONAL REPRESENTATIONS  
FOR  
CAUSAL DISCOVERY**

A Dissertation Presented

by

MATTHEW J. H. RATTIGAN

Approved as to style and content by:

---

David Jensen, Chair

---

Andrew Barto, Member

---

Andrea Foulkes, Member

---

Erik Learned-Miller, Member

---

Foster Provost, Member

---

Lori Clarke, Department Chair  
Computer Science

*To all my teachers, be they professors, Italian carpenters,  
or the love of my life. Ancora imparo.*

## ACKNOWLEDGMENTS

I would like to thank my advisor, David Jensen, for the support he has afforded me over the years. I can still remember my first meeting in his office, where he expressed the importance of human communication and understanding as a key to being a successful researcher. While I am indebted to David for his instruction in the craft of scientific inquiry, above all else I admire his humanity—David’s willingness to see his students as people rather than just researchers is what truly sets him apart from his peers. I am also grateful for the time and effort of the members of my committee: Andrew Barto, Andrea Foulkes, Erik Learned-Miller, and Foster Provost. I look forward to having your input on my work and career in the years to come.

The students and staff of the Knowledge Discovery Laboratory have made my time at UMass wonderful. Over the years, I’ve had the privilege of working alongside some extraordinary graduate students: David Arbour, James Atwood, Hannah Blau, Andrew Fast, Lisa Friedland, Amanda Gentzel, Michael Hay, Phil Kirlin, Marc Maier, Katerina Marazopoulou, Jennifer Neville, Huseyin Oktay, Özgür Şimşek, and Brian Taylor. Among them, I would especially like to thank my co-authors: Jen provided me with an example of how to be an excellent graduate student, and Marc’s genuine kindness and sense of humor somehow made the long days and nights before a paper deadline enjoyable. I would like to thank Agustin Schapira and Matthew Cornell for providing the infrastructure that made much of my research possible, Dan Corkill for his helpful input, and Deb Bergeron for helping me to navigate the sometimes daunting bureaucracy of a state university. Lastly, I would like to thank Cindy Loiselle for all her help with my writing over the years.

Several others have contributed to my work in less direct (but no less crucial) ways. My parents, Brian and Joanne, have been incredibly supportive and encouraging over the years (“Enough is enough already!”), and were, along with my brother Brian, my first mentors. My grandparents, Michelina and Italo, showed me the value of hard work. The members of the extended Hale clan, who have only known me as a perpetual student, have supported me and provided me with a warm home away from home on the other side of the world. I’d also like to thank Arnold Almquist, who was “that teacher”, instilling in me a mathematical curiosity and willingness to challenge myself; Dr. Dan Dougherty, for introducing me to computer science; Dr. Harry Penn, for listening; and Dr. Brendan Kiernan, who offered me my first piece of graduate school advice.

In addition, I’d like to extend my gratitude to all the friends who have nourished my soul over my years in graduate school. There are too many of you to list here, but all of you are appreciated. Over coffee and beer you have listened to me, advised me, and motivated me. You have taken me in when I needed shelter and fed me when I was hungry, both literally and figuratively.

Finally, and most importantly, I’d like to thank my friend, love, and partner Marilyce. Your affection, understanding, and patience know no bounds and have made all of this possible. With you by my side, I look forward to writing the next chapter.

# ABSTRACT

## LEVERAGING RELATIONAL REPRESENTATIONS FOR CAUSAL DISCOVERY

SEPTEMBER 2012

MATTHEW J. H. RATTIGAN

B.A., WESLEYAN UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor David Jensen

This thesis represents a synthesis of relational learning and causal discovery, two subjects at the frontier of machine learning research. Relational learning investigates algorithms for constructing statistical models of data drawn from multiple types of interrelated entities, and causal discovery investigates algorithms for constructing causal models from observational data. My work demonstrates that there exists a natural, methodological synergy between these two areas of study, and that despite the sometimes onerous nature of each, their combination (perhaps counterintuitively) can provide advances in the state of the art for both.

Traditionally, propositional (or “flat”) data representations have dominated the statistical sciences. These representations assume that data consist of independent and identically distributed (iid) entities which can be represented by a single data table. More recently, data scientists have increasingly focused on “relational” data

sets that consist of interrelated, heterogeneous entities. However, relational learning and causal discovery are rarely combined. Relational representations are wholly absent from the literature where causality is discussed explicitly. Instead, the literature on causality that uses the framework of graphical models assumes that data are independent and identically distributed.

This unexplored topical intersection represents an opportunity for advancement — by combining relational learning with causal reasoning, we can provide insight into the challenges found in each subject area. By adopting a causal viewpoint, we can clarify the mechanisms that produce previously identified pathologies in relational learning. Analogously, we can utilize relational data to establish and strengthen causal claims in ways that are impossible using only propositional representations.



# TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS .....	v
ABSTRACT .....	vii
LIST OF TABLES .....	xii
LIST OF FIGURES .....	xiii
CHAPTER	
INTRODUCTION .....	1
1. BACKGROUND .....	5
1.1 Graphical models and causality .....	8
1.1.1 d-separation .....	11
1.1.2 d-separation with determinism .....	12
1.1.3 Ground graphs .....	14
1.1.4 Learning algorithms .....	15
1.2 Relational data representations .....	20
1.2.1 Relational semantics .....	21
1.2.2 Data graphs and entity-relationship diagrams .....	24
1.3 Graphical models for relational data .....	26
1.3.1 DAPER models .....	26
1.3.2 Ground graphs .....	28
1.3.3 Regression-based relational modeling techniques .....	30
1.4 Validity .....	31

<b>2. PROPOSITIONALIZATION</b> .....	<b>32</b>
2.1 Propositionalization defined .....	33
2.1.1 Algebraic approach to propositionalization .....	34
2.1.2 Graphical approach to propositionalization .....	37
2.2 Propositionalization and instance dependence bias .....	37
2.2.1 Graphical analysis of iid .....	39
2.3 Aggregation and degree disparity .....	48
2.4 Discussion .....	50
<b>3. HYPOTHESIS TESTS FOR REPLICATED DATA WITH   INSTANCE DEPENDENCE</b> .....	<b>53</b>
3.1 Statistical consequences of instance dependence .....	53
3.2 Understanding instance dependence bias with graphical models .....	56
3.3 Link sampling .....	60
3.4 Novel hypothesis tests with <i>ID</i> variables .....	63
3.5 Empirical results .....	66
3.6 Conclusion .....	68
<b>4. HYPOTHESIS TESTS FOR AGGREGATED DATA WITH   DEGREE DISPARITY</b> .....	<b>70</b>
4.1 Statistical bias in aggregated domains .....	71
4.2 Graphical models for aggregated data .....	72
4.3 Empirical results .....	75
4.3.1 NHL scoring .....	76
4.3.2 Stack Overflow .....	80
4.4 Conclusion .....	81
<b>5. RELATIONAL BLOCKING</b> .....	<b>84</b>
5.1 Example .....	85
5.2 Background .....	88
5.3 Blocking versus conditioning .....	91
5.3.1 Common causes .....	93
5.3.2 Common Effects .....	94
5.3.3 Latent Confounders .....	96
5.3.4 Power .....	97

5.4	Blocking in Practice .....	97
5.4.1	Wikipedia .....	98
5.4.2	Stack Overflow .....	100
5.5	Blocking in many-to-many data .....	102
5.6	Less is more: Sampling for power .....	103
5.7	Discussion .....	107
5.8	Conclusions .....	113
<b>6.</b>	<b>AUTOMATED IDENTIFICATION OF RELATIONAL MARKOV EQUIVALENCE CLASSES.....</b>	<b>115</b>
6.1	Discussion .....	122
<b>7.</b>	<b>CONCLUSIONS AND THE FUTURE.....</b>	<b>123</b>
	<b>BIBLIOGRAPHY .....</b>	<b>126</b>

## LIST OF TABLES

Table	Page
1.1 Summary of different graph representations. . . . .	7
3.1 Results of marginal ( <i>badge, score</i> ) and conditional ( <i>ID, score   badge</i> ) hypothesis tests for Stack Overflow (replication). Boldface indicates statistical significance, while italicized text highlights cases where associations are judged not causal due to a lack of significance in the conditional test. . . . .	69
4.1 Z-scores for Stack Overflow hypothesis tests (aggregation). Significant values are in <b>bold</b> . Italicized values indicate that conditioning on degree has changed the result of the test. . . . .	83
5.1 Details of Wikipedia data . . . . .	99
5.2 Details of Stack Overflow data . . . . .	100
5.3 Contingency tables and chi-square results for each grade group in the classroom example. While only one group (“F”) shows a significant dependence between <i>W</i> and <i>S</i> , the data set as a whole is significant when tested with a CMH statistic ( $CMH = 11.20, p = 0.0008$ ). . . . .	112

## LIST OF FIGURES

Figure	Page
1.1 Relationships between different graphical representations. Adding explicit relational information to a traditional, table-based data description yields the more expressive ER diagram, which can then be instantiated (“rolled out”) into a relational data graph. Adding causal semantics to each of the three yields a Bayesian network, a DAPER model and a ground graph, respectively. . . . .	6
1.2 A simple Bayesian network. . . . .	10
1.3 Causal DAGs represent conditional independence relationships with <i>d-separation</i> . Marginally, $\{A\}$ and $\{F\}$ are d-connected by the collider-free undirected path $A \rightarrow D \rightarrow F$ , as are $\{G\}$ and $\{H\}$ with path $G \leftarrow E \rightarrow H$ . $\{B\}$ and $\{C\}$ are marginally d-separated, since the only path connecting them ( $B \rightarrow E \leftarrow C$ ) contains a collider ( $E$ ). If we condition on $\{E\}$ , then $\{G\}$ and $\{H\}$ become d-separated, but $\{B\}$ and $\{C\}$ become d-connected. . . . .	13
1.4 Graphical model with a deterministic edge ( $A \rightarrow D$ ). When deterministic relationships are present, a d-connecting path is blocked by non-colliders who are determined by variables in the conditioning set. Here, $F \perp\!\!\!\perp G \mid A$ . This relationship differs from the system depicted in Figure 1.3, which had no determinism. . . . .	13
1.5 Given a set of instances (a), graphical models representing their conditional independence relationships (b) can be rolled out to produce a ground graph representing dependence relationships across worlds (c). . . . .	15
1.6 Edge orientation rules for constructing a DAG from a causal skeleton. Dashed lines denote the explicit nonexistence of an edge. . . . .	17

1.7	Application of edge orientation rules to the graph from Figure 1.3. Starting with a skeleton and a set of separating sets (table), edges are oriented through successive application of the rules shown in Figure 1.6. Since the direction of the edge between $B$ and $D$ is not identifiable through conditional independence, it cannot be oriented. ....	19
1.8	The Stack Overflow data comprises <i>users</i> , <i>questions</i> , and <i>answers</i> , and are connected by three types of relationships (user-question, user-answer, question-answer). ....	22
1.9	Data graph for a small relational data set with two entity types. Entities of type $A$ ( $a_1, \dots, a_4$ ) have attributes $X$ and $Z$ , while entities of type $B$ ( $b_1, \dots, b_10$ ) have attribute $Y$ . ....	23
1.10	Entity-relationship diagram (a) and three possible relational data sets that it describes (b-d). ....	25
1.11	DAPER models combine the the structural representation of entity-relationship diagrams with the probabilistic dependence representation of plate-structured graphical models. (a) Example of a DAPER model for bipartite data where $X$ and $Y$ are associated, as presented by Heckerman et al.[35] (b) A simplified version of the same model, with attributes drawn inside the boxes representing the entities they are associated with. ....	27
1.12	The probabilistic ground graph (c) can be constructed by applying a DAPER model (b) to an appropriate relational data graph (a). ....	29
2.1	Relational database tables illustrating propositionalization operations (a). Replication (b) is the result of a three-way <code>INNER JOIN</code> of $T_A$ , $T_B$ and $T_{link}$ . Aggregation (c) is the result of a <code>GROUP BY</code> applied to the same join used in conjunction with an aggregation function $f()$ . Common functions include <code>SUM</code> , <code>MAX</code> , <code>MIN</code> , and <code>AVG</code> . ....	33
2.2	Propositionalization can be represented by subgraph sampling from the data graph. (a) Data graph representation matching the tables shown in Figure 2.1a. Propositionalization by replication is performed by drawing connected subgraphs (with replacement) from the data graph. For aggregation (b), the data graph is augmented with aggregated attributes, and subgraphs are sampled from the augmented graph. ....	36

2.3	ER diagram (left) and data graph for academic publishing example. Each journal entity is connected to one or more paper entities, which are in turn related to several author entities. Journals have attributes for format ( $F$ ), prestige ( $P$ ); papers have attributes for length ( $L$ ) and citation count ( $C$ ); authors have a single attribute that measures their happiness ( $H$ ).....	41
2.4	Graphical depiction of propositionalization for the academic publishing domain. Propositionalizing journal-paper using replication produces overlapping instances due to shared journal entities (a). By aggregating instead of replicating, the overlapping problem is avoided (b); however, aggregated subgraphs will overlap when data are related in a many-to-many manner (c). Single-entity instances (papers) will never overlap (d). ....	42
2.5	DAPER models and ground graphs for the propositionalized instance subgraphs shown in Figure 2.4c. (a) Under this generative model, instances are not independent due to the existence of d-connecting paths between the attributes of different instances in the ground graph. (b) Here, there are no such paths, so the set of paper instances will be independent. Note that if we were to condition on author happiness $H$ , these instances would become dependent as well due to the activation of paths through author entities. ....	46
2.6	Equivalence of Condition 1 (non-overlapping subgraphs) and Condition 2 (d-separation) outlined above. By propagating attributes to related entities, we can avoid sampling overlapping subgraphs while still capturing the same attribute information. However, since propagated attributes necessarily introduce deterministic dependence with the source attribute, there will exist a d-connecting path between child entities. ....	48
2.7	Degree disparity occurs when some attribute is associated with the number of relations for that entity. (Left) ER diagram for journals (with attributes format $F$ and publication rate $R$ ) and papers (with attributes length $L$ and citations $C$ ). (Center) data graph showing propositionalization subgraphs and constructed attribute using MAX aggregator (shaded). In this example, journal issues with yearly publication rates ( $R = Y$ ) tend to have more papers than those that publish monthly ( $R = M$ ). (Right) The data table produced by propositionalization through aggregation indicates an apparent relationship between $R$ and $\text{MAX}(C)$ . While there may be a direct dependence between $R$ and $C$ , the association may be due to the degree disparity with $R$ combining with the sensitivity of the MAX aggregator to degree. ....	50

2.8	Differing data sets (a, b) may produce the same data table (c) when propositionalized. Naive learning systems that do not account for relations will mistakenly process the data in the same way, even though only one of the data sets comprises a valid iid sample when propositionalized. . . . .	51
3.1	When applied to relational data, conventional statistical tests such as $\chi^2$ implicitly assume one-to-one relationships (and therefore iid instances) such as those found in (a). When data are related in a one-to-many manner (b), the conventional reference distribution may be inappropriate due to the violation of the independence assumption. . . . .	54
3.2	Values of the chi-square statistic are biased for data with instance dependence. (Left) The empirical distribution has much higher variance than the theoretical $\chi^2$ with one degree of freedom. (Right) The Type I error rate greatly exceeds the expectation based on alpha; the bias becomes more severe for higher levels of dependence. . . . .	56
3.3	DAPER models (left), ground graphs (center), and distributions of the chi-square statistic (right) for different relational data sets. In scenarios (a) and (e), values of both $X$ and $Y$ are non-independent (and therefore not d-separated in the ground graph), resulting in a distribution of the test statistic that does not match the chi-square reference distribution. . . . .	58
3.4	Three possible DAPER models for one-to-many data. In Model $H_0$ , variables $X$ and $Y$ are independent. In Model $H_1$ , $X$ influences $Y$ , while in $H_2$ , $Y$ is independent of $X$ but related to a latent variable $Z$ . Data generated by $H_1$ and $H_2$ will exhibit instance dependence when propositionalized with replication. . . . .	59
3.5	Link sampling modifies the data graph such that no entity has degree greater than one. While this reduces the data available to a hypothesis test, the transformed graph will produce an iid sample when propositionalized. . . . .	61
3.6	When generated from independent subgraphs through link sampling, the distribution of $\chi^2$ closely matches the theoretical $X^2$ distribution in cases where observed dependence is due to instance dependence bias (a). In cases where there is a direct dependence (b), the distribution is unaffected. . . . .	62



3.7	Propositionalized versions of generative models for non-iid data. The plate structure is included here for clarity only, and is not part of the graphical model. . . . .	63
3.8	Empirical chi-square distributions for $ID, Y$ . (Left) Data generated under Model $H_1$ is indistinguishable from data generated by Model $H_2$ as both models create autocorrelation among $Y$ values (captured here as an association between $ID$ and $Y$ ). (Right) The effect of conditioning on $X$ , allowing clear discrimination between models. . . . .	65
3.9	ER diagram describing the Stack Overflow data set. Users post questions as well as answer questions from others. Users are awarded badges, while both questions and answers are given scores based on the number of up and down votes. . . . .	66
3.10	Alternative models that explain the marginal dependence between Stack Overflow badge awards on user and scores of their answers. In (a), badge awards have a direct influence on answer score; in (b), the perceived dependence is due to instance dependence bias brought on by a hidden factor $H$ . The two models can be differentiated by performing a conditional hypothesis test on $ID$ and $Score$ conditioned on $Badge$ . . . . .	67
4.1	Distribution of z-score values for AVG, MAX, MIN, and SUM in a relational data set with moderate degree disparity. The sampling distributions indicate dependence even in the absence of dependence in the original data. Here, even though $X$ and $Y$ are marginally independent, $X$ appears significantly correlated with aggregations of $Y$ . . . . .	71
4.2	Type I error as a function of alpha for MAX (left) and SUM (right) aggregations under degree disparity. The value of $X$ varies linearly with degree (parameterized by coefficient $\beta_{deg}$ ). At the $\alpha = 0.01$ level, the Type I error rates are 15% and 70% for MAX and SUM, respectively. . . . .	72
4.3	DAPER models for one-to-many data with degree disparity. Model $H_0$ represents the null hypothesis that $X$ is marginally independent of both $Y$ (and therefore, any aggregation $f(Y)$ ) and $deg$ , while Model $H_1$ specifies that $X$ has influence over $Y$ . Model $H_2$ represents data that exhibit degree disparity. Here, the value of $X$ varies with the number of B entities that each A connects to, but is independent of the $Y$ values on those entities. . . . .	73

4.4	<p>DAPER models corresponding to the DAPER models in Figure 4.3, along with ground graphs and Z-score distributions for <math>X</math>, <math>f(Y)</math>. The effects of degree disparity are represented by the dependence of the <i>deg</i> on <math>X</math>, coupled with an aggregation <math>f(Y)</math> that is sensitive to degree (and therefore dependent on <i>deg</i>). In both cases there are d-connecting paths in the ground graph connecting <math>X</math> with <math>f(Y)</math>. . . . .</p>	74
4.5	<p>Models and Z-score distributions for data sets with (a) no direct dependence between <math>X</math> and degree, and (b) no dependence between the aggregator used and degree. In both cases, the lack of a d-connecting path between <math>X</math> and <math>f(Y)</math> eliminates the degree disparity effects shown in Figure 4.4b. . . . .</p>	75
4.6	<p>Conditioning on degree removes bias for statistics based on data with degree disparity, allowing differentiation from data containing actual association between <math>X</math> and <math>Y</math>. (a) Empirical distribution of Z-score after conditioning on degree for data generated under Model <math>H_1</math>. (b) Conditional Z-score distribution for data from model <math>H_2</math>. . . . .</p>	75
4.7	<p>(a) ER diagram depicting the hockey scoring domain. Each Game entity is related in a one-to-many manner to Goal A and Goal B entities, which have a timestamp attribute. (b) DAPER diagram for the hockey scoring domain. The Game entities carry additional aggregated attributes capturing the minimum goal time for both Team A and Team B goals, which together determine the value of the <i>First</i> attribute. The dashed edge, whose existence we wish to evaluate, represents the causal effect of scoring first on winning. . . . .</p>	77
4.8	<p>(a) Graphical model representing the hypothesis that winning in hockey (<math>W</math>) is causally dependent on scoring first (<math>F</math>), which is deterministically related to the aggregated minimum scoring times of both teams (<math>M_a, M_b</math>). (b) Results of hypothesis tests conducted on 18k NHL hockey contests from 1993-2009 . . . . .</p>	78
4.9	<p>(left) Alternative hypothesis for the hockey scoring domain. Here, scoring first has no causal effect on winning; rather, any perceived dependence relationships are an effect of degree. (right) Distribution of first goal time as a function of total goals scored. The vertical lines designate the mean of each density curve. . . . .</p>	79

4.10	Augmented models for aggregation in Stack Overflow data. By including the structural <i>degree</i> variable, we can differentiate the two models from data by testing for conditional dependence between <i>Badge</i> and <i>Score</i> conditioned on <i>deg</i> . . . . .	80
4.11	Results of marginal and conditional tests of independence for aggregated Stack Overflow data. The plot shows the number of badge attributes that are marginally dependent with answer score for different aggregators. Conditioning on degree removes this dependence for several badge types. In the case of MIN, conditioning on degree can introduce a dependence. . . . .	81
5.1	(a) A simple graphical model describes the dependence between the number of editors <i>E</i> and quality <i>Q</i> of an article, but it does not account for common causes. (b) A more complex graphical model incorporates latent common causes <i>T</i> associated with project. . . . .	86
5.2	Different generative models for bipartite one-to-many data. In case (a), <i>X</i> directly influences <i>Y</i> . In (b), <i>X</i> and <i>Y</i> have a common cause ( <i>Z</i> ), and blocking and conditioning will both render them conditionally independent. In (c), blocking and conditioning are able to factor out the influence of confounder <i>Z</i> , but the two remain conditionally dependent. Case (d) depicts <i>Z</i> as a common effect of <i>X</i> and <i>Y</i> ; here, <i>X</i> and <i>Y</i> are rendered dependent when conditioned on <i>Z</i> (Berkson’s paradox), yet remain independent when <i>Z</i> is held constant through blocking using entities of type <i>A</i> . In all models, the double circles represent the deterministic dependence between $ID_A$ and <i>Z</i> . . . . .	91
5.3	Unlike conditioning, blocking does not induce conditional dependence when holding constant a common effect of two marginally independent variables. The line labelled “split” indicates a conditioning analysis with statistical power identical to the blocking analysis. . . . .	94
5.4	Blocking and conditioning are distinct operations, as they stratify the data in different ways. For the above relational data set, conditioning groups the data into two strata, yielding a combined $\chi^2$ value of 9.44 ( $p=0.009$ ) while blocking groups the data into three strata, producing a $\chi^2$ value of 8.75 ( $p=0.033$ ). . . . .	95

5.5	Models for bipartite data with latent variables. Models (a) and (b) depict cases where a latent common cause $H$ exerts influence on $X$ and $Y$ . In these cases, blocking is able to render $X$ and $Y$ conditionally independent, while conditioning is not. In models (c) and (d), $X$ and $Y$ have both a latent common cause $H_c$ and a latent common effect $H_e$ . Here, blocking will distinguish between the two models. . . . .	96
5.6	The effects of blocking and conditioning differ for data generated under the models shown in Figure 5.5b. Conditioning can only adjust for measured variable $Z$ , and is susceptible to high rates of Type I error as the strength of the latent effect $\beta_H$ increases. Blocking accounts for both $H$ and $Z$ ; it is not affected by $\beta_H$ . . . . .	97
5.7	Although relational blocking groups the data into smaller strata than conditioning, there is little effect on statistical power. . . . .	98
5.8	Data schemata for Wikipedia and Stack Overflow. Each pair of $X$ and $Y$ variables on the same entity can be tested for dependence, and related parent entities can be used for blocking. For example, Wikipedia Page.Quality and Page.Edits can be blocked through Project or User, while Stack Overflow Question.Score and Question.Length can only be blocked through User. . . . .	99
5.9	In many-to-many domains, relational blocks are determined by sets of parent entities rather than a single parent entity. . . . .	102
5.10	(a) Many-to-many blocking is able to adjust for common causes as effectively as traditional conditioning. (b) Since many-to-many blocking subdivides the data into small groups, statistical power decreases at low strengths of effect. . . . .	103
5.11	Targeted sampling can be used to increase statistical power. (a) and (b): Graphical models representing one-to-many relational data where $X$ and $Y$ are causally dependent (a), or marginally associated due to a common cause (b). For data with an exponential degree distribution (c), power can be increased by only considering large blocks when performing a conditional test (d). . . . .	104
5.12	The effects of targeted sampling on parent-child attributes are quite pronounced for data generated under model (a) for both aggregation (c) and replication (e). Under model (b), the effects are greatly reduced for aggregation (d), and non-existent for replication (f). . . . .	106

5.13	Examples of common effect cases in different domains. In each, two marginally independent factors ( $X$ and $Y$ ) can be rendered conditionally dependent when conditioning on a common effect $Z$ .	108
5.14	DAPER model graphs and ground graph for the common effect case. Although $X$ and $Y$ are marginally independent, conditioning on $Z$ will activate a d-connecting path and may render them conditionally dependent for both replication or aggregation to propositionalize.	109
5.15	The rules of deterministic d-separation agree with empirically derived independence relationships. (left) DAPER model describing the classroom example. (right) Ground graph for the classroom data. While conditioning on $G$ enables a path from $S$ to $W$ , blocking (conditioning on row $R$ ) does not.	111
5.16	When row groups are combined according to grade, the new groups are no longer representative of the overall population. (left) Comparison of relative attribute distributions for each grade group. (right) Absolute population distributions for each grade group.	113
6.1	Alternative hypotheses ( $H_3, H_4$ ) to those presented in Figure 3.7 ( $H_1, H_2$ ). In all four, $X$ and $Y$ are marginally dependent, but the causal structures behind the associations differ. Given that $Z$ is a latent variable, there are no conditional independence tests that can differentiate between $H_2$ and $H_4$ .	116
6.2	D-separation equivalence classes for one-to-many data propositionalized through replication. Gray boxes contain summary graphs for each class, where solid lines represent edges shared by all models of the class, dashed lines represent edges shared by some models, and arrowheads are present where direction is consistent among the models having the edge. The $\perp$ symbol stands for conditional independence, while the $\leftrightarrow$ symbol stands for the converse.	117
6.3	Markov equivalence classes for one-to-many data propositionalized through aggregation and separated without the use of relational degree information.	119
6.4	Markov equivalence classes for one-to-many data propositionalized through aggregation using degree information.	120

6.4	Markov equivalence classes for one-to-many data propositionalized through aggregation using degree information, cont'd. ....	121
-----	---	-----

# INTRODUCTION

This thesis represents a synthesis of *relational learning* and *causal discovery*, two subjects at the frontier of machine learning research. There exists a natural, methodological synergy between these two areas of study, and despite the sometimes onerous nature of each, their combination (perhaps counterintuitively) can provide advances in the state of the art for both.

Traditionally, propositional (or “flat”) data representations have dominated the statistical sciences. These representations assume that data consist of independent and identically distributed (iid) entities which can be represented by a single data table. More recently, data scientists have increasingly focused on data sets that are assumed to consist of interrelated, heterogeneous entities. The analysis of these “relational” data sets, once confined to a niche in the scientific literature [12, 61], has captured the attention of mainstream popular inquiry [5, 91].

Relational representations are more expressive than propositional ones, and can more naturally model many real world systems. However, given that an assumption of iid data is common to many statistical tests, the inherent interdependencies of relational data violate the assumptions of many widely used statistical procedures. For instance, previous work has demonstrated that failing to account for the interdependence among variables of related data instances can lead to an erroneous statistical conclusion of association when no such association exists [38, 41]. In addition, adopting a relational perspective is sometimes necessary merely to construct accurate models, because the most significant causal dependencies in the data hold between variables of related entities rather than merely within the variables of single data entities.

The second subject area of this thesis is causality. While we postpone a more formal definition of causality until Section 1.1, an intuitive sense of the term will suffice for the time being: We say that event  $A$  causes event  $B$  if and only if manipulating  $A$  changes the probability distribution of  $B$ . Nearly all work in machine learning and much of the work in statistics, in contrast, deals with statistical association alone, examining only the conditional and joint distributions of  $A$  and  $B$  and foregoing any inference about the effects of manipulation. Much of the current work in causality utilizes the graphical models framework, a useful tool for describing the causal relationships in data. Using the semantics of d-separation (see Section 1.1), we can enumerate the conditional independence facts that are entailed by different causal structures. As a result, by examining the conditional independence facts found in data, we can often draw causal conclusions.

Perhaps surprisingly, relational learning and causal discovery are rarely combined. Relational representations are wholly absent from the literature where causality is discussed explicitly. Instead, the literature on causality that uses the framework of graphical models assumes that data are propositional and thus iid.

Furthermore, very little of the work done in machine learning makes causal claims. While common in disciplines such as philosophy [85], economics [93], and epidemiology [36, 88], causal reasoning is largely absent from the machine learning literature, despite the widespread use of graphical model representations [69]. Research in relational learning in particular nearly always ignores causal mechanisms [23, 62]. In general, relational learning algorithms focus solely on establishing statistical correlation between properties or events. While useful, this goal is but a first step toward discovering a causal relationship.

This unexplored topical intersection represents an opportunity for advancement — by combining relational learning with causal reasoning, we can provide insight into the challenges found in each subject area. While the relational learning litera-



ture identifies some errors associated with naive analysis of non-iid data, very little work has been done to explain why or how these errors arise. By adopting a causal viewpoint, we can clarify the mechanisms that produce these errors. In addition, for many machine learning systems to be truly applicable and actionable, they must seek to discover causal knowledge rather than perform simple prediction. Analogously, we can utilize relational data to establish and strengthen causal claims in ways that are impossible using only propositional representations.

## Contributions of the thesis

This thesis will focus on structure learning of joint causal models for relational data sets through the synergy of work in statistical relational learning and causal discovery. To that end, I will present the following four primary contributions:

- *Defining propositionalization using graphical models* — Propositionalization is a set of widely used practices to convert a relational data set to a propositional data set. In this work, I show how to represent the propositionalization process using formal language and graphical models. In doing so, I identify graphically the conditions necessary for accurate statistical testing and causal conclusions. I show how to transform relational graphical models to their propositionalized forms. By including additional variables in our models to represent relational structure, we can explicitly model the interdependencies found within relational data and enable the extension of existing work in causality to relational domains.
- *Explaining previously identified biases in statistical tests on relational data* — I utilize graphical models to explain previously identified pathologies in relational learning. I discuss two sources of Type I error in particular—*instance dependence bias* and *degree disparity bias*—and explain their effects from a causal viewpoint. Furthermore, I use these results to suggest simple statistical tests

that account for the biases introduced by these pathologies, and provide evidence of their effectiveness on both real and synthetic data.

- *Defining and describing the properties of relational blocking* — I demonstrate that when modeled correctly, relational data sets enable types of causal reasoning that are impossible with iid data sets. Much of the past work in causality hinges on the assumption that all *common causes* are accounted for. I show that relational data allow for sound causal reasoning even in the presence of latent variables by conditioning on relational structure using *relational blocking*, a novel, relational generalization of a traditional analysis technique.
- *Demonstrating methods for automated causal discovery* — I show how the equivalence classes that are defined by conditional independence testing in relational data can be identified by analyzing the data schemata algorithmically. In addition, I show how to identify which specific tests will indicate the existence and/or direction of dependence between any pair of variables.

# CHAPTER 1

## BACKGROUND

In this chapter, we present several important concepts and representations useful for understanding causal discovery in relational data. At the heart of our discussion will be the use of the *ground graph* to represent the causal dependencies in a data set. Before introducing the ground graph, however, we review several related representations for causal and relational data sets. Examples of each can be found in Figure 1.1.

**Bayesian networks** are used to represent the causal dependency structure between variables in iid data in the form of a directed acyclic graph (DAG). The *d-separation* criteria can be used to identify conditional independence relationships between sets of variables, regardless of parameter settings. However, these models cannot represent dependencies that occur in (non-iid) relational data sets, since they are not expressive enough to represent different types of entities and probabilistic dependencies among the variables of such entities.

**Relational data graphs** explicitly represent the relationships between individual entities in a non-iid data set, though they do not represent the dependencies between the attributes of those entities.

**Entity-Relationship (ER) diagrams** compactly summarize the abstract structure of a relational data graph. In an ER diagram, each entity type is explicitly represented along with all of its associated attribute values. Possible relations are represented by edges augmented with a “crow’s foot” that indicate the existence of one-to-one, one-to-many, or many-to-many relationships.

**Directed Acyclic Probabilistic Entity-Relationship (DAPER)** diagrams add attribute dependency information to ER diagrams, using arrows to indicate direct causal dependence between attributes. The causal arrows are constrained by the existence of relations between the appropriate entities.

Finally, **ground graphs** combine the relational specificity of data graphs with the attribute dependency information of DAPER. The ground graphs represents an instantiation of a DAPER model for a particular data graph. A ground graph is a valid Bayesian network that represents dependencies between the variables associated with specific entities in the data graph. Since ground graphs define a coherent probability distribution, they may be analyzed using d-separation criteria.

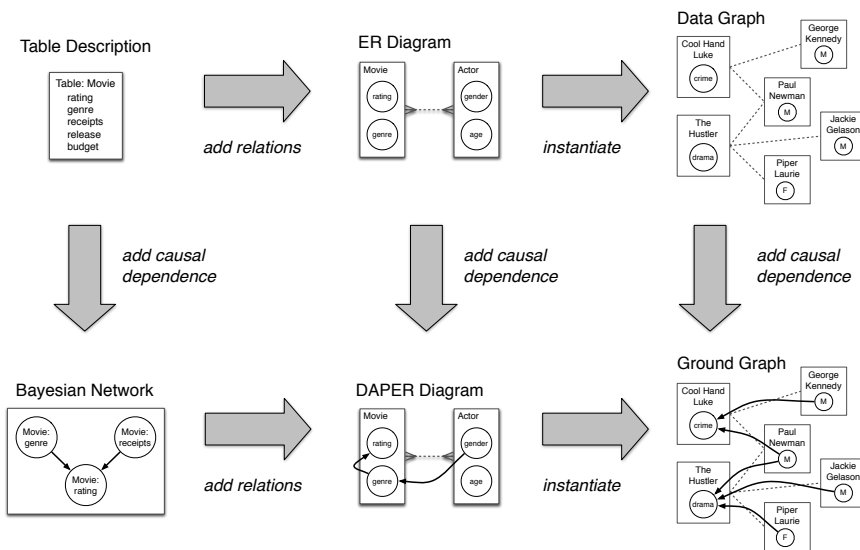


Figure 1.1: Relationships between different graphical representations. Adding explicit relational information to a traditional, table-based data description yields the more expressive ER diagram, which can then be instantiated (“rolled out”) into a relational data graph. Adding causal semantics to each of the three yields a Bayesian network, a DAPER model and a ground graph, respectively.

A schematic representation of the relationships between graph types is depicted in Figure 1.1. Here, we can clearly see the parallel relationships between different

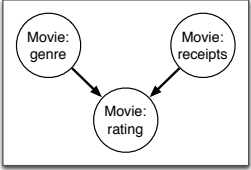
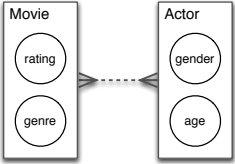
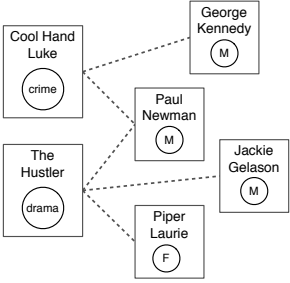
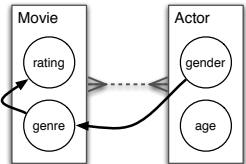
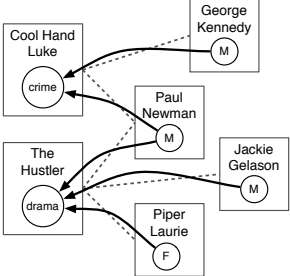
graph type	example	vertices	causal edges	relational structure
Bayesian Network		Attributes	Summary	None
ER Diagram		Entity types	None	Summary
Relational Data Graph		Entities	None	Instances
DAPER Model		Entity types (relations), Attributes (causal edges)	Summary	Summary
Ground Graph		Entity types (relations), Attributes (causal edges)	Instances	Instances

Table 1.1: Summary of different graph representations.

representations: Relational data graphs and ground graphs are instantiations of ER and DAPER, respectively; adding causal dependence to ER and data graphs produce DAPER and ground graphs.

The details of each type of graph are summarized in Table 1. In general, boxes represent entities (or types of entities), circles represent attributes, dashed lines represent relations, and solid lines represent causal dependence. In the sections that follow, we will examine each in greater detail.

## 1.1 Graphical models and causality

A small but growing effort in machine learning has focused on causal, rather than associational, models. In addition to computer science, formal reasoning about causal structures has roots in several fields; these include philosophy, economics, and statistics.

There is an active debate over the proper way to define causal dependence (see Holland [37] and associated comments in the *Journal of the American Statistical Association*, for example). Shadish, Cook and Campbell present a definition of causality that is rooted in experimental design [82], while Rubin provides a framework based on counterfactual logic [77], often referred to as the “potential outcome approach”. In the late 1980s and early 1990s, both Pearl [66] and Spirites, Glymour, and Scheines [85] formulated the “graphical models approach” to studying causal systems. The differing frameworks are not incompatible, but focus on differing aspects of analysis [17]. As Greenland and Brumback point out [31], potential outcome models are often useful for making inferences about individuals with regard to a single treatment and outcome, while the graphical approach is most suited to characterizing the existence and direction of the joint causal dynamics for an entire system and population. In this work, we are primarily focused on learning the structure of joint causal systems

rather than estimating individual effects, and therefore we adopt the graphical models perspective. A brief review of this framework is provided below.

The graphical approach to causality has its roots in Bayesian network modeling, with the added stipulation that the edges of the DAG are oriented to point from cause to effect. Any variable, whether measured or latent, can be considered both a cause and effect of disjoint sets of variables, and while it is not always made explicit, it is assumed that the direction of each edge respects the flow of time (e.g., a person’s height cannot cause their sex). In addition, we note that causality is inherently probabilistic in nature. If  $A$  is causally related to  $B$ , then changing the value of  $A$  changes the probability of  $B$ . For example, while it has been shown that smoking is causally related to certain types of cancer, smoking does not guarantee that cancer will occur.

A Bayesian network is a form of graphical model that compactly represents the joint probability distribution of a given set of random variables. At the core of this formulation is the representation of a probabilistic system as a directed acyclic graph (DAG). Given a set of variables  $V$  that characterize any given data instance, we can represent the joint probability distribution of  $V$  with the directed graph  $G = V, E$ . Given two variables,  $S, T \in V$ , the directed edge  $(S, T) \in E$  represents dependence between the two variables, and we refer to  $S$  as the “parent” variable and  $T$  as the “child.” For a node  $A$ , we define  $par(A) = \{S \mid (S, A) \in E\}$ . In addition, we let  $desc(A)$  denote the set of all nodes  $T$  such that there exists a directed path from  $A$  to  $T$ .

The validity of any conclusions drawn using the graphical models approach hinges on the assumption of the the *Causal Markov Condition*, which we briefly describe here (for a more complete treatment, we refer the reader to Pearl [66] or Scheines [80]). The Causal Markov Condition states that “A variable  $X$  is independent of every other variable (except  $X$ ’s effects) conditional on all of its direct causes” [80].

Symbolically, we denote the incoming edges to vertex  $X \in V$  as  $par(X)$  (“parents” of  $X$ ) and all nodes reachable with a directed path as  $desc(X)$  (“descendants” of  $X$ ). The above can then be written:  $X \perp\!\!\!\perp V \setminus desc(X) \mid par(X)$ .

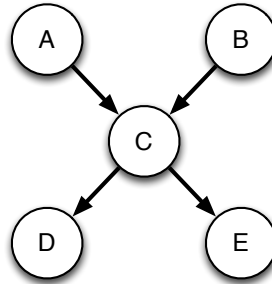


Figure 1.2: A simple Bayesian network.

Given a DAG  $G$  and a joint probability distribution  $P$ , we say that  $G$  and  $P$  are *compatible* under the Markov condition if we can factor the joint distribution such that  $P(A \mid \{G \setminus A\}) = P(A \mid parents(A))$ . For example, figure 1.2 depicts a Bayesian network for a small domain with five variables:  $V = \{A, B, C, D, E\}$ ,  $E = \{(A, C), (B, C), (C, D), (C, E)\}$ . Using the chain rule, we can express the joint distribution  $P$  as follows:

$$P(A, B, C, D, E) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)P(E|A, B, C, D)$$

The semantics of the network representation allow us to express the above in a far more compact form, however. Assuming that  $P$  and  $G$  are compatible, we can express the joint distribution in its factored form, as dictated by the edges in  $G$ :

$$P(A, B, C, D, E) = P(A)P(B)P(C|A, B)P(D|C)P(E|C)$$

In the equation above, the conditional probability of each variable is expressed only in terms of its parents in the DAG. This allows us to describe the system with far fewer parameters, making both learning and inference more computationally tractable. In



the above example, if each variable is binary, then explicitly representing full joint distribution requires  $2^5 = 32$  parameters. By factoring, we can instead represent the joint with a series of *conditional probability distributions* (CPDs), where each CPD represents the distribution of a variable conditioned on its parents. In our example, the CPDs require  $2^0 + 2^0 + 2^2 + 2^1 + 2^1 = 9$  parameters to represent the conditional probability distributions of variables  $A, B, C, D, E$ , respectively. For a more complete treatment of Bayesian networks, we refer the reader to Charniak [13], Heckerman [34], or Jensen [43].

### 1.1.1 d-separation

As with associational Bayesian networks, the causally interpreted DAG offers a compact way to represent conditional independence relationships within data. The mechanism for identifying these relationships is Pearl’s notion of *d-separation* [66]. The d-separation criteria describe the graphical scenarios that entail conditional independence relationships in data, and can be derived directly from the Markov condition [60]. When nodes in a DAG are d-separated, they are conditionally independent; when they are d-connected, they can be dependent. We briefly review these concepts below; for a more thorough introduction, see Geiger [25], Scheines [80], or Spirtes [85].

Two sets of nodes  $U$  and  $V$  are *d-connected* if there exists an undirected, collider-free path from some node  $u$  in  $U$  to some node  $v$  in  $V$ . “Collider-free” means that no nodes along the path have two incoming edges that are also part of the path. In the small graph in Figure 1.3,  $\{A\}$  and  $\{E\}$  are d-separated, since the only path connecting them ( $A \rightarrow D \rightarrow G \leftarrow E$ ) contains a collider ( $G$ ).  $\{A\}$  and  $\{F\}$  are d-connected, since there is a collider-free path connecting them ( $A \rightarrow D \rightarrow F$ ). For simplicity, we may notate a singleton set as a single variable (e.g.,  $\{A\}$  as  $A$ ). Symbolically, we express these facts as  $A \perp\!\!\!\perp E$  ( $A$  and  $E$  are independent) and  $A \not\perp\!\!\!\perp F$

( $A$  and  $F$  not independent). Whether or not a node is considered a collider is with respect to the path being considered, thus  $B$  and  $C$  are d-separated by collider  $E$  on path  $B \rightarrow E \leftarrow C$ , but  $B$  and  $\{G, H\}$  are d-connected via the paths  $B \rightarrow E \rightarrow G$  and  $B \rightarrow E \rightarrow H$ .

Conditioning plays an important role in the definition of d-separation. The paths mentioned above are valid for the marginal case, where no variables are used for conditioning. According to the semantics of d-separation, if we condition on a non-collider, it “blocks” any undirected paths on which it lies, and that path becomes d-separating rather than d-connecting. For instance, in Figure 1.3 we have  $A \not\perp G$  in the marginal case due to the existence of path  $A \rightarrow D \rightarrow G$ . However, conditioning on  $D$  will block the path connecting  $A$  and  $G$ , rendering them conditionally independent ( $A \perp G \mid D$ ).

Conversely, conditioning on a collider (or any of its descendants) will “unblock” a path. Marginally, we have  $F \perp H$ , since the only path connecting them ( $F \leftarrow D \rightarrow G \leftarrow E \rightarrow H$ ) contains collider  $G$ . Conditioning on  $G$  will unblock the path and render  $F$  and  $H$  conditionally dependent ( $F \not\perp H \mid G$ ). Likewise, even though  $B \perp C$ , we have  $B \not\perp C \mid H$ . Since  $H$  is a descendent of collider  $E$ , conditioning on  $H$  will unblock path  $B \rightarrow E \leftarrow C$ .

### 1.1.2 d-separation with determinism

As described above, all the conditional dependence entailed by the DAG must be probabilistic in order to satisfy the causal faithfulness assumption [85]. However, with slight modification, the d-separation criteria can be expanded to apply to systems that include deterministic relationships between variables [25]. We say that a set of variables  $W$  determines  $X$  if all variables in  $X$  can be computed from some deterministic function of the variables in  $W$ . In the probabilistic case,  $X$  and  $Y$  are d-connected if there exists a path such that all colliders (or one of their descendants)

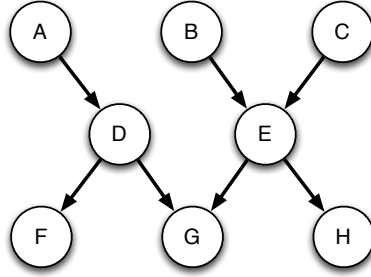


Figure 1.3: Causal DAGs represent conditional independence relationships with *d-separation*. Marginally,  $\{A\}$  and  $\{F\}$  are d-connected by the collider-free undirected path  $A \rightarrow D \rightarrow F$ , as are  $\{G\}$  and  $\{H\}$  with path  $G \leftarrow E \rightarrow H$ .  $\{B\}$  and  $\{C\}$  are marginally d-separated, since the only path connecting them ( $B \rightarrow E \leftarrow C$ ) contains a collider ( $E$ ). If we condition on  $\{E\}$ , then  $\{G\}$  and  $\{H\}$  become d-separated, but  $\{B\}$  and  $\{C\}$  become d-connected.

are part of the conditioning set, and all non-colliders are not part of the conditioning set. When deterministic variables are present, paths can be blocked by unconditioned non-colliders if they are determined by variables in the conditioning set. In the DAGs shown here, variables depicted with a gray double ring are determined by their parents. For example, in the network shown in Figure 1.4,  $\{F\}$  is d-separated from  $\{G\}$  when we condition on  $\{A\}$ , since  $D$  is determined by  $A$ .

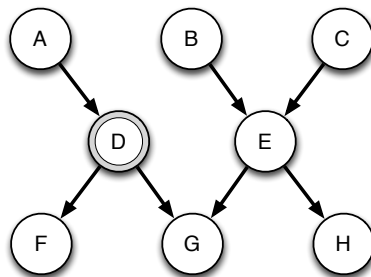


Figure 1.4: Graphical model with a deterministic edge ( $A \rightarrow D$ ). When deterministic relationships are present, a d-connecting path is blocked by non-colliders who are determined by variables in the conditioning set. Here,  $F \perp\!\!\!\perp G \mid A$ . This relationship differs from the system depicted in Figure 1.3, which had no determinism.

The notion of d-separation under determinism (sometimes denoted as “D-separation” with a capitalized D) was pioneered by Geiger [25]. Spirtes et al. [85] expanded this work to include a class of systems where the deterministic relationships are too complex to be represented in the DAG (instead, a complete list of deterministic dependencies is generated to accompany the network). In this thesis, we limit our discussion to the systems discussed by Geiger, where all deterministic dependencies can be explicitly represented by edges in the DAG.

### 1.1.3 Ground graphs

Bayesian networks are a compact way of representing the overall dependency structure for an entire data set. They generalize the probabilistic dependencies between variables of a given system over all worlds, under the assumption that all instances are independent and identically distributed (iid). Typically, these instances are represented by a single table or database view, with one row per instance and one column per variable. Given a Bayesian network  $G$  and a set of data instances  $D$ , we can generate the ground graph—a larger, “rolled out” graphical model which represents the system over all *worlds* having the same set of instances, generative process, and dependence structure. The ground graph is a more specific representation; the worlds it represents only differ in the actual attribute values associated with each instance.

Figure 1.5 illustrates the rollout process for a data set with five instances and four variables, resulting in a ground graph consisting of 20 vertices. For a propositional model, the procedure is simple: For each instance in  $D$  (a), we create a variable in the rolled out model  $G_g$  (c) for each variable in the DAG  $G$  (b). We draw edges between vertices in  $G_g$  when the corresponding variables in  $G$  share an edge.

When generated in this manner, the resulting graph  $G_g$  is a valid graphical model, and the independence relationships between instance-level variables in  $G_g$  can be identified using d-separation. For example, in Figure 1.5c, variable  $a_1 \perp\!\!\!\perp d_1 \mid b_1$ .

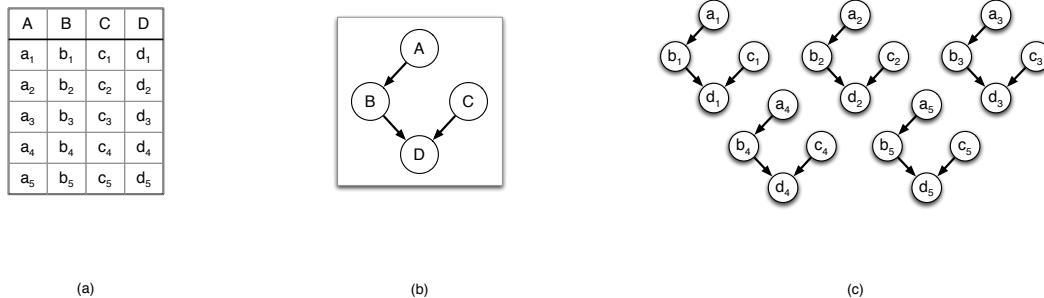


Figure 1.5: Given a set of instances (a), graphical models representing their conditional independence relationships (b) can be rolled out to produce a ground graph representing dependence relationships across worlds (c).

In addition,  $a_1 \perp\!\!\!\perp b_2$ , by virtue of the fact that there exists no path (collider-free or otherwise) between  $a_1$  and  $b_2$ . For the moment, the simple example shown in Figure 1.5c does not seem to offer any representational power or convenience over the compact graphical model in 1.5b. However, the ground graph representation will be important for our discussion of non-iid (relational) domains, where edges in the ground graph can reach across instances, rather than being limited to connecting variables of the same instance.

#### 1.1.4 Learning algorithms

Several algorithms exist for learning causal graphical models from data. These algorithms often divide the learning process into the subtasks of *structure learning* and *parameter estimation*. Structure learning identifies which variables in a system are causally dependent; parameter estimation quantifies the strength of these associations by making maximum likelihood estimates of parameters given an assumed functional form.

Algorithms for structure learning fall into at least two categories. *Constraint-based* algorithms, such as the LCD [15] and PC [84], identify constraints on the space of causal models that are implied by conditional independencies observed in the data. Alternatively, *search-and-score* algorithms [10, 16, 34] evaluate the space of possi-

ble models in terms of a penalized likelihood function, finding the most likely model given training data. While search-and-score algorithms are often effective at finding high-likelihood structures, they do not necessarily capture the conditional independence relationships that are more suited to causal reasoning [19]. Furthermore, they typically return a single, maximum-likelihood model rather than a family of related models that are possible given the data.

The work presented here falls into the category of constraint-based structure learning under the assumptions outlined in Section 1.1.1. Here, we will examine the PC algorithm as an exemplary constraint-based approach; however, the applicability of the techniques presented in this thesis range beyond a single algorithm.

PC further subdivides structure learning into two phases. First, the algorithm determines the *skeleton* of the DAG by exploring the space of possible conditional independencies among variables by using statistical tests for conditional independence. All pairs of variables that cannot be rendered marginally or conditionally independent are then connected with an undirected edge in the skeleton. Once the skeleton of the DAG is in place, a series of *edge orientation* rules is applied to convert some undirected edges into directed edges.

To identify the skeleton, PC starts with a completely connected, undirected graph  $G^* = V, E$ , with a vertex for each variable in the data, and proceeds as follows:

```

 $G \leftarrow G^*$ 
 $l \leftarrow 0$ 
while  $\exists s \in V$  such that  $|neigh(s)| \geq l$  do
    for all  $s, t \in E$  do
        for all  $S_n \in subs_l(neigh(s))$  do
            if  $s \perp\!\!\!\perp t | S_n$  then
                 $E \leftarrow E \setminus (s, t)$ 
                break
     $l \leftarrow l + 1$ 

```

Where  $neigh(x)$  denotes the neighbors of  $x$  in the graph, and  $subs_l(X)$  is the set of all subsets of  $X$  of size  $l$ . The intuition behind the PC algorithm is simple: Start with the assumption that all pairs of variables are dependent, then systematically check for conditional independence using all possible conditioning sets for each variable pair. The specifics of the algorithm exploit the fact that, when one pair of variables is found to be conditionally independent, the number of possible conditioning sets for other variable pairs is decreased, thus mitigating the computational complexity of a naive approach. Note that the core of this algorithm depends on the ability to accurately check for conditional independence, a subject we will visit throughout this thesis.

Edge orientation transforms an undirected causal skeleton by applying a series of orientation rules and directing edges such that the resulting DAG entails the conditional independence relations defined by d-separation. The literature contains numerous algorithms for directing edges; for the purposes of illustration we discuss the orientation rules outlined by Meek [44, 58], illustrated graphically in Figure 1.6.

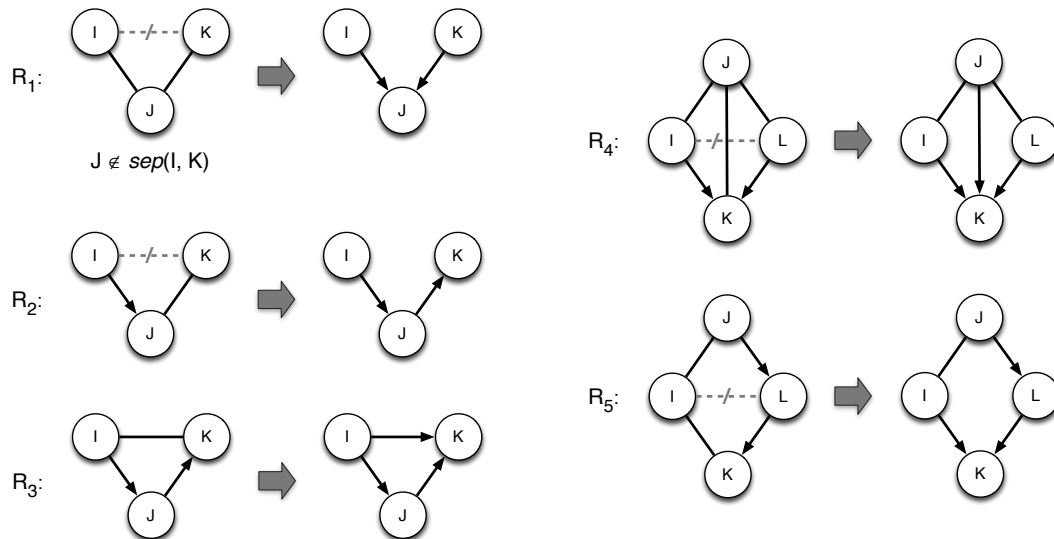


Figure 1.6: Edge orientation rules for constructing a DAG from a causal skeleton. Dashed lines denote the explicit nonexistence of an edge.

The edge orientation algorithm takes a causal skeleton and a set of minimal conditioning sets as input. The conditioning sets are defined over all pairs of nonadjacent nodes from the skeleton. The function  $sep(X, Y)$  returns the set  $S_{XY}$  such that  $X \perp\!\!\!\perp Y \mid S$ . Starting with the skeleton, the rules are applied repeatedly (in order) until all possible edges have been oriented.

Rule  $R_1$ , also known as the “collider rule,” states that any time we have two nonadjacent variables ( $I$  and  $K$ ) that share a neighbor ( $J$ ), and that neighbor is not part of their conditioning set, we should orient the edges to form a collider. Doing otherwise would violate the assumption that  $I \not\perp\!\!\!\perp K \mid J$ . This rule is applied exhaustively before all others, after which no new colliders may be created through the application of subsequent rules. Rule  $R_2$  follows directly from this assumption, as orienting the edge otherwise would create a collider in  $J$ .  $R_3$  enforces the acyclicity constraint of the DAG. Rule  $R_4$  follows from the fact that orienting an edge from  $K$  to  $J$  would create a collider in  $J$  upon a subsequent application of  $R_3$ . Rule  $R_5$  follows similar logic, but with two subsequent applications of  $R_3$ . For a proof of the correctness of these rules with respect to conditional independence and d-separation, we refer the reader to Meek [58].

Figure 1.7 shows the application of the edge orientation rules to a simple DAG. The algorithm starts with the skeleton and conditioning sets for each nonadjacent pair of variables. First, rule  $R_1$  is applied twice to identify colliders  $C$  and  $F$ . Next,  $R_2$  is applied to orient  $C \rightarrow E$ . Not all edges can be oriented from conditional independence information, as exemplified by the edge between  $B$  and  $D$  remaining undirected. No matter how this edge is directed, the conditional independence relationships among the variables do not change. Accordingly, none of the rules shown in Figure 1.6 apply. Sets of models that represent indistinguishable probability distributions in the data are referred to as *Markov equivalent* sets.



<i>sep()</i>	B	C	D	E	F
A	{D}	$\leftrightarrow$	{F}	{}	{B,C}
B		$\leftrightarrow$	$\leftrightarrow$	{C}	{C,D}
C			{B}	$\leftrightarrow$	$\leftrightarrow$
D				{B}	$\leftrightarrow$
E					{C}

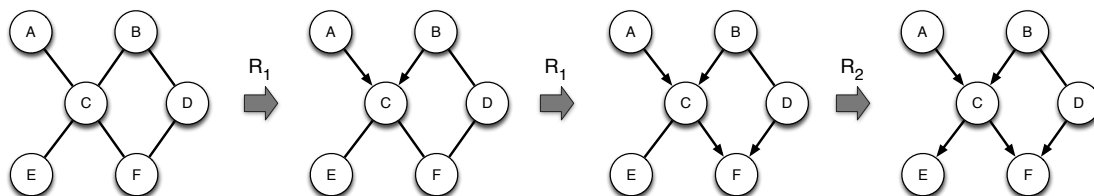


Figure 1.7: Application of edge orientation rules to the graph from Figure 1.3. Starting with a skeleton and a set of separating sets (table), edges are oriented through successive application of the rules shown in Figure 1.6. Since the direction of the edge between  $B$  and  $D$  is not identifiable through conditional independence, it cannot be oriented.

In addition to the causal Markov condition, the correctness of PC hinges on two additional assumptions that are commonly made about the generative process underlying the data. We describe them here, but refer the reader to Pearl [66] or Scheines [80] for a more thorough explanation.

- *Causal Sufficiency* asserts that any common causes of variables in  $V$  are explicitly also in  $V$ . Note that this does not preclude us from examining systems in which hidden causes are present (at some level of granularity, there are *always* hidden causes behind our data); rather, the causal sufficiency assumption guarantees that latent causes are not shared among modeled variables.
- *Faithfulness* states that the only conditional independence relationships that exist in a data set are those that are explicitly represented in the graph. For example, faithfulness assumes that two causal pathways between two variables cannot cancel out to make the variables marginally independent.

## 1.2 Relational data representations

Nearly all machine learning algorithms assume that data are composed of independent, identically distributed (iid) records. These data are often represented in the form of a single table, in which each row corresponds to a single entity (or unit) of interest (person, event, etc.), and each column contains the values of the attributes associated with that unit. Under the iid assumption, the attribute values for any one entity provide no information about the values in any other. In other words, the table exhibits *row independence*. However, the attribute vectors across instances are not multivariate iid in the traditional sense, as there may exist dependence between attributes for a given instance (i.e., they are not *column independent*). For example, a data table consisting of heights and weights for randomly selected individuals would be considered iid—while the values across any given row are dependent (since height and weight are associated), the values are independent across rows.

In many real-world scenarios, supposedly independent units can exhibit causal influence on each other, resulting in data that exhibit dependence among instances. For example, individuals targeted by a survey may communicate, patients in a hospital may infect each other, and peer groups may encourage like behavior. In addition, many real-world systems are made up of heterogeneous entity *types*, and instances of one type may influence the attributes of another. For example, a manager may influence employee behavior or record companies may partially control artist output.

In the past decade, a growing effort in machine learning research has focused on relational data sets that are known to violate the iid assumption in these ways (for a good overview, see Getoor [28]). In relational data, individual records are not statistically independent, and information about some records may provide insight into the values of others. Furthermore, they are not necessarily identically distributed, as they may be composed of multiple, heterogeneous data types.

Relational data sets can be represented graphically, with vertices (or nodes) corresponding to entities, and edges (or relationships) representing the connections between them. For instance, the Yahoo! Music data graph contains relationships between artists, albums, and songs [73], and the Enron email graph consists of nodes representing individuals joined by relationships representing their email correspondence [56]. Bibliographic data sets, such as HEP-TH, consist of scientific papers joined by citations [57].

In the chapters that follow, we will present several empirical results of analysis performed on data drawn from Stack Overflow<sup>1</sup>, a website that allows users to post questions and answers concerning problems in computer programming. The Stack Overflow data comprises *users*, *questions*, and *answers*, as illustrated in Figure 1.8. Users may post new questions or provide answers to existing ones, as well as score the quality of the questions and answers posted by others. Given the rich relational structure, the data exhibit dependencies among attributes. For instance, the scores of questions posted by a common author tend to be associated.

### 1.2.1 Relational semantics

In this section, we provide a formal definition of the elements of relational data sets (while not identical, the definitions provided below draw heavily on the work of Getoor [27] and Heckerman [35]). Relational data sets are made up of set of *entities*  $E$ , *relationships*  $R$ , and *attributes*  $A$ , defined as follows:

- **Entities**  $E$  represent the statistical units of observation [82], divided into a mutually exclusive *type groups*, determined as follows. We define a finite set of *types*  $T = \{t_0, \dots, t_n\}$  and the function  $type : E \mapsto T$ . We define a *type group*  $E_t = \{e \in E \mid type(e) = t\}$ , thus  $E = E_{t_0} \cup E_{t_1} \cup \dots \cup E_{t_n}$ .

---

<sup>1</sup><http://stackoverflow.com>

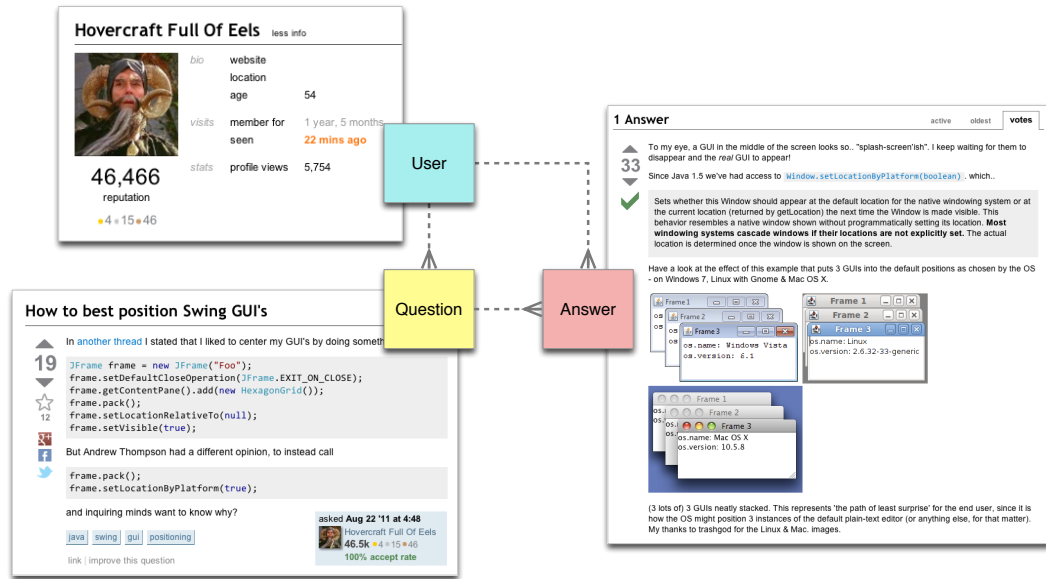


Figure 1.8: The Stack Overflow data comprises *users*, *questions*, and *answers*, and are connected by three types of relationships (user-question, user-answer, question-answer).

- **Relationships**  $R = \{r \in E \times E\}$  denote relationships between pairs of nodes. Formally,  $R$  is a binary relation in the mathematical sense, equal to a subset of the cartesian product of  $E$  with itself. We let  $R_{s,t}$  denote relationships between entities of type  $s$  and  $t$ ,  $R_{s,t} = \{(a, b) \in R \mid a \in E_s, b \in E_t\}$
- **Attributes**  $A = \{A_0, A_1, \dots\}$ , a set of mathematical functions mapping entities or relations to some value. The domain of each function  $A$  is one or more type groups  $E_t$  (for entity attributes) or the set of relationships  $R_{s,t} \subset R$  connecting nodes of type group  $E_s$  with  $E_t$  (for relationship attributes).

In the aforementioned examples, relationships are used to represent some real world interaction<sup>2</sup> between entities. Regardless of domain, the semantic meaning of

<sup>2</sup>In this work, we use the term “relationship” to describe a connection between entities, and “dependence” or “association” to denote probabilistic correspondence between the attributes of those entities.

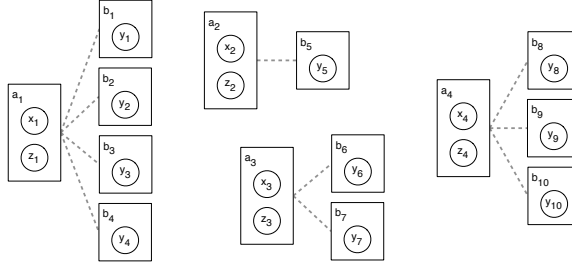


Figure 1.9: Data graph for a small relational data set with two entity types. Entities of type  $A$  ( $a_1, \dots, a_4$ ) have attributes  $X$  and  $Z$ , while entities of type  $B$  ( $b_1, \dots, b_{10}$ ) have attribute  $Y$ .

the relationships is the same: Relationships represent a possible dependence between the attribute values of connected entities (or, perhaps more naturally, the absence of a relationship guarantees independence). Thus, for all  $u, v \in E$ , if  $r = \{u, v\} \in R$ , then  $A_s(u)$  and  $A_t(v)$  are possibly dependent for all attributes  $A_s$  and  $A_t$  defined on  $u$  and  $v$ , respectively. Additionally, there is possible dependence between  $A_s(u)$ ,  $A_t(v)$  and  $A_r(r)$  for all attributes defined on relationship  $r$ . For this work, we make the simplifying assumption similar to Xu et al. [92]; that is, causal dependence can only exist between attributes of entities that share a direct relationship, and that “multi-hop” influence does not exist without an intermediate variable. When necessary, we denote attributes and the entities or relationships with which they are associated by the expression “entity.attribute”. Thus, for an attribute named “age” and an entity named “person”, person.age would be the full name of the attribute. Where it is clear from context, we omit the entity designation.

In the Stack Overflow data, we have three entity types (users, questions, answers), with three sets of relationships (user-question, user-answer, question-answer). Thus, answer scores may be dependent on attributes of the author who provided them or the specific question that they are posted to, but not others.

### 1.2.2 Data graphs and entity-relationship diagrams

We can represent the entities and relationships in a relational data set using an undirected graph called a *relational data graph*. An example of a small data graph is shown in Figure 1.9. In this example, the data graph  $G$  consists of 14 entities connected with 10 relationships:

$$E = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$$

$$R = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_1, b_4), (a_2, b_5), \\ (a_3, b_6), (a_3, b_7), (a_4, b_8), (a_4, b_9), (a_4, b_{10})\}$$

$$A = \{A_x : E_a \mapsto X, A_z : E_a \mapsto Z, A_y : E_b \mapsto Y, \}$$

Again, causal dependencies are limited to pairs of attributes on the same entity or to pairs of attributes on entities connected by one or more relationships. In the data set shown in Figure 1.9, attribute value  $x_1$  may be dependent on  $y_4$ , since their associated entities  $a_1$  and  $b_4$  share a relationship. However,  $x_1$  and  $y_5$  are necessarily independent, since there is no relationship between  $a_1$  and  $b_5$ .

The above example exhibits a common characteristic of relational data sets: heterogeneity of data types. In propositional data, data consist of a single type of entity, or unit, and its associated attributes. In contrast, relational data can consist of multiple entity types. For example, the Stack Overflow data set consists of users, questions, and answers, each with different attributes. To compactly represent the different types in our data and the relational structure between them, we utilize *entity-relationship* (ER) diagrams. ER diagrams are commonly used to describe the table structure of relational databases [70]. Relational database management systems (RDBMS) are often used to store relational data sets. In a RDBMS, data are stored in tables that can be queried and exported as tuples using SQL operators. When multiple table rows contain the same foreign key, their associated data are often nonindependent, and tables generated by joins performed on foreign keys will be non-iid. In this sense, foreign keys correspond to the relationships mentioned above.

Of course, the above definition of relational data sets is not limited to those that can be represented by an RDBMS; thus, we use the ER diagram to schematically represent the link structure between entity types rather than to reflect the table structure of an RDBMS. For our purposes, we represent each entity type with a plate (box). Attributes associated with that entity are shown as circles drawn within that plate, and possible relations between entity types are represented with dashed lines. Additionally, these lines are annotated with “crow’s foot” notation that indicates whether the connected entities are related in a one-to-one, one-to-many, or many-to-many manner.

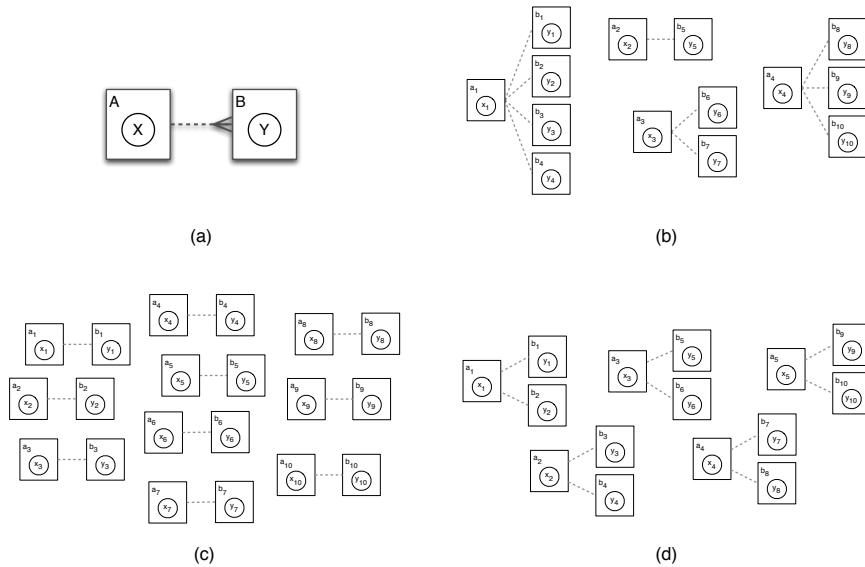


Figure 1.10: Entity-relationship diagram (a) and three possible relational data sets that it describes (b-d).

Figure 1.10a depicts a simple ER diagram for bipartite one-to-many data. For this data set, each  $A$  entity is linked to one or more  $B$  entities. Note that the ER diagram under-specifies the data graph, as the specific relational structure of the data is not captured. For example, Figures 1.10b-d depict different data sets that are all valid for the diagram in 1.10a. Thus, while a valid ER diagram can be constructed from

any data graph, the reverse is not true, as ER is merely a template for the relations that are actually present.

### 1.3 Graphical models for relational data

The relational data graphs described above serve a different purpose than the graphical models discussed in Section 1.1. Bayesian networks represent the dependencies between the *attributes* of a given data set, while relational data graphs represent a possible dependency between the attribute values of specific entities in a given data set. In this section, we demonstrate how to combine these two representations in order to reason causally with relational data sets.

As we will detail in Section 3.1, statistical tests and algorithms that do not account for the inter-entity dependencies may be prone to error [38, 41], due to a lack of causal sufficiency. Conversely, if the dependencies between entities are modeled and exploited, learning performance can increase dramatically [40, 61, 72]. A key contribution of this research is to demonstrate how the advantage provided by a relational model representation carries over to causal claims.

#### 1.3.1 DAPER models

ER diagrams summarize relational structure between entities, but do not represent the dependence structure among the attributes of those entities. *Directed acyclic probability entity relationship* (DAPER) models combine the graphical syntax and semantics of traditional Bayesian networks with the ER representation of link structure [35]. In the DAPER representation, boxes and diamonds indicate entities and relationships, respectively. Dotted lines indicate connections between entities and relationships via primary/foreign keys, and line endings indicate relationship types (e.g., one-to-many). Circles indicate random variables, connected by a dashed line to their associated entities and relations. Solid arrows represent directed dependencies



between attributes, and may connect the attributes of the same entity (intra-entity edges) or different ones (inter-entity edges). Self-relationships are allowed for domains where entities of the same type share relations and represent peer-influence on the same variable. An example DAPER diagram can be found in Figure 1.11a.

Using a rich annotation syntax, the DAPER representation can model a variety of relational structures and attribute dependencies. In this work, we do not utilize the full expressive power of DAPER and omit some of its graphical conventions for clarity. For our purposes, variables are drawn inside the entities to which they correspond, and relationships are represented without an explicit existence variable. Given this, our notion of a DAPER model is equivalent to a class dependency graph as defined by Getoor [27]. In addition DAPER models are functionally similar (and expressively equivalent) to “plate” models found in the graphical modeling literature (see Heckerman for an of the equivalence of different representations[35]).

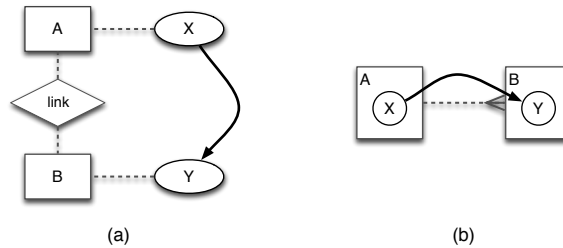


Figure 1.11: DAPER models combine the the structural representation of entity-relationship diagrams with the probabilistic dependence representation of plate-structured graphical models. (a) Example of a DAPER model for bipartite data where X and Y are associated, as presented by Heckerman et al.[35] (b) A simplified version of the same model, with attributes drawn inside the boxes representing the entities they are associated with.

While DAPER models capture the generative process of relational data sets, they are typically not combined with the machinery of d-separation. Furthermore, since relational structure is represented separately from attribute values, characteristics of the relational structure (save relationship existence) cannot be used for conditioning,

limiting the usefulness of the standard DAPER representation for causal reasoning in relational data.

### 1.3.2 Ground graphs

The relational structure of the data graph represents constraints on the possible dependencies between attribute values and thus abstracts the actual dependence structure. To instantiate this relational structure, we combine the dependence information in the DAPER model with the instantiated detail of the data graph to produce a representation called a *ground graph*. Just as DAPER models combine graphical model semantics with ER diagrams, ground graphs attach attribute dependence structure to data graphs. In the same way that Bayesian networks for propositional data sets can be rolled out into an instance-level graphical model of a propositional data set (as in Figure 1.5), DAPER models can be rolled out to produce an instance-level graphical model of a relational data set.

Given a data graph  $G$  and compatible DAPER model  $D$ , the ground graph can be constructed algorithmically (what follows is equivalent to the rollout procedures described by Getoor [27] and Heckerman [35]). For each intra-entity DAPER edge connecting variables  $P$  and  $Q$  that are associated with the same entity type, draw an edge in the ground graph between the corresponding attribute values  $P(u)$  and  $Q(u)$  for all entities  $u$  of the appropriate type. For each inter-entity edge from  $P$  to  $Q$ , draw an edge in the ground graph from  $P(u)$  to  $Q(v)$  for all node pairs  $u$  and  $v$  that are connected with a relation in the data graph. Figure 1.12 depicts the ground graph constructed from a small data graph and accompanying DAPER model. While the entities and relations are not technically part of the ground graph, we will often depict them along with the graphical model for clarity. Note that for DAPER models that contain self-relationships, the resulting ground graph may not be acyclic; in these

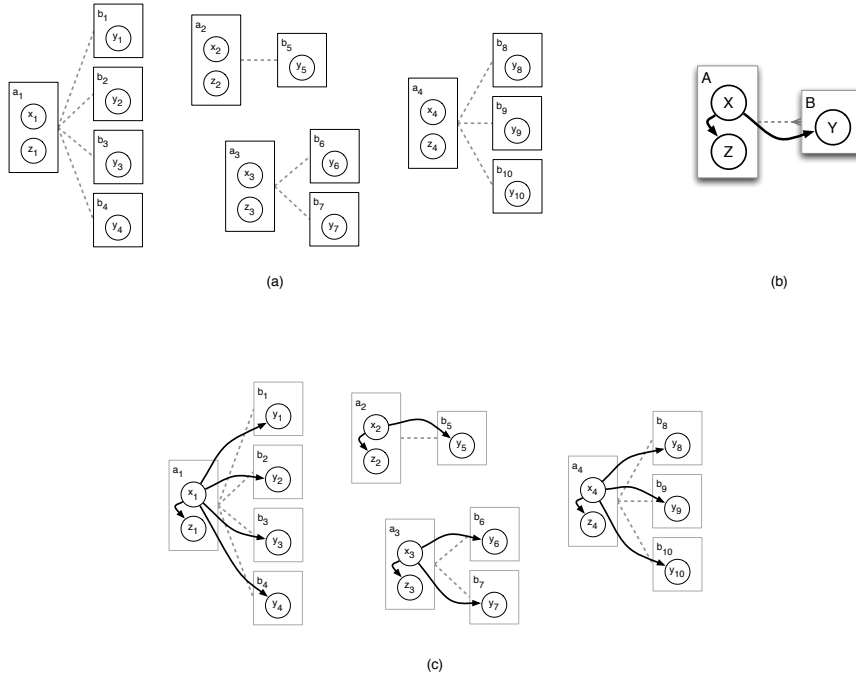


Figure 1.12: The probabilistic ground graph (c) can be constructed by applying a DAPER model (b) to an appropriate relational data graph (a).

cases, we must incorporate some form of additional constraints to ensure that the rolled out graph is a valid DAG.

Getoor demonstrated that a rolled out network constructed in this fashion defines a coherent Bayesian network [27]. In her terminology, a “relational skeleton” (data graph) combined with an appropriate “class dependency graph” (DAPER) begets an “instance dependency graph” (ground graph). Given this result, we can examine the ground graph using d-separation criteria. Getoor also proves that an acyclic DAPER model will necessarily generate an acyclic ground graph for any suitable data graph, and that even DAPER models that contain cycles (such as those for unipartite data sets) can be associated with a well-defined ground graph. Using similar reasoning, we can show that that d-separation properties can be partially extended to a DAPER model as well.

**Theorem 1.3.1.** *Given a DAPER model  $D$  and a causally sufficient ground graph  $G_g$  with variables  $X_i$  and  $Y_j$  corresponding to variables  $X$  and  $Y$  in  $D$ . Let  $Z^* = \{Z_1, Z_2, \dots, Z_n\}$  be a set of variables in  $G_g$  such that  $Z^*$  comprises all the variables  $Z_k$  in  $G_g$  that correspond to one or more variables  $Z$  in  $D$ . If  $X \neq Y$ , and  $X_i$  and  $Y_j$  are  $d$ -connected in  $G_g$  when conditioned on  $Z^*$ , then the corresponding  $X$  and  $Y$  will be  $d$ -connected in  $D$  when conditioned on  $Z$ .*

*Proof.* We can prove this result by contradiction. Assume that for some ground graph  $G_g$  we have a  $d$ -connecting path  $p$  from  $X_i$  to  $Y_j$ , but no corresponding path in the DAPER model  $D$ . For each edge along  $p$ , there is a corresponding edge in  $D$  that is oriented in the same direction. We can construct a  $d$ -connecting path  $p_D$  in  $D$  using these edges, which is a contradiction.  $\square$

**Corollary 1.3.2.** *Given a DAPER model  $D$  and instantiated ground graph  $G_g$ . If variables  $X$  and  $Y$  are  $d$ -separated in  $D$  given conditioning set  $Z$ , then all corresponding  $X_i$  will be  $d$ -separated from  $Y_j$  when conditioned on all  $Z_k$ .*

*Proof.* This follows directly by contrapositive restatement of Theorem 1.3.1.  $\square$

### 1.3.3 Regression-based relational modeling techniques

In the social sciences, data that consist of hierarchical types are often modeled using *mixed effects models* [2, 21]. Mixed effects models are a type of generalized linear regression model where the factors that influence an outcome variable are characterized as *random effects*, whose parameters (slopes and/or intercepts) are directly modeled, and *fixed effects*, whose parameters are not modeled. These models are subsumed by *multilevel models* (also known as *hierarchical models*), where several “levels” of parameters can be learned simultaneously from data [26, 30]. For example, a model of movie receipts might have a parameter governing the influence of the studio on its success, where the coefficient associated with the studio is itself the outcome variable in a higher-level model equation.

While these models are quite effective at predicting the value of target variables given a set of input variables, they do not make explicit the direct causal dependence and conditional independence relationships in the system being studied [85]. Since they are conditional rather than joint probability models, they cannot represent the types of reasoning about causal dependence exemplified by the skeleton construction and edge orientation in the PC algorithm. Furthermore, most multilevel methods assume a fixed hierarchy of influence, or at the very least a regular data structure (e.g., all employees have exactly one boss, all children have exactly two parents).

## 1.4 Validity

In this chapter, we have reviewed several pieces of prior work. Causal representations such as Bayesian networks allow us to probabilistically represent causal systems and reason about them using d-separation and conditional independence. Additionally, we outlined four complementary relational data representations (ER diagrams, data graphs, DAPER models, and ground graphs).

In the following chapters we demonstrate how to effectively combine causal semantics with relational data in novel ways, allowing us to draw causal conclusions in non-iid domains. At the heart of this discussion are the connections between the use of different models and the implications in terms of different threats to validity. In this work, we will primarily focus on two such threats. *Statistical conclusion validity* refers to the inappropriate use of a test statistic or violation of assumptions such as iid. In addition, we will outline cases that threaten *internal validity*, where incorrect causal conclusions can be drawn from merely associational results. For a more in-depth discussion of different types of threats to validity, we refer the reader to the work of Shadish, Cook, and Campbell [82].

First, we introduce the concept of *propositionalization*, the process of transforming a relational data set into a form suitable for conditional independence testing.

## CHAPTER 2

### PROPOSITIONALIZATION

The chief difficulty of working with relational data is often statistical in nature rather than representational. In relational domains, we often want to assess the association between variables on two different entity types (e.g., studio size and movie success) using statistical tests of independence that operate on data that can be represented by a single table. Many tests of association that assume that data instances are drawn from an iid population, even though data instances are actually drawn from sets of relational data. For example, the subjects of medical studies are related by their neighborhoods, hospitals, or workplaces. The undergraduate subjects in social psychology studies are often related by their courses, majors, or dormitories.

Propositionalization, sometimes called flattening, transforms data from a relational representation into a propositional one. Many relational learning algorithms incorporate propositionalization either as a pre-processing step or as an integral part of their search algorithms [50]. For instance, logic-based systems such as the relational subgroup discovery algorithm [54] or the LINUS system [51] preprocess a single data table using predicates defined in first-order logic. The ACORA system [67] utilizes a rich set of relational aggregations to construct a propositional feature vector that is then fed into a conventional learning algorithm. Other algorithms for learning probabilistic relational models [23], relational probability trees [63], relational Bayesian classifiers [64], or structural logistic regression [68] propositionalize on the fly as they search over the space of structural relationships between attributes. In all cases, the

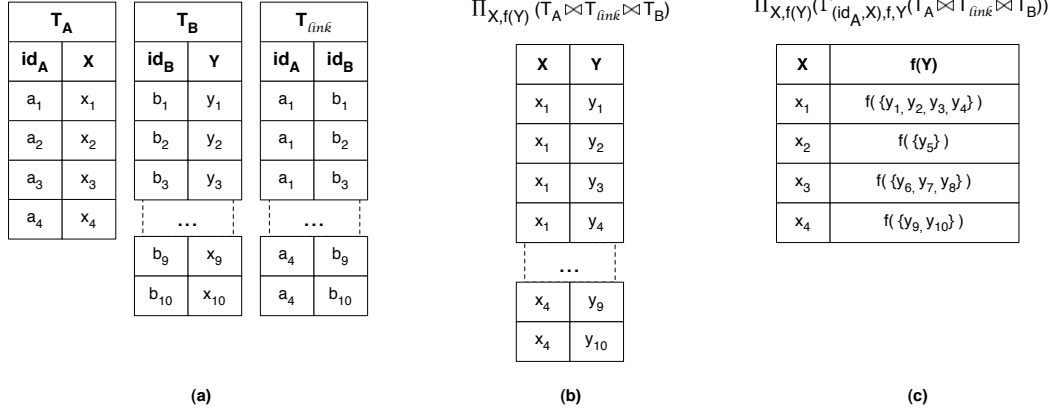


Figure 2.1: Relational database tables illustrating propositionalization operations (a). Replication (b) is the result of a three-way `INNER JOIN` of  $T_A$ ,  $T_B$  and  $T_{link}$ . Aggregation (c) is the result of a `GROUP BY` applied to the same join used in conjunction with an aggregation function  $f()$ . Common functions include `SUM`, `MAX`, `MIN`, and `AVG`.

learning algorithms propositionalize relational data prior to testing for marginal or conditional independence among sets of variables.

## 2.1 Propositionalization defined

Formally, we define a propositionalization as a mapping from a data graph  $G_d$  to a set of attribute vectors  $W$ . Below, we define several terms that will be useful in our discussion and analysis.

**Definition 1.** Given a relational data graph  $G_d = \{E, R, A\}$ , we say that  $S_i \subseteq E$  is an *instance subgraph* of  $G_d$  if  $\forall u \in S_i, \exists v \in S_i$  such that  $(u, v) \in R$ .

**Definition 2.** Let  $G_d = \{E, R, A\}$  be a relational data graph. An *attribute mapping*  $F$  is a vector of set functions  $[f_0, f_1, \dots, f_k]$  such that the domain of each  $f_i$  is a multiset of values from the range of some attribute  $A_i \in A$ .

**Definition 3.** Let  $S_i$  be an instance subgraph of a relational data graph  $G_d = \{E, R, A\}$ , and  $F = [f_0, f_1, \dots, f_k]$  be an attribute mapping of  $G_d$ . Let  $A_x(S_i)$  designate the mul-

tiset of values of attribute  $A_x$  associated with the entities in  $S_i$ . An **instance vector** is a  $1 \times k$  vector of values  $W_i = [w_0, w_1, \dots, w_k]$  where  $w_k = f_k(A_k(S_i))$ .

**Definition 4.** Given a relational data graph  $G_d$ , a set of instance subgraphs  $S$ , and an attribute mapping  $F$ . We define a **propositionalization mapping**  $P : G_d, S, F \mapsto W$  as a function mapping  $G_d$  to the set of instance vectors constructed by applying  $F$  to the instance subgraphs in  $S$ . Furthermore, we say that  $W$  is a **propositionalization** of  $G_d$  if  $W = P(G_d, S, F)$  for some  $P, S$ , and  $F$ .

Traditionally, propositionalization is defined in terms of functional transformations of record sets using relational algebra or SQL operations. Below, we briefly review the propositionalization process using the standard terminology. In addition, we present a novel definition of propositionalization as a graph sampling operating on the ground graph. We will demonstrate that while the two definitions are equivalent, the latter approach is especially useful for examining the validity of samples obtained after propositionalizing data that are not iid.

### 2.1.1 Algebraic approach to propositionalization

Two key operations for propositionalization are replication and aggregation. Propositionalizing simple relational data sets requires only one of these operations, while propositionalizing more complicated relational data may require several replication and aggregation steps. To better understand these operations, consider the bipartite data illustrated in Figure 1.10. Every entity of type  $A$  is related to several entities of type  $B$ . Each entity type has a single associated categorical variable ( $X$  and  $Y$ , respectively). Figure 2.1a depicts an alternative representation: three relational database tables corresponding to this data set. These include a table to store IDs and attributes of each entity type ( $T_A$  and  $T_B$ ), and one table to hold the relationship ( $T_{link}$ ).



Propositionalizing with replication can be illustrated with a two-column projection of a three-way inner join between  $T_A$ ,  $T_B$ , and  $T_{link}$ :

$$\Pi_{X,Y}(T_A \bowtie T_{link} \bowtie T_B)$$

A tabular illustration of this join can be seen in Figure 2.1b. In standard Structured Query Language (SQL), we would write:

```
SELECT ta.x, tb.y
FROM ta JOIN tlink ON ta.id=tlink.ida
      JOIN tb ON tlink.idb=tb.id
```

Here, each relation in the data set produces a tuple in the resulting table. Since nodes with degree greater than one (e.g.,  $A_1$ ) participate in several tuples, their attribute values (in this case,  $x_1$ ) are replicated in several rows.

Propositionalizing with aggregation can be illustrated with the same three-way inner join between  $T_A$ ,  $T_B$ , and  $T_{link}$ . However, in this case, multiple values of  $Y$  corresponding to a single entity  $A$  are aggregated (Figure 2.1c). The query uses an aggregation function  $f$  (e.g., SUM, AVG, MIN, or MAX) to operate over sets of values and produce a single value for the tuple. In SQL, a GROUP BY operator with a specified aggregation function or functions is applied to the same three-way join as above:

```
SELECT ta.x, f(tb.y)
FROM ta JOIN tlink ON ta.id=tlink.ida
      JOIN tb ON tlink.idb=tb.id
GROUP BY ta.id, ta.x
```

In our example, the  $X$  values of the group of  $B$  entities associated with each  $A$  entity produce a tuple in the target table, as seen in Figure 2.1. The above can be expressed in relational algebraic form<sup>1</sup>:

$$\Pi_{X,f(Y)}(\Gamma_{(id_A,X),f,Y}(T_A \bowtie T_{link} \bowtie T_B))$$

---

<sup>1</sup>Standard relational algebra lacks a grouping operator; here, we utilize the extended set of operators as defined by Grefen [32].

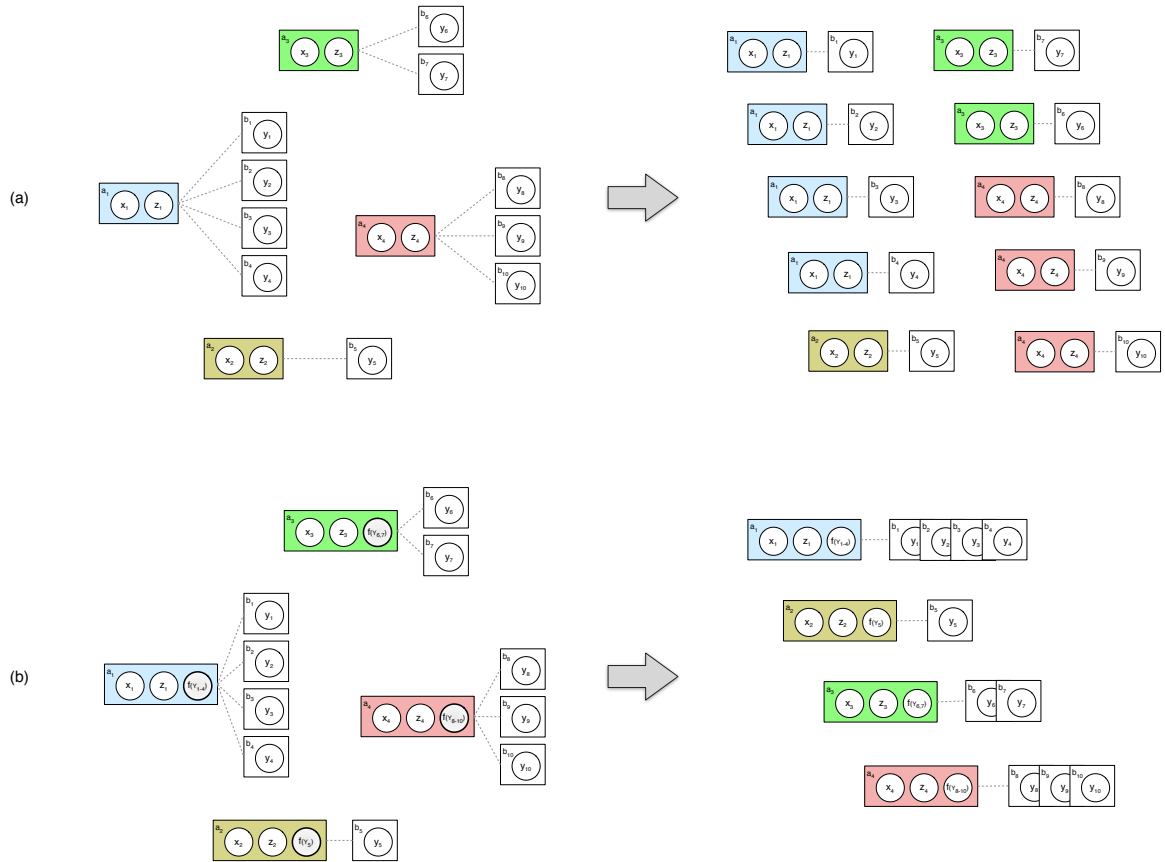


Figure 2.2: Propositionalization can be represented by subgraph sampling from the data graph. (a) Data graph representation matching the tables shown in Figure 2.1a. Propositionalization by replication is performed by drawing connected subgraphs (with replacement) from the data graph. For aggregation (b), the data graph is augmented with aggregated attributes, and subgraphs are sampled from the augmented graph.

Certainly, the algebraic SQL approach is the most common way for practitioners to process relational data for statistical analysis. However, as we will demonstrate, the relational algebra can be quite limiting when it comes to determining whether a given data set is iid or causally sufficient. Below, we present a novel framework for defining propositionalization in terms of graph sampling.

### 2.1.2 Graphical approach to propositionalization

Propositionalization can be defined as a graphical sampling procedure. Given a data graph, subgraph sampling can be used to create data instances. In accordance with Definition 1, only connected subgraphs are sampled, since the attribute values of entities not connected by a relational path are by definition not associated.

Figure 2.2 depicts the propositionalization procedure from Figure 2.1 graphically. In Figure 2.2a, we have a data graph with entity types  $A (a_1, \dots, a_4)$  and  $B (b_1, \dots, b_10)$ . Here, graphical sampling using replication produces ten total instances, one corresponding to each  $B$  entity. Since the resulting subgraphs share entities in the data graph, some entities are duplicated in the propositionalized sample (for instance, the entity  $a_1$  is replicated in four separate subgraphs).

In order to represent propositionalization by aggregation graphically, we perform a transformation on the data graph prior to sampling. We represent each variable aggregation with a new variable on the parent entities. When the transformed graph is sampled and subsequently analyzed, the unaggregated child attributes are ignored. Figure 2.2b depicts the augmented data graph with an extra variable on each  $A$  entity to represent aggregation  $f$  of the  $y$  values of the  $B$  entities. Note that the same variable can be aggregated using several functions (e.g. SUM, MAX, etc.), creating several new parent variables.

## 2.2 Propositionalization and instance dependence bias

Procedures for testing marginal and conditional independence are central to many algorithms for machine learning. For example, algorithms for learning the structure of Bayesian networks (e.g., PC) search over possible conditioning sets to identify pairs of variables that are conditionally independent [85, 90]. Algorithms that perform feature selection test whether a new feature is correlated with a dependent variable conditioned on the existing features [63]. Algorithms for learning association rules

evaluate whether new items are unexpectedly correlated with a target item conditioned on the existing items in the rule [83]. In each of these cases, assertions of marginal and conditional independence are one of the key statistical inferences made by the algorithm.

Unsurprisingly, inaccurate independence tests can cause serious errors in these algorithms. When tests incorrectly indicate independence, the algorithms disregard important predictive features, reducing the accuracy of learned models. When tests incorrectly infer dependence, algorithms add unnecessary structure to models, increasing the computational complexity of storing and employing those models. Finally, absent or superfluous statistical dependencies can cause a cascade of incorrect inferences in algorithms for learning model structure, particularly causal structure.

Prior research by Jensen and Neville has demonstrated that when the underlying generative process for the data contains relational dependencies—statistical influences that cross the boundaries of individual entities such that the variables of related entities are correlated—conventional tests of independence may be inaccurate [38, 41]. Common domains that exhibit relational dependencies include social networks (the attributes of one person can affect the attributes of their friends), organizational networks (the attributes of an organization can affect the attributes of its members), and web pages.

The errors described by Jensen and Neville have their origins in the mismatch between two data representations: the relational representation of the original data and the propositional representation required by a conventional test of independence. A propositional representation carries the assumption that each data instance can be represented solely by a vector of values, and that these instance vectors are iid.

Relational representations often include multiple entity types and explicitly represent relationships among instances, and as a result are not iid. When the data are propositionalized, the resulting data set may not be iid. Jensen and Neville [38] show

that statistical tests that assume independence among data instances are strongly biased if the original relational data graph exhibits one-to-many relationships and strong autocorrelation.

This closely parallels a long history of work in social science that has demonstrated errors in independence tests when the propositional data are not iid due to social groups or spatio-temporal relationships [46]. In 1889, Sir Francis Galton criticized some findings by Sir Edward Tylor by pointing out that many of the units being measured (societies) were not independent. From the proceedings of the Royal Anthropological Institute:

It was extremely desirable for the sake of those who may wish to study the evidence for Dr. Tylor’s conclusions, that full information should be given as to the degree in which the customs of the tribes and races which are compared together are independent. It might be, that some of the tribes had derived them from a common source, so that they were duplicate copies of the same original.

While Galton’s ideas were not expressed in terms of graphs, the issues raised are familiar. In this context, the tribes or societies being studied are theorized to have descended from the same larger group. As a result, they should not be considered independent instances; moreover, doing so may bias findings. In modern social sciences literature, the name “Galton’s problem” is used to denote the phenomenon of “group effects” causing instance dependence and elevating Type I error [8, 18, 47].

### **2.2.1 Graphical analysis of iid**

Although Jensen and Neville demonstrated that using independence tests naively can produce serious errors, the work produced neither a clear theoretical framework for analyzing those errors nor efficient methods for correcting them. Description of these errors has been informal and based largely on examples, identifying sufficient

conditions for the effects to occur, but not delineating the full range of situations that can produce the observed errors.

Here, we show how to use graphical representations to examine the effects of propositionalization. While equivalent to the more traditional algebraic approach, the graphical representation will be helpful for examining the independence characteristics of data after propositionalization and their effects on the validity of statistical tests. Violations of causal assumptions are sometimes hidden by the more compact DAPER representation, and application of the rules of d-separation to the ground graph will be useful when evaluating samples produced through propositionalization. Below, we outline in graphical terms the necessary conditions for a set of instances to be considered iid. In general, given the Markov assumption of the graph graph, propositionalization subgraphs will be iid if they do not overlap, and when there is no path between instances in the ground graph from which they are drawn.

Formally, for a propositionalized data set to be iid, it must consist of an *iid propositionalization* of the data graph.

**Definition 5.** *Let  $W$  be a propositionalization of some relational data graph  $G_d$  with associated ground graph  $G_g$ .  $W$  is said to be an **iid propositionalization** of  $G_d$  if and only if for all instance vectors  $W_i, W_j \in W$ , for all attributes values  $w_{ia} \in W_i, w_{jb} \in W_j$ , we have  $w_{ia} \perp\!\!\!\perp w_{jb}$  given  $G_g$ .*

Intuitively, for a propositionalization to be independent, there must be no dependence between instance vectors. Note that each element of the instance vectors may be calculated from several attribute values in the data graph, and that two vector elements will only be independent if all elements of these sets are independent between instances.

Below, we will examine a graphical representation of iid data using an illustrative example from the academic publishing domain with three entity types: journals, papers, and authors. For this example, each journal publishes several papers (one-

to-many), and each paper is authored by multiple authors (many-to-many). Journal entities have two attributes: *format* ( $F$ ), which determines the length and number of pages of a typical paper, and *prestige* ( $P$ ), which measures the impact and notoriety of the journal. Papers also have two attributes: *length* ( $L$ ), and *citation count* ( $C$ ). Finally, author entities have a single attribute representing their level of *happiness* ( $H$ ). An ER diagram and small data graph for this domain can be found in Figure 2.3.

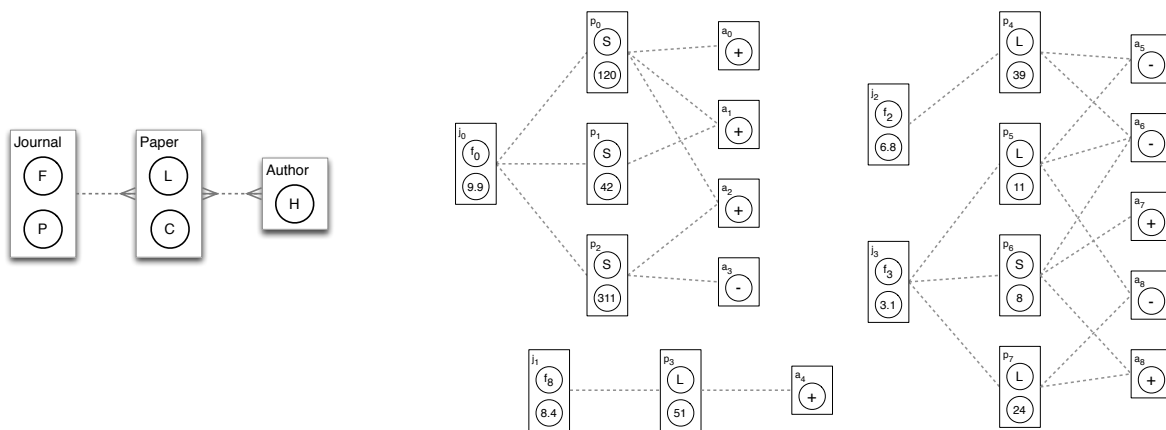


Figure 2.3: ER diagram (left) and data graph for academic publishing example. Each journal entity is connected to one or more paper entities, which are in turn related to several author entities. Journals have attributes for format ( $F$ ), prestige ( $P$ ); papers have attributes for length ( $L$ ) and citation count ( $C$ ); authors have a single attribute that measures their happiness ( $H$ ).

Below, we detail the two necessary and collectively sufficient graphical conditions that a propositionalized sample must meet to produce an iid data set: Subgraph samples must be disjoint, and the attribute values of each instance must be d-separated from all others. Note that the graphical approach to propositionalization presented here is an analytical framework and does not change the operational retrieval of data. In order to ground this work in common practice, we provide SQL code compatible with a typical RDBMS. Finally, we postpone a discussion of the precise statistical consequences of a violation of the iid assumption until the following chapter; for the

time being, we will focus on when and how propositionalization produces data that are iid, and how this process can be represented graphically.

**Condition 1: Instance subgraphs must not overlap**

Depending on the unit of interest and method of propositionalization (replication or aggregation), propositionalized instances drawn from the data graph may overlap. Formally, given a ground graph  $G_g$  and a propositionalization created from a

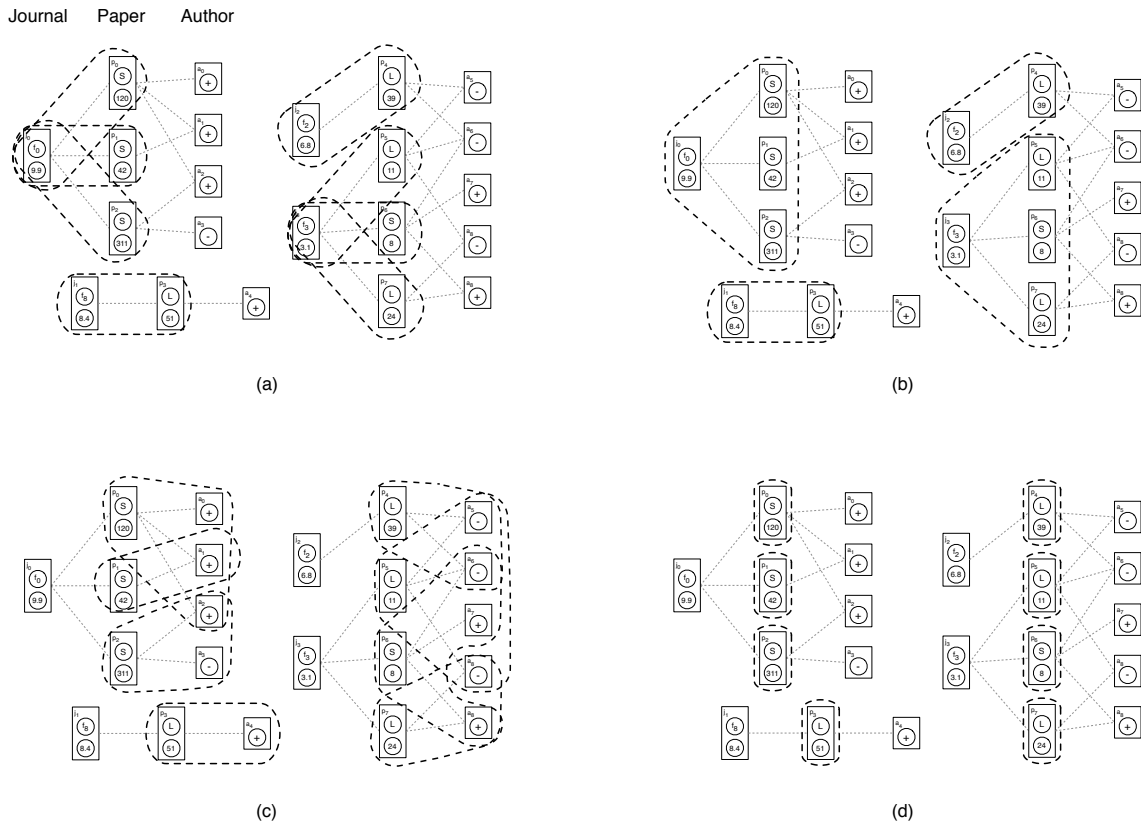


Figure 2.4: Graphical depiction of propositionalization for the academic publishing domain. Propositionalizing journal-paper using replication produces overlapping instances due to shared journal entities (a). By aggregating instead of replicating, the overlapping problem is avoided (b); however, aggregated subgraphs will overlap when data are related in a many-to-many manner (c). Single-entity instances (papers) will never overlap (d).



set of subgraph instances  $S$ ,  $\forall S_i, S_j \in S, S_i \cap S_j = \emptyset$ . For example, in order to test dependence between journal prestige  $P$  and paper citations  $C$ , one might select journal-paper pairs using replication as illustrated in Figure 2.4a. In SQL:

```
SELECT journal.prestige, paper.citations
FROM journal JOIN paper ON journal.id=paper.journal_id
```

Clearly, these instances are not iid; since journal entries are replicated over multiple instances, they are not independent. By aggregating rather than replicating (allowing us to examine the relationship between  $P$  and, for example,  $AVG(C)$ ), we can avoid the issue of having overlapping subgraphs (albeit at the expense of sample size) as shown in Figure 2.4b (note that in the figure, we have omitted the constructed aggregation variables for clarity). Recall that in SQL, aggregation requires the addition of a `GROUP BY` clause:

```
SELECT journal.prestige, f(paper.citations)
FROM journal JOIN paper ON journal.id=paper.journal_id
GROUP BY journal.id
```

While aggregation solves the overlapping problem for one-to-many relationships such as journal-paper, it will not do so for many-to-many relationships such as paper-author, as depicted in Figure 2.4c. The SQL commands for selecting papers and authors will be syntactically similar to the one shown above for journals and papers. However, in the latter case, the output will not be iid. While the graphical depiction makes the difference between these scenarios plain, the query itself offers no indication of a possible threat to validity.

Of course, by focusing on a single entity type, we can be sure that no subgraphs will overlap. For instance, to study the effects of paper length on citation count, a practitioner might sample only paper entities as instances, as in Figure 2.4d:

```
SELECT length, citations FROM paper)
```

Here, as in (b), the propositionalized instances appear relatively independent, as they do not share entities between them. However, before we can be sure that they are iid, we must examine the attribute dependencies of our sample.

**Condition 2: Instance subgraph attributes must be d-separated**

The lack of overlapping instances as depicted in Figures 2.4b and c is not enough to guarantee an iid sample. In addition, all attribute values of interest must be d-separated in the ground graph, rendering our instances independent. That is, for ground graph  $G_g$  and propositionalization instance vectors  $W, \forall W_i, W_j \in W$ , there does not exist a d-connecting path  $U = s, \dots, t$  in  $G_g$  such that for some  $w_{ia} \in W_i$  and  $w_{jb} \in W_j$ ,  $w_{ia}$  is a function of  $s$  and  $w_{jb}$  is a function of  $t$ . For brevity, when this is the case we will say that  $W_i$  and  $W_j$  are d-separated in  $G_g$ .

Thus, instance independence relies on *both* relational structure *and* attribute dependence together. In some cases, the relational structure alone can guarantee that this condition is met; since relations are the sole conduit for attribute dependence, domains in which no relationships exist between instances (e.g., the data shown in Figure 1.10c) will trivially produce d-separated subgraph samples.

Figure 2.5 depicts DAPER representations and ground graphs for the same relational data structure and unit of interest (papers) under two different generative models. In 2.5a, the existence of d-connecting paths between instances (e.g.,  $p_0.L(S) \leftarrow j_0.F(f_0) \rightarrow p_1.L(S)$ ) indicates that the instances are not iid. In 2.5b, we have the same relational structure, but there is no such path connecting instances, so our sample is valid, at least in the marginal case. The results for propositionalizing journal-paper instances like the ones in Figure 2.4b are similar.

Under both models, conditioning on *Author.H* will d-connect the paper instances through author objects by activating the path through a collider. While this effect is clearly illustrated by the graphical representation of propositionalization, the

traditional algebraic approach may hide it. Consider, for instance, the case where a practitioner chooses to limit her focus to happy professors. Algebraically, this is accomplished through the use of an additional join and WHERE clause:

```
SELECT paper.length, paper.citations)
FROM paper JOIN author_paper ON paper.id=author_paper.paper_id
      JOIN author ON author_paper.author_id=author.id
WHERE author.happiness=TRUE
```

When used in this manner, an SQL WHERE clause is equivalent to conditioning, which can radically alter the independence relationships found in data by enabling paths through colliders. While most practitioners are aware that selecting a subset of their data will possibly affect the generalizability of their conclusions, the algebraic approach provides no way of knowing that the tuples drawn from the database are no longer iid and a threat to statistical conclusion validity.

**Theorem 2.2.1.** *Given a relational data graph  $G_d$ , associated ground graph  $G_g$ , attribute mapping  $F$ , instance subgraphs  $S$ , and their corresponding instance vectors  $W$ . If  $S$  consists of non-overlapping instances and  $\forall W_i, W_j \in W$ ,  $W_i$  and  $W_j$  are d-separated, then  $W$  is an iid propositionalization of  $G_g$ .*

*Proof.* The proof is trivial by contradiction. Assume that for some ground graph  $G_g$ , we have a set of non-overlapping, d-separated instances  $W$  such that  $W$  is not an iid propositionalization of  $G_g$ . By the above definition, there must exist some pair of instances  $W_i$  and  $W_j$ , such that for some  $w_{ia} \in R_i, w_{jb} \in W_j$ ,  $w_{ia} \not\perp w_{jb}$ . Therefore, for some attribute values  $x_a, y_b$  in  $G_g$  and set functions  $f_a, f_b \in F$ ,  $w_{ia} = f_a(*, x_a), w_{jb} = f_b(*, y_b)$  and  $x_a \not\perp y_b$ . Since  $S_i$  and  $S_j$  cannot overlap, we know that  $W_i \cap W_j = \emptyset$ , and therefore  $w_{ia} \neq w_{jb}, f_a(*, x_a) \neq f_b(*, y_b), x_a \neq y_b$ . Thus, by the semantics of d-separation,  $x_a$  and  $y_b$  must be d-connected in  $G_g$ , making  $w_{ia} \not\perp w_{jb}$ , a contradiction. □

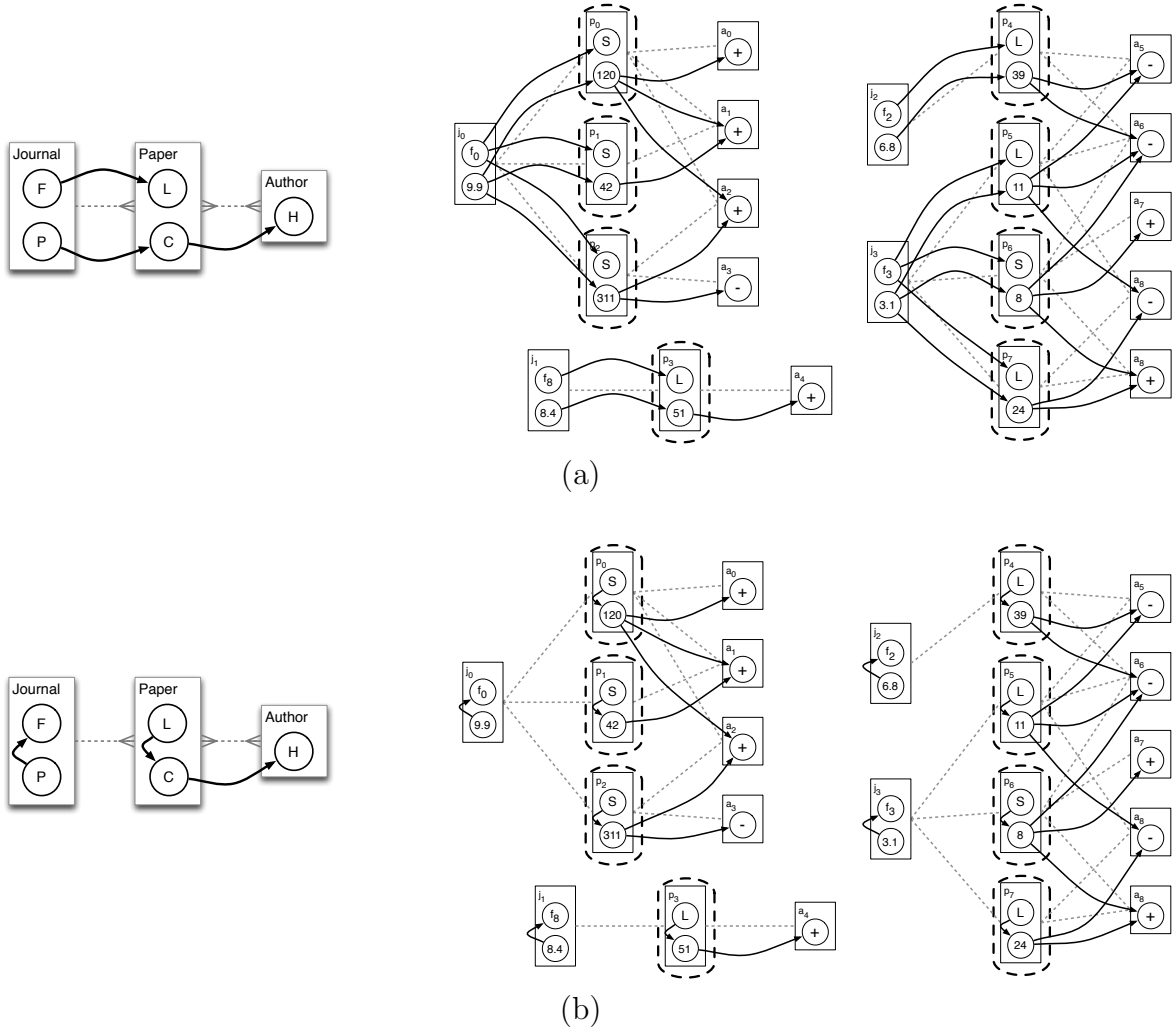


Figure 2.5: DAPER models and ground graphs for the propositionalized instance subgraphs shown in Figure 2.4c. (a) Under this generative model, instances are not independent due to the existence of d-connecting paths between the attributes of different instances in the ground graph. (b) Here, there are no such paths, so the set of paper instances will be independent. Note that if we were to condition on author happiness  $H$ , these instances would become dependent as well due to the activation of paths through author entities.

In essence, the two graphical conditions outlined above are describing the singular statistical requirement of row independence for the propositionalized data table. Child entity variables that are dependent on a common parent entity variable will be

dependent in the propositionalized table. When overlapping subgraphs are replicated, some values of the parent variables in the data table will exhibit perfect inter-instance dependence (they will necessarily have identical values). In both cases, the result is a set of instances that may exhibit instance dependence bias.

Condition 1 and Condition 2 are equivalent under some conditions, and we can illustrate this using a simple graph transformation. Figure 2.6 depicts a one-to-many subset of the publishing domain containing journals and papers. On the left, we have overlapping propositionalization subgraphs similar to those depicted in Figure 2.4. On the right, we have a transformed version of the same graph. Here, the prestige attribute on journals is first propagated to related paper entities. Since this new attribute is a copy, it is deterministically dependent on the parent attribute (recall that deterministic variables are depicted with a double circle). After this transformation, sampling paper entities alone (rather than journal-paper pairs) will capture the same attribute information, but the sampled subgraphs will no longer overlap. However, due to the deterministic dependence introduced by the transformation procedure, there will necessarily be a d-connecting path between the constructed attributes on papers.

Of course, given the causal Markov assumption of the ground graph, any sampled instances will be independent if their respective Markov blankets (parents, children, and other parents of children) do not overlap and are disconnected in the data graph. While valid, this condition is much more restrictive than the two outlined above. Additionally, in some cases, an examination of the DAPER representation alone will be sufficient to establish that propositionalization will result in a valid, iid sample. For example, data sets where relationships between entities are one-to-one will always produce iid samples. In the case of one-to-many data sets, systems where no child-entity variables have incoming edges originating from parent-level variables will be

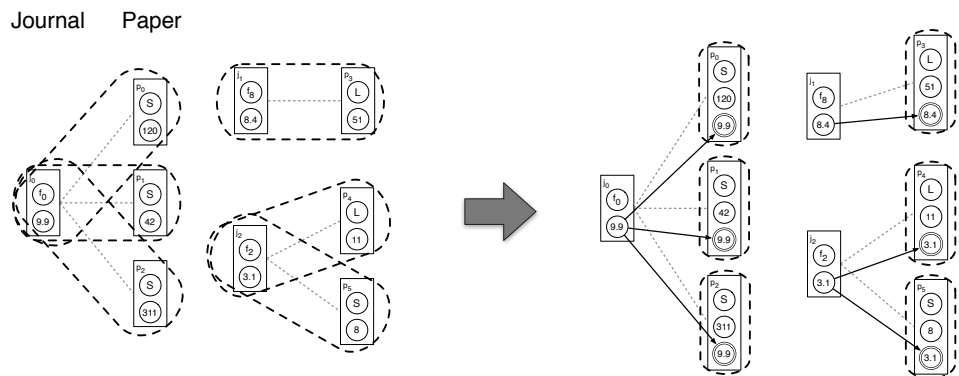


Figure 2.6: Equivalence of Condition 1 (non-overlapping subgraphs) and Condition 2 (d-separation) outlined above. By propagating attributes to related entities, we can avoid sampling overlapping subgraphs while still capturing the same attribute information. However, since propagated attributes necessarily introduce deterministic dependence with the source attribute, there will exist a d-connecting path between child entities.

iid when propositionalized, but those with parent-child attribute dependence may not be.

Previous work in statistical relational learning has addressed issues of instance dependence using algorithmic sampling. For example, Jensen and Neville utilize a resampling procedure to estimate relational feature variance [39], while Koerner and Wrobel present a method for generalized subgraph sampling in order to obtain unbiased training/test set splits for training models [49]. While these works do not consider in bias from the perspective of graphical models and conditional independence, the issues raised are similar to those presented here.

### 2.3 Aggregation and degree disparity

The alternative to replication is propositionalization through aggregation. While aggregation can avoid instance dependence for one-to-many domains, prior work has shown that aggregation can also lead to mistaken judgments of dependence. Often,

algorithms (and practitioners) make the implicit assumption that a measured dependence between a variable  $X$  and some aggregation of a variable  $Y$  is indicative of an underlying dependence between  $X$  and  $Y$ . However, Jensen, Neville, & Hay [41] show that this apparent attribute dependence can be the result of relational structure alone. In the presence of *degree disparity*, aggregation can make uncorrelated variables appear correlated when those data are propositionalized. Degree disparity occurs when an attribute on an entity is correlated with the number of relationships to or from that entity. For instance, chronologically older researchers tend to have authored more research papers, and persons from certain religious or ethnic backgrounds tend to have larger numbers of siblings. Again drawing from the movie domain, Jensen et al. showed that aggregations of randomly generated attributes on actors appear to be significant predictors of movie success, perhaps due to the fact that successful movies have (on average) higher numbers of actors listed in the IMDb.

Figure 2.7 depicts an ER diagram and data graph from the publishing domain (author entities are omitted here for simplicity). Here, each journal entity has a publication rate ( $R \in \{Yearly, Monthly\}$ ) attribute in addition to the format attribute  $F$ , and paper objects have a length  $L$  and citation count  $C$ . Perhaps a researcher wishes to identify which types of journals produce the most well-known papers for a given time period. He propositionalizes using aggregation as follows:

```
SELECT journal.rate, MAX(paper.citations)
FROM journal JOIN paper ON journal.id=paper.journal_id
GROUP BY journal.rate
```

This query produces the propositionalized data table depicted in the figure. Analyzing the statistical dependence in this table might (erroneously) lead to the conclusion that journals that publish yearly produce better papers than those that publish monthly. However, this observed association is due to the fact that yearly journals tend to publish more papers in a given issue than monthly ones, and that the MAX aggregator is quite sensitive to degree. Unlike the examples shown above, the subgraphs

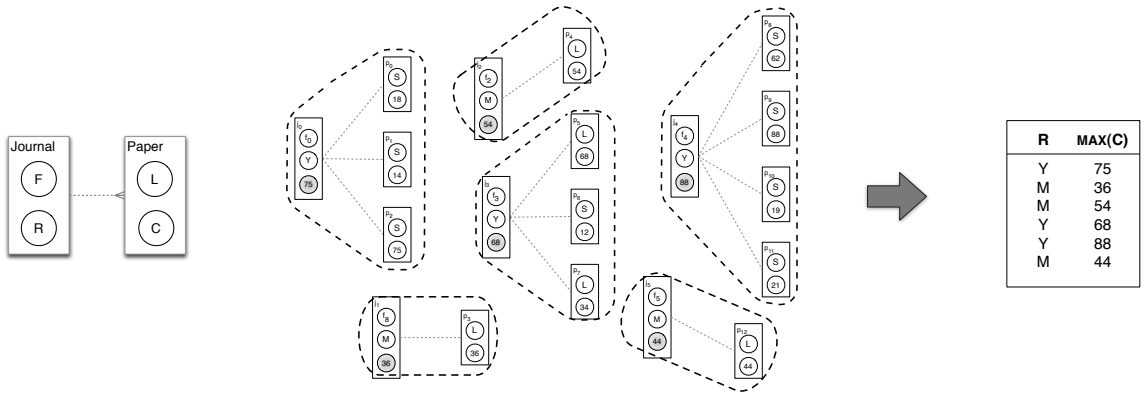


Figure 2.7: Degree disparity occurs when some attribute is associated with the number of relations for that entity. (Left) ER diagram for journals (with attributes format  $F$  and publication rate  $R$ ) and papers (with attributes length  $L$  and citations  $C$ ). (Center) data graph showing propositionalization subgraphs and constructed attribute using MAX aggregator (shaded). In this example, journal issues with yearly publication rates ( $R = Y$ ) tend to have more papers than those that publish monthly ( $R = M$ ). (Right) The data table produced by propositionalization through aggregation indicates an apparent relationship between  $R$  and  $\text{MAX}(C)$ . While there may be a direct dependence between  $R$  and  $C$ , the association may be due to the degree disparity with  $R$  combining with the sensitivity of the MAX aggregator to degree.

shown here are iid; however, the aggregation process may introduce a bias estimate of associations in the data. We postpone a more thorough, graphical treatment of degree disparity until Chapter 4, where we will examine the path of unexpected dependence on degree using d-separation. For the time being, it is important to note that the graphical consideration of the aggregation process makes clear the details of the relational structure that are lost using a purely algebraic, SQL-based approach.

## 2.4 Discussion

Although the above examples are based on relational data sets where the relations between entities are explicitly represented as a graph, the issues raised apply to *any* machine learning system that assumes an iid sample for statistical decision making.



The relational representation subsumes the traditional, propositional one, and makes explicit data interdependencies that cannot be traced once the data set has been converted to a single table. A lack of explicit relational information does not guarantee that data instances will be iid; rather, it indicates that we have no way of knowing whether the instances are iid.

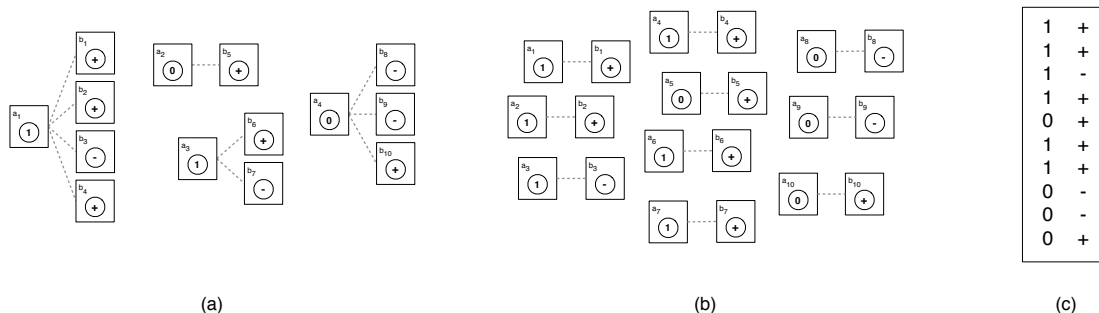


Figure 2.8: Differing data sets (a, b) may produce the same data table (c) when propositionalized. Naive learning systems that do not account for relations will mistakenly process the data in the same way, even though only one of the data sets comprises a valid iid sample when propositionalized.

Information about the relational structure of the data is lost during propositionalization. Differently structured relational data sets may produce the same propositionalized data tables, and statistics calculated on such tables will have the same values. For instance, when propositionalized using replication, the data sets depicted in Figure 2.8a and b will produce the same table of values. However, the validity of statistical inferences based on these values depends partially on this lost information.

The data in Figure 2.8b yield an iid sample, the data in Figure 2.8a do not. However, a naive learning system relying on an RDBMS for data storage will treat these two data sets identically; moreover, it may not even be able to tell the difference. For example, association rule learners [1, 59] are often used to discover frequent item sets from purchasing data. These algorithms assume that purchases are independent,

whereas a more sophisticated, relational system would take into account that multiple purchases by the same customer are heavily correlated.

In this chapter we have illustrated the propositionalization process in graphical terms, and highlighted some of the mechanisms that can lead to two types of bias associated with propositionalization: instance dependence and degree disparity. The former is due to a violation of assumptions in traditional statistical analysis, while the latter comes from an unrepresented interaction between attribute values, relational structure, and aggregation functions. In the following chapters, we examine the effects of instance dependence and degree disparity in more statistical detail. We show how the iid assumption (along with the causal Markov, causal sufficiency, and faithfulness assumptions) can be met for any data graph using targeted sampling or by incorporating structural variables. In addition, we demonstrate how to capture the effects of degree in a graphical model and how to make unbiased causal conclusions even when degree disparity is present.

## CHAPTER 3

# HYPOTHESIS TESTS FOR REPLICATED DATA WITH INSTANCE DEPENDENCE

As detailed in the previous chapter, propositionalization transforms data from a relational representation into a propositional one. Many relational learning algorithms incorporate propositionalization either as a pre-processing step or as an integral part of their search algorithms [50].

However, the loss of relational information that results from propositionalization can lead to inaccurate hypothesis tests. In this chapter, we examine those errors in more detail. Using the principles of d-separation [66], we show how several classes of generative models can produce the same observed correlations, and thus cause errors in algorithms that infer a specific generative structure from these correlations.

We present two solutions for conducting accurate hypothesis testing with non-iid data. The first, *link sampling*, transforms a bipartite graph in a manner that creates an iid sample once the graph is propositionalized [71]. Second, we show how to translate relational models into propositionalized models that capture key aspects of relational dependence in the form of structural attributes, and we use these enhanced models to perform novel hypothesis tests of conditional independence [74]. Using both real and synthetic data, we show how these tests allow algorithms to draw valid inferences despite conditions that mislead conventional tests.

### 3.1 Statistical consequences of instance dependence

Prior work has demonstrated that propositionalization with replication can lead to large increases in Type I errors (falsely inferring statistical dependence). Varieties

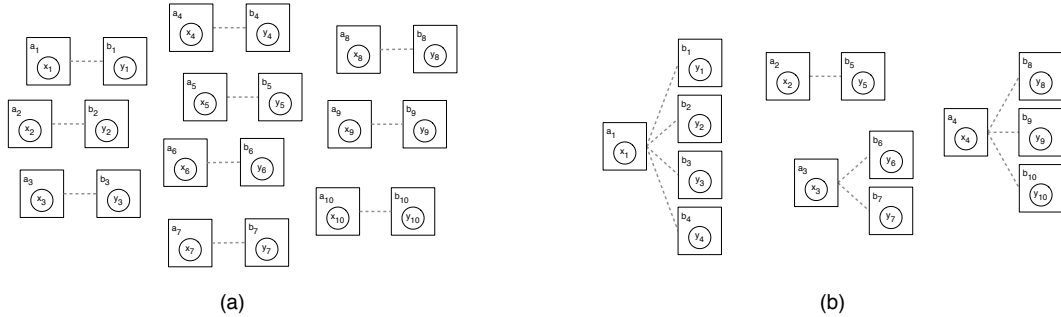


Figure 3.1: When applied to relational data, conventional statistical tests such as  $\chi^2$  implicitly assume one-to-one relationships (and therefore iid instances) such as those found in (a). When data are related in a one-to-many manner (b), the conventional reference distribution may be inappropriate due to the violation of the independence assumption.

of this effect have been known for more than a century [24], although the consequences of this effect for relational learning algorithms were first identified by Jensen & Neville [38]. In this work, the authors illustrate the effects of autocorrelation (non-independence of the values of a single variable across data instances) for relational domains that exhibit high concentrated linkage (one-to-many relationships with high cardinality). If both are present, even randomly generated attributes on parent entities may appear significantly associated with autocorrelated attributes on child entities.

When entities are replicated, attribute values of the tuples associated with each parent entity are perfectly autocorrelated; that is, there is a deterministic dependence between the parent attribute values in each entity-based group. For example, in the data graphs shown in Figure 3.1b, tuples representing subgraphs  $\{a_3, b_6\}$  and  $\{a_3, b_7\}$  will have identical values for attribute  $X$ . As we will see below, when there exists inter-instance dependence among the values for both attributes in a conditional independence test, the sampling distribution for a test statistic may differ substantially from the sampling distribution appropriate for an iid sample. Therefore, while repli-

cation does not always change the inference that should be drawn from a specific value of the test statistic, it sets the stage for possible errors by creating groups of instances with at least one attribute that exhibits inter-instance dependence.

Consider the effect of propositionalization with replication on the one-to-many data set shown in Figure 3.1b. Each row of the resulting 10-row propositionalized data set will contain a value for  $X$  and  $Y$ . Some instances will exhibit perfect dependence among some  $X$  values (since those multiple  $X$  values in the propositionalized data derive from the same  $X$  value in the relational data). For some ground graphs, dependence would also exist among the  $Y$  values (for example, due to a latent variable on  $A$  entities that causes values of  $Y$  on related  $B$  entities), and this would produce probabilistic inter-instance dependence among  $Y$  values in the propositionalized data. These twin violations of the iid assumption effectively reduce the sample size of a data set, increasing the variance of scores estimated using that set. This increased variability of the estimated value of any test statistic [38, 46] for one-to-many data results in Type I error rates much higher than those expected from independent instances such as those found 3.1a.

Figure 3.2 depicts the observed distribution of the chi-square statistic along with its Type I error rate for synthetic data containing two types of entities,  $A$  and  $B$ , each of which contains a single variable,  $X$  and  $Y$ , respectively. We generate 200  $A$  entities and link each to between 1 and 20  $B$  entities. The level of dependence is expressed as the probability that any two “sibling”  $B$  entities will share the same  $Y$  value, calculated from the class distribution of  $Y$  and a parameter governing the strength of effect (for a data set with no autocorrelation effect, this quantity is equal to  $p^2 + (1 - p)^2$  for a binary variable with class probabilities  $p$  and  $1 - p$ ). Here, for a simulation with an autocorrelation level of 0.8 (moderate effect, given an even class split), 38% of the data sets generated had a chi-square value that was statistically significant at the  $\alpha = 0.01$  level, quite a bit larger than the expected Type I error

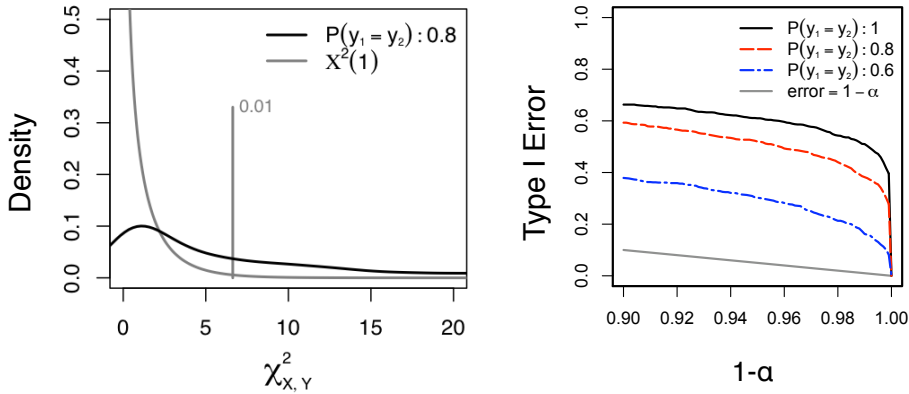


Figure 3.2: Values of the chi-square statistic are biased for data with instance dependence. (Left) The empirical distribution has much higher variance than the theoretical  $\chi^2$  with one degree of freedom. (Right) The Type I error rate greatly exceeds the expectation based on alpha; the bias becomes more severe for higher levels of dependence.

rate of 1%. As seen in the figure, the higher the level of dependence among  $Y$  values, the more severe the bias.

### 3.2 Understanding instance dependence bias with graphical models

The circumstances under which these errors arise can be illustrated using the ground graph. The problem stems from the fact that the set of instances being tested for dependence violates the iid assumption, resulting in increased Type I error. Here, the graphical interpretation of the propositionalization process is especially informative. Given a set of instances based on subgraph sampling from the data graph, each subgraph instance must be independent (and, therefore, its variables d-separated in the ground graph) from all others.

Figure 3.3 depicts several data scenarios for bipartite relational data. For each scenario, a DAPER model is shown alongside a ground graph that is appropriate to

that model, along with an empirically derived chi-square distribution for variables  $X$  and  $Y$  (each distribution was derived from 2000 synthetic data sets with 100  $A$  entities, 2500  $B$  entities, and discrete values  $X, Y \in [1, 10]$ ). In all cases, the data are propositionalized with replication (one instance per  $B$  entity). The instance dependence bias occurs when *both*  $X$  and  $Y$  values are autocorrelated, and therefore non-independent.

Scenario (a) is similar to the one outlined by Jensen and Neville [38]. Here,  $B.Y$  values exhibit autocorrelation resulting from a dependence on the latent variable  $A.Z$ , while  $A.X$  is generated independent from  $Y$ . When propositionalized, the  $X$  values in the resulting tuples are replicated, resulting in perfect autocorrelation (non-independence) among tuples sourced from the same  $A$  entity. Additionally, there are clear d-connecting path between several values of  $Y$  (e.g.,  $Y_1 \rightarrow Y_2$ ,  $Y_8 \rightarrow Y_{10}$ ). Not surprisingly, the empirically derived distribution for chi-square is heavily biased.

Case (b) is identical to case (a), but here the values of  $Y$  are not autocorrelated. While the values of  $X$  are still non-independent due to replication, there exist no d-connecting paths between  $Y$  values in the ground graph (in fact, there are no causal paths at all). As a result, the distribution of chi-square is not biased.

The third scenario (c) is the same as case (a) in terms of attribute dependence, but here the data are one-to-one rather than one-to-many. In this case, neither  $X$  values nor  $Y$  values are non-independent, since no replication for  $X$  takes place and no paths exist connecting  $Y$  values. Again, the empirically derived distribution is unbiased.

Scenario (d) depicts a case where  $B.Y$  is autocorrelated (stemming from latent variable  $A.Z$ ), but the  $X$  variable is associated with  $B$  entities. Like case (a), there are d-connecting paths between  $Y$  values; however, the propositionalization process does not replicate  $X$  values, so again there is no bias.

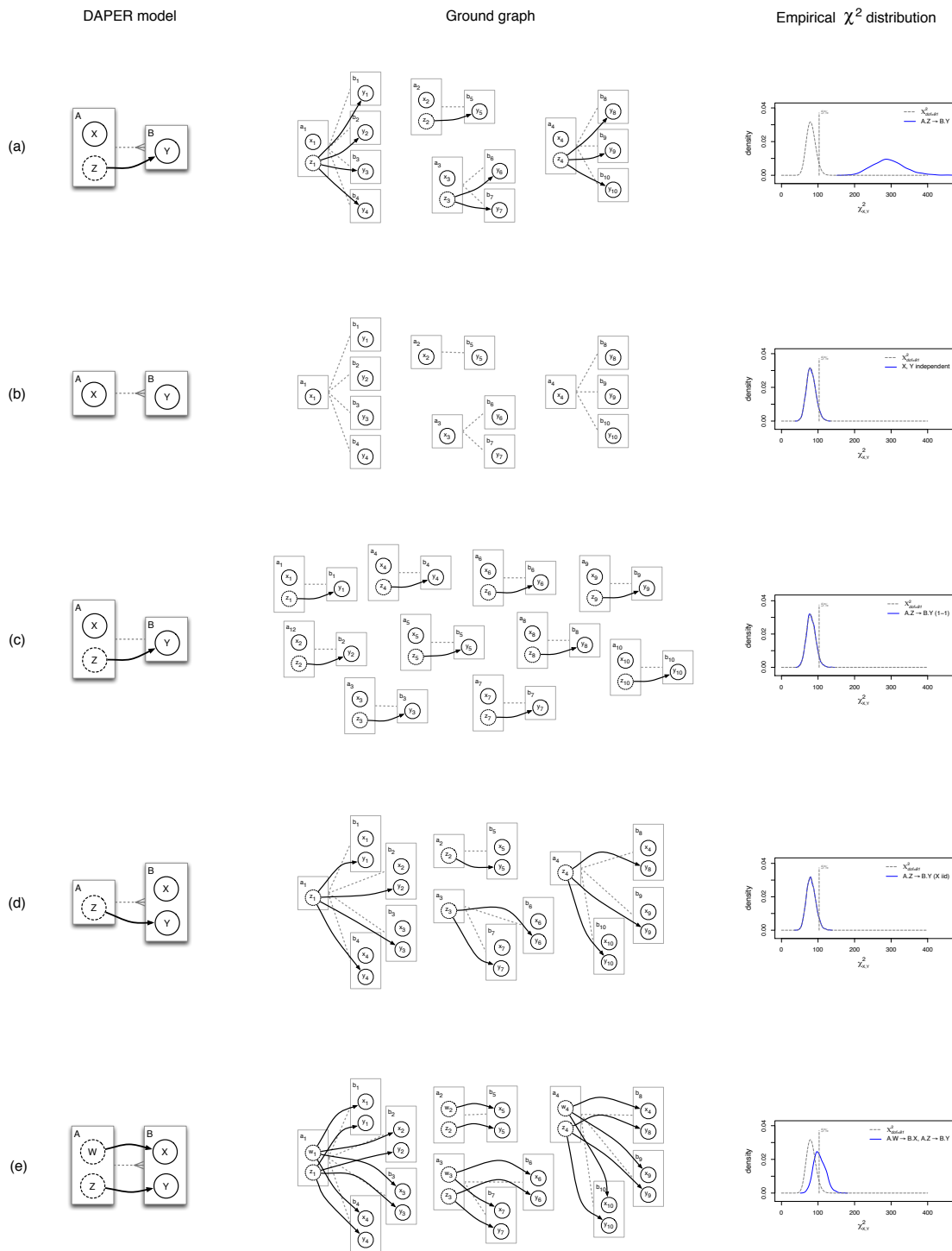


Figure 3.3: DAPER models (left), ground graphs (center), and distributions of the chi-square statistic (right) for different relational data sets. In scenarios (a) and (e), values of both  $X$  and  $Y$  are non-independent (and therefore not d-separated in the ground graph), resulting in a distribution of the test statistic that does not match the chi-square reference distribution.



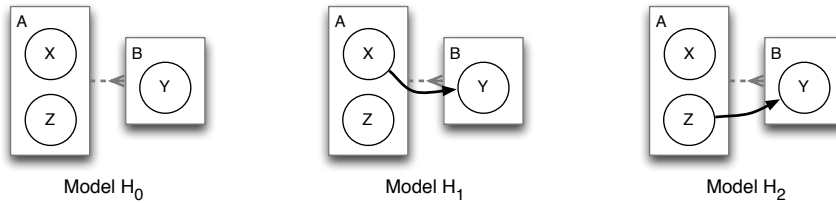


Figure 3.4: Three possible DAPER models for one-to-many data. In Model H<sub>0</sub>, variables  $X$  and  $Y$  are independent. In Model H<sub>1</sub>,  $X$  influences  $Y$ , while in H<sub>2</sub>,  $Y$  is independent of  $X$  but related to a latent variable  $Z$ . Data generated by H<sub>1</sub> and H<sub>2</sub> will exhibit instance dependence when propositionalized with replication.

Finally, case (e) depicts a scenario similar to case (d), except the  $B.X$  values are autocorrelated (via a second latent variable  $A.W$ ). As a result, there are disconnecting paths between both  $X$  and  $Y$  variables in the ground graph, resulting in a biased distribution of chi-square.

The situation we described informally in Section 3.1 can be described more formally by the DAPER models in Figure 3.4. Model H<sub>0</sub> corresponds to the null hypothesis that  $X$  and  $Y$  are marginally independent. Model H<sub>1</sub> indicates that  $X$  causes  $Y$ . Model H<sub>2</sub> indicates that  $Y$  is caused by a latent variable  $Z$  on the same entity as  $X$ , but otherwise is marginally independent of  $X$ . The values of  $Y$  on different entities  $B$  connected to the same  $A$  will be autocorrelated in either of the models H<sub>1</sub> or H<sub>2</sub>.<sup>1</sup>

As in the examples in Figure 3.3, model H<sub>2</sub> uses a common convention in graphical models to produce autocorrelation among related entities. The relational structure of the data indicates that a single entity  $A$  will be connected to several entities  $B$ . As a result, the dependence between a variable  $Z$  on  $A$  and several different instances of a variable  $Y$  on  $B$  will induce dependence among the values of  $Y$  on related entities  $B$ . This approach is often used in the social sciences to represent a “group effect” [47].

---

<sup>1</sup>The models in Figure 3.4 clearly do not exhaust the possible models that could relate these variables, but are meant to demonstrate that multiple generative models are consistent with the observed correlations. The full space of models is discussed in more detail in Section 6.2.

Models in machine learning frequently use this approach to model autocorrelation among members of latent groups [62] or among topics of related text documents [56, 76].

According to the theories advanced in prior work [38], independence tests will frequently indicate that  $X$  and  $Y$  are marginally dependent when those data are generated using either model  $H_1$  or model  $H_2$ . In general, given a significant value of a statistic alone, it is impossible to determine whether model  $H_1$  or  $H_2$  generated the data. Whether this distinction is important depends on the domain. However, if gaining a causal understanding is important, the distinction is crucial to determining whether manipulating  $X$  will change  $Y$  [80]. To address the issue of biased hypothesis tests for relational data, Jensen et al. presented a computationally intensive method to derive accurate reference distributions using randomization tests [42]. Below, we outline two new strategies for accurately assessing independence in relational data sets that are informed by the graphical models described above.

### 3.3 Link sampling

Link sampling is a novel technique for accurate hypothesis testing in bipartite relation data [71]. Rather than adjusting the reference distribution, link sampling works by modifying the calculation of the test statistic itself such that it will be correctly distributed with a  $X^2$  distribution. Recall that the problem identified in the previous section stemmed from non-independence between attribute values for the instances used to populate a contingency table. Using link sampling, we can “enforce” the independence assumption by constructing a contingency table out of an iid propositionalization of the relational data graph.

To select the set of instances for inclusion in the contingency table, we draw subgraphs from a modified version of the data graph. This modified data graph contains an identical set of objects and attributes as the original; however, the set of

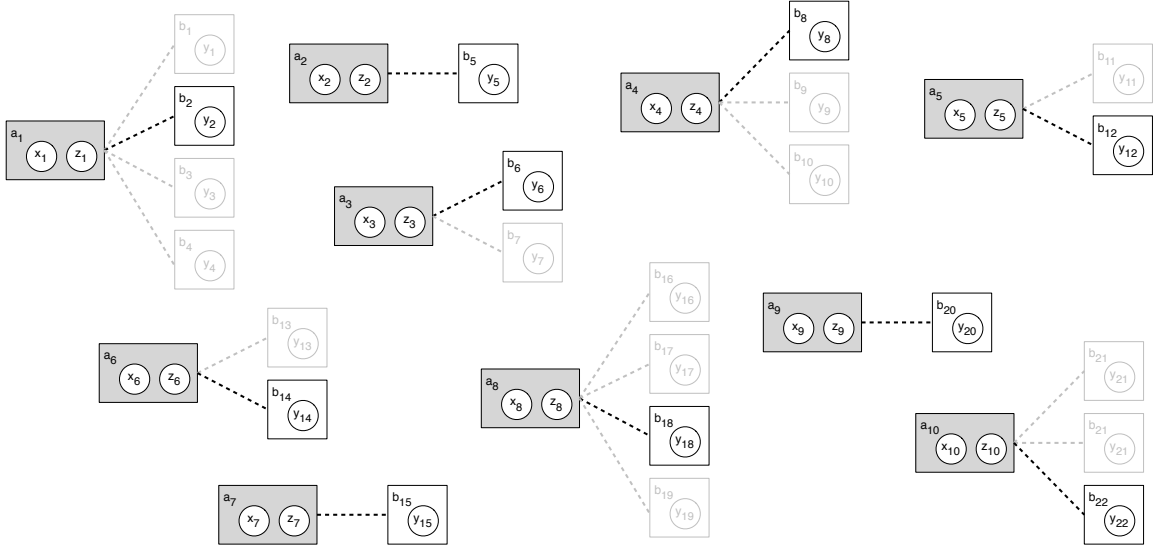


Figure 3.5: Link sampling modifies the data graph such that no entity has degree greater than one. While this reduces the data available to a hypothesis test, the transformed graph will produce an iid sample when propositionalized.

relations is a subset of those from the original graph, such that the relations included are a proper *matching* (a set of edges which share no common vertices). More formally, given a relational data graph  $G = (V, E, A)$ , we construct  $G' = (V, E', A)$  such that  $\forall s, t \in E', \nexists u, v \in E' : (s = u \wedge t \neq v) \vee (s \neq u \wedge t = v)$ .

To produce a link matching, we use the randomized greedy matching algorithm presented by Aronson et al.[4]. While the algorithm as presented seeks to find a maximal matching, it can be trivially adapted to select a set of independent links of a given target size (assuming that one exists).

Any propositionalization of the modified graph will be necessarily iid, as any set of instance subgraphs drawn from the graph will have no common neighbors (and therefore no d-connecting paths between them) in the associated ground graph. Since the independence assumption is no longer violated, the  $x, y$  pairs that fill the contingency table will be independent, and a  $\chi^2$  statistic calculated from the contingency table will be distributed with the theoretical  $X^2$ .

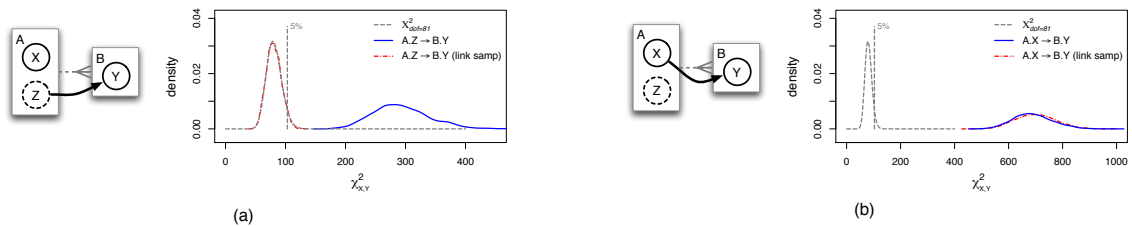


Figure 3.6: When generated from independent subgraphs through link sampling, the distribution of  $\chi^2$  closely matches the theoretical  $\chi^2$  distribution in cases where observed dependence is due to instance dependence bias (a). In cases where there is a direct dependence (b), the distribution is unaffected.

The effect of link sampling on the empirical distribution of chi-square is shown in Figure 3.6. In (a), we have empirical distribution of chi-square with and without link sampling for model  $H_2$ , the biased autocorrelation case. Here, the link sampling procedure removes the bias, and the statistic is distributed as with propositional data. Figure 3.6b shows the same distributions for the case where there is a direct dependence between  $X$  and  $Y$  (model  $H_1$ ). In this scenario, the distribution is unaffected by sampling (for legibility, the strength of effect used to generate plot (b) was greatly reduced).

The link sampling technique is applicable to any form of relational data, regardless of relational structure or attribute distribution and dependencies. Of course, the link sampling procedure greatly reduces the amount of data available to a statistical test. For bipartite, one-to-many data sets, the available sample size may be reduced by a factor equal to the average degree of the parent entities. Whether this reduction in available testing data negatively affects power is domain dependent; certainly, for suitably large networks (such as those used to produce the plot in Figure 3.6b) it is not an issue.

### 3.4 Novel hypothesis tests with *ID* variables

The graphical structure of model  $H_1$  in Figure 3.4 provides a clear indication of why  $X$  and  $Y$  are dependent in data drawn from this model, but model  $H_2$  does not provide any correspondingly clear indication. One reason for this is that the DAPER model represents data in its relational state, and the results discussed in Section 3.1 derive from propositionalized data. Propositionalization may introduce additional dependencies not explicit in the DAPER model. Note that this fact does not invalidate the claim made by Corollary 1.3.2, which states that d-separation in a DAPER model guarantees d-separation in the ground graph. Even under model  $H_2$ ,  $X$  and  $Y$  are d-separated in the ground graph; they only exhibit dependence in the replicated propositionalization of the ground graph.

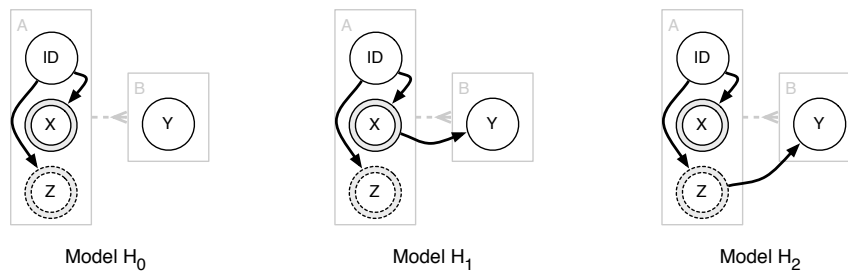


Figure 3.7: Propositionalized versions of generative models for non-iid data. The plate structure is included here for clarity only, and is not part of the graphical model.

Figure 3.7 shows propositionalized models corresponding to DAPER models  $H_1$  and  $H_2$ . The entity-relationship structure is shown in gray for reference only and is not part of the model. The propositionalized models introduce a new variable: *ID*. The *ID* variable models the replication of the values of the  $X$  and  $Z$  variables during propositionalization. In the same way that  $Z$  models the autocorrelation among values of  $Y$ , *ID* models the (perfect) autocorrelation among replicated values of  $X$  and  $Z$  variables.

The  $ID$  variable corresponds to the  $ID_A$  and  $ID_B$  columns in the relational database tables depicted in Figure 2.1. The value itself is arbitrary and has no intrinsic meaning; although frequently represented as an integer it is a categorical attribute. Such variables have unbounded cardinality; formally, the “support” of the variable’s distribution (the smallest closed set whose complement has probability zero) is of infinite size. Furthermore, it carries the constraint that no two entities in the relational data share the same value, although multiple data instances in propositional data can (and often do) have the same value of  $ID$ .

The  $ID$  variable deterministically causes every other variable whose values are replicated during propositionalization since information about an entity’s  $ID$  completely determines the value of any variable associated with that entity. Given this, the  $ID$  attribute is an example of an infinite latent variable as proposed by Xu [92] (only having perfect predictive ability), or a cluster identifier in the sense used by Kemp [45]. Leveraging  $IDs$  as variables has also been shown to improve inference in relational learning [67, 51].

Given the propositional models in Figure 3.7, the semantics of d-separation provides a formal explanation for the results from Section 3.1 [66]. In both models, the existence of an undirected collider-free path from  $X$  to  $Y$  corresponds to the observed correlations between the variables. In Model  $H_1$ , the path is direct; in Model  $H_2$ , the path flows from  $X \leftarrow id \rightarrow Z \rightarrow Y$ . We can block the causal path by conditioning on any of the variables along that path. Conditioning on  $ID$  will d-separate  $X$  and  $Y$  under Model  $H_2$  (but not Model  $H_1$ ), allowing us to differentiate between the two. However, this fact does not provide a feasible statistical test, since holding  $ID$  constant will also hold  $X$  constant.

Fortunately, this propositional model suggests another conditional independence test to differentiate Model  $H_1$  from Model  $H_2$ . If the data were generated by Model  $H_1$ , we would expect that  $ID \perp\!\!\!\perp Y \mid X$ . Figure 3.8 shows the empirical distributions of

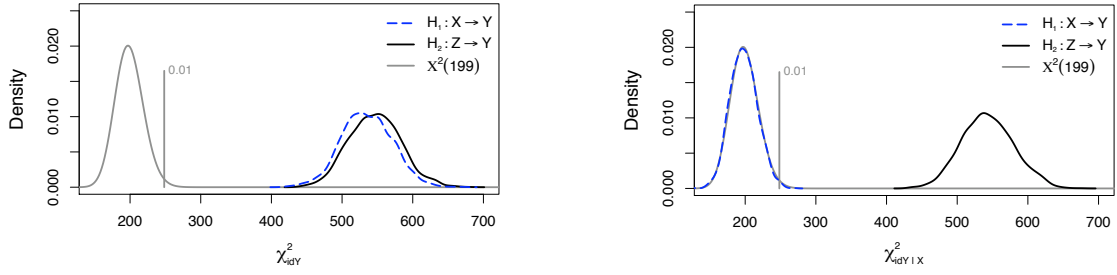


Figure 3.8: Empirical chi-square distributions for  $ID$ ,  $Y$ . (Left) Data generated under Model  $H_1$  is indistinguishable from data generated by Model  $H_2$  as both models create autocorrelation among  $Y$  values (captured here as an association between  $ID$  and  $Y$ ). (Right) The effect of conditioning on  $X$ , allowing clear discrimination between models.

$\chi^2_{ID-Y}$  when conditioned on  $X$ . The association between  $ID$  and  $Y$  disappears when we condition for Model  $H_1$ , allowing us to retain the null hypothesis. For data from Model  $H_2$ , conditioning on  $X$  does not diminish the value of  $\chi^2$ , allowing us to reject Model  $H_1$  in favor of Model  $H_2$ . Thus, even with a graphical model that relies on a latent variable ( $Z$ ), we have a test based only on measured variables that allows us to differentiate between the two models.

Even though the conditional test between  $ID$  and  $Y$  is being performed with a data table generated through a non-iid propositionalization, the test will be unbiased, as the data instances are *conditionally iid* given  $X$  under Model  $H_2$ .

**Definition 6.** Let  $W$  be a propositionalization of some relational data graph  $G_d = \{V, E, A\}$  with associated ground graph  $G_g$ .  $W$  is said to be a **conditionally iid propositionalization** of  $G_d$  if and only if  $\exists C \subseteq A$  such that  $\forall W_i, W_j \in W, \forall w_{ia} \in W_i, w_{jb} \in W_j, w_{ia} \perp w_{jb} \mid C$  in  $G_g$ .

For example, propositionalized data generated under Model  $H_1$  is conditionally iid given  $X$  or  $ID$ , while propositionalized data for Model  $H_2$  is conditionally iid given  $Z$  or  $ID$ . Given an iid propositionalization, we can use conditioning to obtain an iid

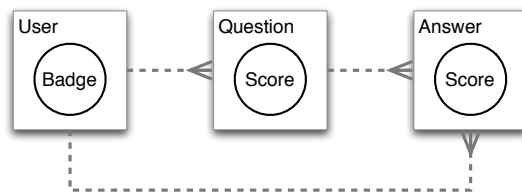


Figure 3.9: ER diagram describing the Stack Overflow data set. Users post questions as well as answer questions from others. Users are awarded badges, while both questions and answers are given scores based on the number of up and down votes.

set of instances, allowing us to perform hypothesis tests that are free from instance dependence bias.

### 3.5 Empirical results

The Stack Overflow data consists of *users*, *questions*, and *answers*. Users may post new questions or provide answers to existing ones, and may *vote* (up or down) on the quality of both questions and answers posted by others. Furthermore, as users use the system, they are awarded *badges* designating some accomplishment. For example, the “Fanatic” badge is awarded to users who visit the site for a hundred days in a row, while the “Guru” badge is given to users who provide an answer that receives forty or more votes. An ER diagram describing a simple schema for Stack Overflow can be seen in Figure 3.9.

We examined the relationship between badge acquisition and answer *score* (up-votes minus down-votes). The dataset records activity on the site between February 1 and April 1, 2010. During this time period, there were 237,505 answers provided by 61,625 distinct users. For each of the 43 badge types, we generated a binary attribute on each user designating whether that badge had been awarded *before* April 1.

In the Stack Overflow data set, answer scores are heavily autocorrelated through user; that is, users are fairly consistent in the quality of the posts they provide



(the Pearson corrected contingency coefficient is 0.75). Thus, in the replication case, every single badge attribute appeared to be correlated with a discretized answer score when naively tested. However, as discussed in Section 3.1, the marginal dependence between badges and score can be explained by different causal mechanisms as depicted in Figure 3.10. Again, we emphasize that the causal mechanisms in Figure 3.10 do not exhaust the possible mechanisms. Rather, we use these two specific models to demonstrate the utility of the tests.

Using the *ID* of each user, we can differentiate between model  $H_1$  and  $H_2$  by performing a hypothesis test on *User.ID* and *Answer.score* conditioned on *User.badge*. The results of these tests are depicted in Table 3.1. In 22 of the 43 cases (shown in bold), the value of chi-square in the conditional test is not significant, allowing us to conclude that the relationship between that badge and answer score is not, in fact, causal, and that the marginal relationship is due to some other factor associated with users (model  $H_2$ ). For the other badges, the conditional test does produce a significant value, suggesting a causal relationship (model  $H_1$ ) in the form of a direct



Figure 3.10: Alternative models that explain the marginal dependence between Stack Overflow badge awards on user and scores of their answers. In (a), badge awards have a direct influence on answer score; in (b), the perceived dependence is due to instance dependence bias brought on by a hidden factor  $H$ . The two models can be differentiated by performing a conditional hypothesis test on *ID* and *Score* conditioned on *Badge*.

edge from *User.badge* to *Answer.score*. Of course, this conclusion is only valid if we are certain that there does not exist latent confounders associated with an entity other than users or answers.

### 3.6 Conclusion

In the previous chapter, we described the ways in which relational data can be transformed into propositional form, and how entity replication may produce data sets that are not iid. In this chapter, we demonstrated the statistical ramifications of naive hypothesis testing with replicated data. Any time entities that are related in a one-to-many or many-to-many manner are replicated, we introduce a threat to statistical conclusion validity in the form of instance dependence bias. To address this issue, we have introduced the formal framework of the ground graph, as well as two novel techniques for overcoming instance dependence bias. In the first, we sample from the data graph in order to produce a propositionalized data table that is guaranteed to be iid. In the second, we construct *ID* variables to capture relational structure, and show how to utilize these variables for unbiased hypothesis testing. Of course, aggregating rather than replicating, we can (at least in the case of one-to-many data) avoid the issue of dependent instances altogether. However, as we shall see in the following chapter, aggregation may introduce a different type of statistical bias stemming from a common interaction between relational structure and attribute values.

Table 3.1: Results of marginal ( $badge, score$ ) and conditional ( $ID, score \mid badge$ ) hypothesis tests for Stack Overflow (replication). Boldface indicates statistical significance, while italicized text highlights cases where associations are judged not causal due to a lack of significance in the conditional test.

<b>Badge</b>	$\chi^2_{badge, score}$	<b>p-value</b>	$\chi^2_{ID, score \mid badge}$	<b>p-value</b>
<i>Autobiographer</i>	<b>4163</b>	0.0000	<i>221663</i>	<i>0.9627</i>
Beta	<b>366</b>	0.0000	<b>228923</b>	0.0000
<i>Citizen Patrol</i>	<i>10232</i>	<i>0.0000</i>	<i>219673</i>	<i>1.0000</i>
<i>Civic Duty</i>	<i>11412</i>	<i>0.0000</i>	<i>218641</i>	<i>1.0000</i>
<i>Cleanup</i>	<i>7635</i>	<i>0.0000</i>	<i>222220</i>	<i>0.8280</i>
<i>Commentator</i>	<i>10894</i>	<i>0.0000</i>	<i>219617</i>	<i>1.0000</i>
<i>Critic</i>	<i>12200</i>	<i>0.0000</i>	<i>218073</i>	<i>1.0000</i>
<i>Disciplined</i>	<i>9723</i>	<i>0.0000</i>	<i>220198</i>	<i>1.0000</i>
<i>Editor</i>	<i>9768</i>	<i>0.0000</i>	<i>218850</i>	<i>1.0000</i>
Electorate	<b>924</b>	0.0000	<b>228595</b>	0.0000
<i>Enlightened</i>	<i>15346</i>	<i>0.0000</i>	<i>216007</i>	<i>1.0000</i>
<i>Enthusiast</i>	<i>12041</i>	<i>0.0000</i>	<i>217621</i>	<i>1.0000</i>
Epic	<b>5007</b>	0.0000	<b>224674</b>	0.0032
Famous Question	<b>1541</b>	0.0000	<b>228424</b>	0.0000
<i>Fanatic</i>	<i>6167</i>	<i>0.0000</i>	<i>222709</i>	<i>0.5843</i>
Favorite Question	<b>1922</b>	0.0000	<b>227859</b>	0.0000
<i>Good Answer</i>	<i>10325</i>	<i>0.0000</i>	<i>219223</i>	<i>1.0000</i>
Good Question	<b>2886</b>	0.0000	<b>227058</b>	0.0000
Great Answer	<b>3187</b>	0.0000	<b>226596</b>	0.0000
Great Question	<b>1462</b>	0.0000	<b>228523</b>	0.0000
<i>Guru</i>	<i>5940</i>	<i>0.0000</i>	<i>223200</i>	<i>0.3007</i>
Legendary	<b>3195</b>	0.0000	<b>226949</b>	0.0000
<i>Mortarboard</i>	<i>15209</i>	<i>0.0000</i>	<i>217677</i>	<i>1.0000</i>
Necromancer	<b>3260</b>	0.0000	<b>225250</b>	0.0002
<i>Nice Answer</i>	<i>14769</i>	<i>0.0000</i>	<i>215927</i>	<i>1.0000</i>
Nice Question	<b>4332</b>	0.0000	<b>225250</b>	0.0002
Notable Question	<b>1280</b>	0.0000	<b>227932</b>	0.0000
<i>Organizer</i>	<i>13232</i>	<i>0.0000</i>	<i>217462</i>	<i>1.0000</i>
Peer Pressure	<b>2517</b>	0.0000	<b>227307</b>	0.0000
Popular Question	<b>1933</b>	0.0000	<b>225966</b>	0.0000
Populist	<b>4514</b>	0.0000	<b>225775</b>	0.0000
Pundit	<b>3448</b>	0.0000	<b>226213</b>	0.0000
Reversal	<b>1499</b>	0.0000	<b>228477</b>	0.0000
<i>Scholar</i>	<i>3274</i>	<i>0.0000</i>	<i>221923</i>	<i>0.9182</i>
Self-Learner	<b>2629</b>	0.0000	<b>226300</b>	0.0000
Stellar Question	<b>1436</b>	0.0000	<b>228434</b>	0.0000
<i>Strunk &amp; White</i>	<i>6219</i>	<i>0.0000</i>	<i>222928</i>	<i>0.4544</i>
<i>Student</i>	<i>3529</i>	<i>0.0000</i>	<i>222361</i>	<i>0.7688</i>
<i>Supporter</i>	<i>9518</i>	<i>0.0000</i>	<i>218409</i>	<i>1.0000</i>
Taxonomist	<b>428</b>	0.0000	<b>229044</b>	0.0000
<i>Teacher</i>	<i>10071</i>	<i>0.0000</i>	<i>215652</i>	<i>1.0000</i>
Tumbleweed	<b>53</b>	0.0000	<b>228639</b>	0.0000
<i>Yearling</i>	<i>3471</i>	<i>0.0000</i>	<i>223583</i>	<i>0.1368</i>

## CHAPTER 4

# HYPOTHESIS TESTS FOR AGGREGATED DATA WITH DEGREE DISPARITY

Degree disparity occurs when an attribute on an entity is statistically associated with the number of other entities with which it shares a relationship [41]. Degree disparity combines with some common aggregation functions to produce systematically higher or lower aggregated values when the cardinality of the input values is high. Thus, any time relational data are propositionalized using aggregation, the transformed data table may exhibit degree disparity bias.

For instance, given variability in the values of the underlying variable being aggregated, `SUM`, `MAX`, and `COUNT` will all return systematically higher values given high cardinality; `MIN` will produce lower values; and `MODE` and `AVG` will produce less extreme values. When data are propositionalized using these aggregation functions, statistical dependencies between the values of one attribute and the aggregated values of another attribute can be erroneously interpreted as dependence between the original attributes.

Consider the relationship between the age of a professor and the number of citations on papers she has written. In general, older professors will have higher a degree (pun not intended) with regard to papers by virtue of having spent more time publishing. Even in a world where the citation count of a given paper is completely random, age will appear associated with aggregations of paper citation counts such as `MAX` (capturing the number of citations on the most well-cited paper) or `SUM` (capturing the total number of citations) due to the sensitivity of these aggregations to degree.

## 4.1 Statistical bias in aggregated domains

Figure 4.1 depicts the distribution of z-scores for relational data that exhibit degree disparity. Each of the different aggregators exhibits a different amount of bias, though all will clearly cause Type I errors for a two-tailed hypothesis test. Even AVG, which is centered, has increased variance when compared to the reference distribution. Also pictured are distributions for identically structured data that do not have degree disparity. Again, propositionalization erases the relational structure present in the data, so given the value of a test statistic, it is unclear from the propositionalized data which of the distributions from Figure 4.1 is the appropriate reference distribution.

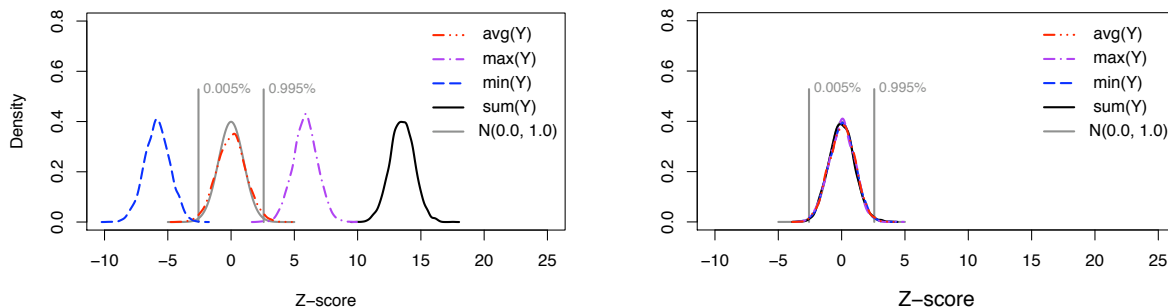


Figure 4.1: Distribution of z-score values for AVG, MAX, MIN, and SUM in a relational data set with moderate degree disparity. The sampling distributions indicate dependence even in the absence of dependence in the original data. Here, even though  $X$  and  $Y$  are marginally independent,  $X$  appears significantly correlated with aggregations of  $Y$ .

Figure 4.2 depicts Type I error curves for data with degree disparity using the SUM and MAX aggregations. As in the case with instance dependence, error rates are much higher than those expected at the  $\alpha = 1\%$  level. For data with a moderate level of degree disparity, the MAX aggregator has an error rate of 15% while SUM is greater than 70%.

As with the errors associated with replication, degree disparity will be entirely invisible to an end-user of a RDBMS. When related tables are joined through foreign

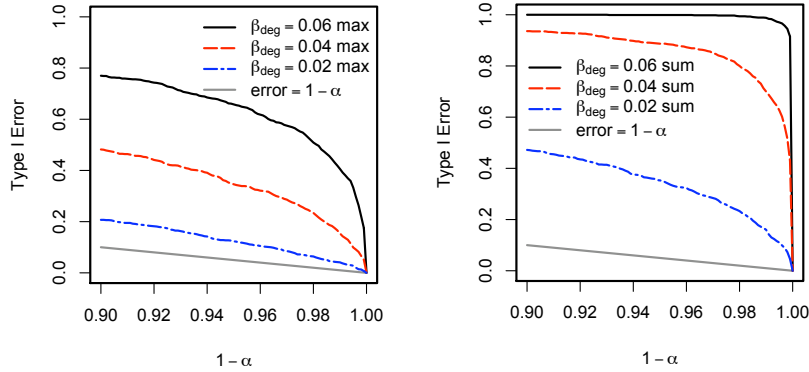


Figure 4.2: Type I error as a function of alpha for MAX (left) and SUM (right) aggregations under degree disparity. The value of  $X$  varies linearly with degree (parameterized by coefficient  $\beta_{deg}$ ). At the  $\alpha = 0.01$  level, the Type I error rates are 15% and 70% for MAX and SUM, respectively.

keys over one-to-many or many-to-many relationships, aggregators are specified to summarize the records in the higher-cardinality table. In doing so, all relational information is lost, along with any evidence that would enable the detection of degree disparity bias.

## 4.2 Graphical models for aggregated data

We can use graphical models to understand and correct the bias introduced by degree disparity. Figure 4.3 shows three DAPER models representing alternative generative structures for the situations discussed in Section 4.1. We assume that degree disparity stems from a direct causal dependence between the variable  $X$  and the probability that one or more relationships exist, affecting the degree variable  $deg$ . Furthermore, we assume that the aggregation function  $f$  is sensitive to changes in degree. Thus, model  $H_2$  indicates that the degree of entities  $A$  depends on the value of  $X$ .

Model  $H_0$  corresponds to the null hypothesis under which  $X$  and  $f(Y)$  are marginally independent. Models  $H_1$  and  $H_2$  represent data in which  $X$  and  $f(Y)$  are correlated.

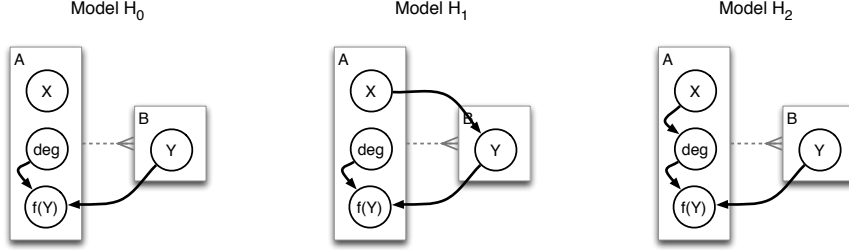


Figure 4.3: DAPER models for one-to-many data with degree disparity. Model  $H_0$  represents the null hypothesis that  $X$  is marginally independent of both  $Y$  (and therefore, any aggregation  $f(Y)$ ) and  $deg$ , while Model  $H_1$  specifies that  $X$  has influence over  $Y$ . Model  $H_2$  represents data that exhibit degree disparity. Here, the value of  $X$  varies with the number of  $B$  entities that each  $A$  connects to, but is independent of the  $Y$  values on those entities.

Once again, knowledge of marginal dependence between  $X$  and  $f(Y)$  can be used to reject  $H_0$ , but it cannot differentiate between  $H_1$  and  $H_2$ .

Rather than explicitly representing link existence, the effects of degree disparity can be alternatively represented in a graphical model that captures degree in a variable, as in the DAPER models shown in Figure 4.4. Here, the variable  $deg$  represents the number of related entities  $B$  (the degree of  $A$ ). As detailed in Section 2.1.2, to propositionalize with aggregation, we construct the variable  $A.f(Y)$  to represent the value produced by aggregating individual  $B.Y$  values. In contrast to the DAPER models in Figure 4.3, when rolled out these models make clear why both models  $H_1$  and  $H_2$  would exhibit dependence between  $X$  and  $f(Y)$ . In both cases, a collider-free undirected path exists between the variables. However, the models differ with respect to a direct causal dependence between  $X$  and  $Y$ .

Figure 4.5 further demonstrates the agreement between the independence relationships described by d-separation and empirically observed results. Here, we present two alternative models similar to model  $H_2$  above. In the first (a), there is no dependence between  $X$  and degree; in (b), the aggregator used (random selection) is insensitive to degree. In both cases, there are no collider-free paths connecting  $X$  to

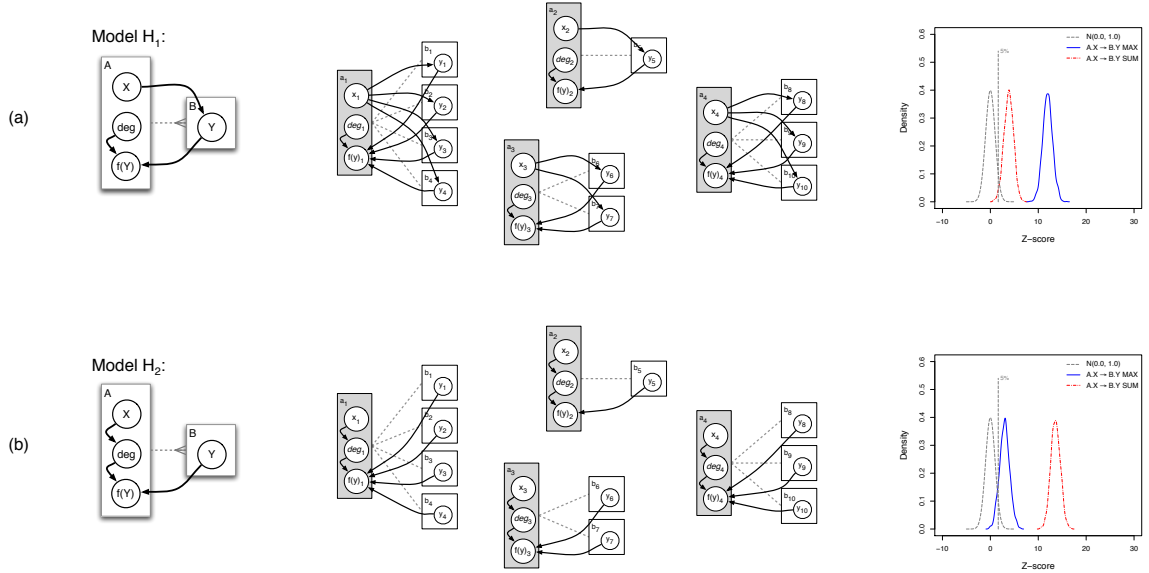


Figure 4.4: DAPER models corresponding to the DAPER models in Figure 4.3, along with ground graphs and Z-score distributions for  $X$ ,  $f(Y)$ . The effects of degree disparity are represented by the dependence of the  $deg$  on  $X$ , coupled with an aggregation  $f(Y)$  that is sensitive to degree (and therefore dependent on  $deg$ ). In both cases there are d-connecting paths in the ground graph connecting  $X$  with  $f(Y)$ .

$f(Y)$ , and Z-scores are normally distributed around 0. It is worth mentioning that the RANDOM aggregator, as shown in Figure 4.5b, is functionally equivalent to the link sampling technique outlined in Chapter 3. With link sampling, a subgraph is generated from the original data such that at most a single related entity is included for each multiply connected entity. Similarly, the RANDOM aggregator selects a single attribute value from each one-to-many or many-to-many group. Tests based on either will be bias-free.

In situations where degree disparity bias is indeed present, the propositional models suggest a simple test of conditional independence: Conditioning on degree will d-separate  $X$  and  $f(Y)$ . Figure 4.6 depicts the empirical distributions of the conditional test for data generated under both models. The data generated under Model H<sub>2</sub>



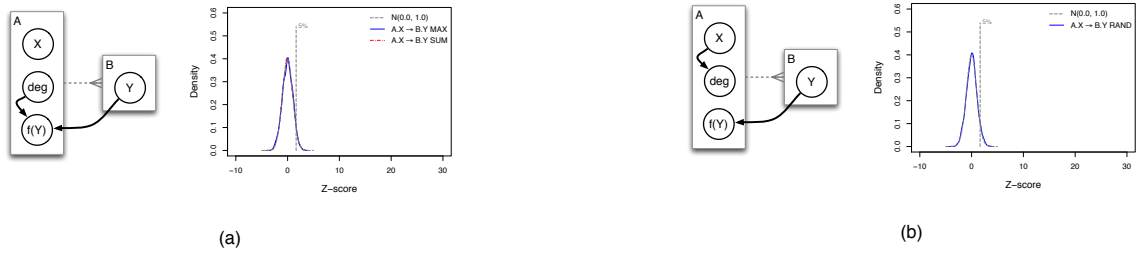


Figure 4.5: Models and Z-score distributions for data sets with (a) no direct dependence between  $X$  and degree, and (b) no dependence between the aggregator used and degree. In both cases, the lack of a d-connecting path between  $X$  and  $f(Y)$  eliminates the degree disparity effects shown in Figure 4.4b.

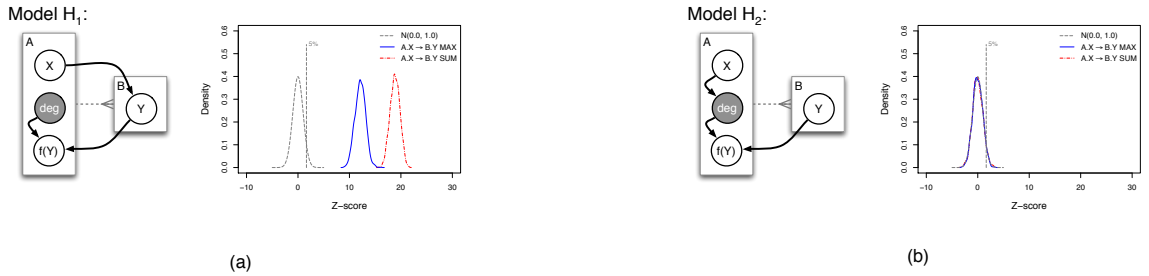


Figure 4.6: Conditioning on degree removes bias for statistics based on data with degree disparity, allowing differentiation from data containing actual association between  $X$  and  $Y$ . (a) Empirical distribution of Z-score after conditioning on degree for data generated under Model  $H_1$ . (b) Conditional Z-score distribution for data from model  $H_2$ .

indicate no significant dependence, while the data under  $H_1$  do show significant dependence. Conditioning on degree successfully differentiates between the two models.

### 4.3 Empirical results

Below, we present empirical results on two real-world domains using the techniques described above. In the first, we examine an erroneous causal claim made about scoring order in professional hockey through the lens of degree disparity bias. In

addition, we examine the errors associated with use of naive hypothesis tests on data from Stack Overflow.

### 4.3.1 NHL scoring

A common claim made about professional hockey is that scoring first in a game is a key to victory [6, 11]. Statistically, the team that scores first tends to win over 60% of the time. However, the mention of this fact often carries with it an explicit causal claim; that is, given two *evenly* matched teams, the one who scores first is going to win more than 50% of the time due to change in strategy that comes with playing with a lead or psychological momentum.

We evaluate the validity of this causal claim using the graphical models framework in conjunction with scoring data collected for over eighteen thousand National Hockey League contests held between 1993 and 2009.<sup>1</sup> Our results provide strong evidence that although scoring first in hockey is strongly correlated with victory, this association is not, in fact, causal. Furthermore, the erroneous attribution of scoring first toward winning can be explained as a form of degree disparity, and factored out using conditional independence tests like the ones described above.

Figure 4.7a depicts an ER diagram for scoring in a hockey game. Each game has a *First* and *Win* attribute that indicate which of the two teams (referred to here as “Team A” and “Team B”) scores first and eventually wins. Note that some games may end in a tie, and that in the case of a 0-0 tie, neither team scores first (or, more precisely, at all). Each goal event is represented by a Goal A or Goal B entity that carries with it a time  $T$ .

Figure 4.7b shows a DAPER model for the scoring domain, with some added Game attributes. We add a *minimum* aggregation over the timestamps of the goal entities. Given these additional attributes, the *First* variable becomes deterministic

---

<sup>1</sup><http://www.hockeyboxscores.com>

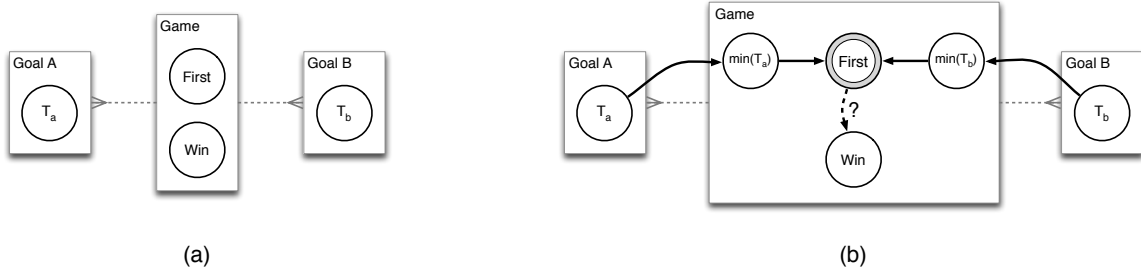


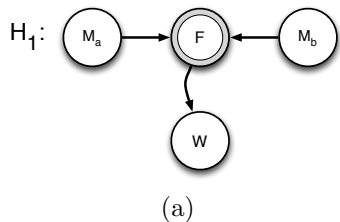
Figure 4.7: (a) ER diagram depicting the hockey scoring domain. Each Game entity is related in a one-to-many manner to Goal A and Goal B entities, which have a timestamp attribute. (b) DAPER diagram for the hockey scoring domain. The Game entities carry additional aggregated attributes capturing the minimum goal time for both Team A and Team B goals, which together determine the value of the *First* attribute. The dashed edge, whose existence we wish to evaluate, represents the causal effect of scoring first on winning.

since the value of *First* is completely determined by a comparison of  $\min(T_a)$  and  $\min(T_b)$ . The question we wish to answer is whether or not scoring first has a direct effect on *Win*, represented by a dashed edge with a question mark.

The claim that scoring first leads to victory is represented by the propositionalized graphical model shown in Figure 4.8a (for clarity, the  $\min(T)$ , *First*, and *Win* variables are shown as  $M$ ,  $F$ , and  $W$ , respectively). Since game entities and goals are related in a strict one-to-many relationship, this simple model is adequate to represent the iid subgraphs that make up the full ground graph after propositionalization through aggregation.

Given this model, naive use of hypothesis testing seems to validate the model and the claim. The results of statistical hypothesis for different combinations of variables are listed in the table in Figure 4.8. Variables  $M_h$  and  $M_a$  are marginally independent, but conditionally associated given  $F$  or  $W$ . In addition, both are marginally dependent on  $F$  and  $W$ , which are in turn marginally dependent on each other.

However, the model depicted in Figure 4.8 fails to capture the structural information that is lost during propositionalization. As detailed above, the *minimum*



association	test	statistic	p-value	conclusion
$M_a \perp\!\!\!\perp F$	t-test	82.03	$< 10^{-16}$	<i>reject</i>
$M_b \perp\!\!\!\perp F$	t-test	88.66	$< 10^{-16}$	<i>reject</i>
$M_a \perp\!\!\!\perp W$	t-test	34.05	$< 10^{-16}$	<i>reject</i>
$M_b \perp\!\!\!\perp W$	t-test	35.91	$< 10^{-16}$	<i>reject</i>
$M_a \perp\!\!\!\perp M_b$	Pearson	0.0001	0.3575	<i>accept</i>
$M_a \perp\!\!\!\perp M_b \mid F$	Guo	61.87	$< 10^{-16}$	<i>reject</i>
$M_a \perp\!\!\!\perp M_b \mid W$	Guo	8.33	$< 10^{-16}$	<i>reject</i>
$F \perp\!\!\!\perp W$	$\chi^2$	1332.20	$< 10^{-16}$	<i>reject</i>

(b)

Figure 4.8: (a) Graphical model representing the hypothesis that winning in hockey ( $W$ ) is causally dependent on scoring first ( $F$ ), which is deterministically related to the aggregated minimum scoring times of both teams ( $M_a, M_b$ ). (b) Results of hypothesis tests conducted on 18k NHL hockey contests from 1993-2009

aggregator is sensitive to degree. This sensitivity is illustrated by the density plot in Figure 4.9. Clearly, the distribution of first goal time differs with degree ( $r^2 = 0.1822$ ,  $p < 10^{-16}$ ).

We can capture this effect by incorporating degree variables into the graphical model, as shown in Figure 4.9. When we include degree in the model,  $W$  (representing the winner of the game) becomes deterministic, as its value derives from a simple comparison of  $D_h$  and  $D_a$  which capture the degree of Goal entities, i.e., score, of each team.

Since  $F$  and  $W$  are both discrete, deterministic variables, we cannot directly verify the validity of the model with the statistical tests used previously. For instance, we cannot test for conditional independence between  $F$  and  $W$  by conditioning on the

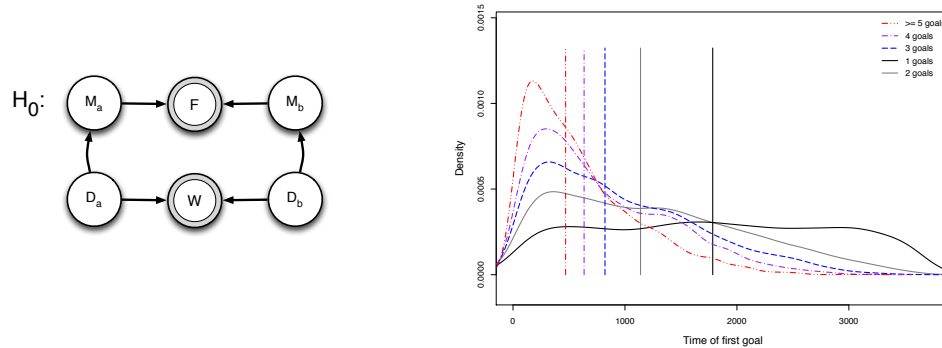


Figure 4.9: (left) Alternative hypothesis for the hockey scoring domain. Here, scoring first has no causal effect on winning; rather, any perceived dependence relationships are an effect of degree. (right) Distribution of first goal time as a function of total goals scored. The vertical lines designate the mean of each density curve.

degree variables, since doing so will hold  $W$  constant as well. Likewise, conditioning on  $M_h$  and  $M_a$  will hold  $F$  constant. Since  $F$  and  $W$  are categorical, we cannot d-separate them with a numeric conditioning set composed of  $M_h$  and  $D_a$  (or  $M_a$  and  $D_h$ ).

We can, however, indirectly evaluate the model using an alternative set of hypothesis tests. According to the model in Figure 4.9,  $M_h$  and  $D_a$  are marginally independent, but conditionally dependent given  $F$  or  $W$ . The NHL data bear this out, as the  $r^2$  value for  $M_h$  and  $D_a$  is 0.0001, ( $p = 0.1624$ ), with a conditional  $z$ -score = 28.0461 when conditioning on  $W$  and  $z$ -score =  $-23.448965$  when conditioning on  $F$  (in both cases,  $p < 10^{-16}$ ).

By treating degree as categorical, rather than discrete, we can perform an additional test on the independence of  $M_h$  and  $D_a$  while conditioning on both  $W$  and  $D_h$ . In this case, the model dictates conditional independence, and the data agree

( $z\text{-score} = 0.7050, p = 0.4808$ ).<sup>2</sup> Furthermore, this allows us to conclude that there is no d-connecting path between  $M_h$  and  $D_a$  other than the one that flows through  $D_h$ . Consequently, there can be no unknown path connecting  $M_h$  (or  $F$ ) with  $D_a$  and  $W$ , providing very strong evidence that there is, indeed, no causal effect on winning from scoring first in hockey.

### 4.3.2 Stack Overflow

Since the Stack Overflow data presented in Section 3.5 is related in a one-to-many manner, it can be propositionalized using aggregation as well as replication. By aggregating, we eliminate the possibility of instance dependence bias; however, doing so may introduce error if degree disparity is present. To see if this effect is present in Stack Overflow, we measured the correlation between the existence of a badge and an aggregated answer score for each user, using the models from 4.10.



Figure 4.10: Augmented models for aggregation in Stack Overflow data. By including the structural *degree* variable, we can differentiate the two models from data by testing for conditional dependence between *Badge* and *Score* conditioned on *deg*.

By conditioning on degree, we can differentiate the cases where marginal dependence is due to degree disparity from those where it is due to a direct causal mecha-

---

<sup>2</sup>Though not reported here, the results of hypothesis tests for  $M_a$  and  $D_h$  are qualitatively similar and directionally identical to those for  $M_h$  and  $D_a$ .

nism. Figure 4.11 summarizes the results for the marginal and conditional tests using each aggregator. For SUM, MAX, and AVG, all 43 badge types have a marginal dependence with *Answer.score*; conditioning on degree removes this dependence for 39, 40, and 41 of these, respectively. Curiously, *score* is marginally dependent on only 3 badges, and conditioning on degree *induces* a dependence. Table 4.4 lists the full results for each aggregator and badge type. Note that in the cases presented above, we considered each badge in isolation in terms of its causal effect on answer score.

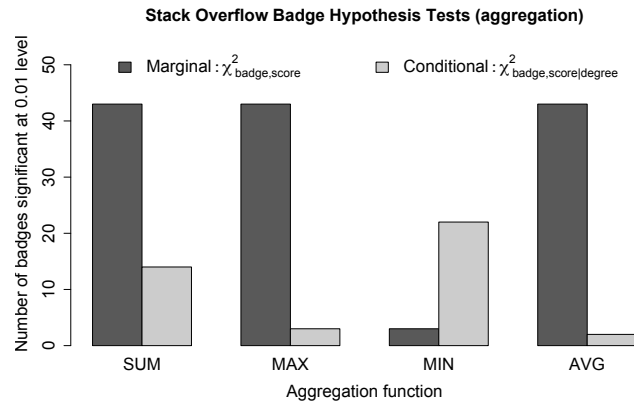


Figure 4.11: Results of marginal and conditional tests of independence for aggregated Stack Overflow data. The plot shows the number of badge attributes that are marginally dependent with answer score for different aggregators. Conditioning on degree removes this dependence for several badge types. In the case of MIN, conditioning on degree can introduce a dependence.

## 4.4 Conclusion

We have used the framework of d-separation to provide the first formal explanation for two previously observed classes of statistical dependencies in relational data. This explanation applies to continuous and discrete variables and essentially any test of conditional independence.

Finally, it is worth repeating that many data sets are created in propositional form, even when their underlying generative processes could more accurately be described

by a relational representation. Thus, the propositional data sets initially provided to many learning algorithms are “born” without the information needed to draw correct inferences about the underlying generative processes that produced them. Disconcertingly, the effects discussed here apply equally to propositional learning algorithms when the data they analyze were originally drawn from relational domains.



Table 4.1: Z-scores for Stack Overflow hypothesis tests (aggregation). Significant values are in **bold**. Italicized values indicate that conditioning on degree has changed the result of the test.

Badge	SUM		MAX		MIN		AVG	
	marg	cond	marg	cond	marg	cond	marg	cond
Autobiographer	<i>23.22</i>	<i>-0.56</i>	<b>24.35</b>	<b>16.49</b>	<i>0.06</i>	<i>3.66</i>	<b>25.85</b>	<b>23.55</b>
Beta	<b>4.24</b>	<b>2.89</b>	<b>7.06</b>	<b>6.34</b>	<b>3.55</b>	<b>4.02</b>	<b>10.72</b>	<b>10.45</b>
Citizen Patrol	<b>42.32</b>	<b>11.58</b>	<b>31.63</b>	<b>19.22</b>	<i>-2.84</i>	<i>2.71</i>	<b>27.57</b>	<b>24.15</b>
Civic Duty	<b>47.66</b>	<b>10.78</b>	<b>35.70</b>	<b>21.45</b>	<i>-3.44</i>	<i>2.97</i>	<b>29.83</b>	<b>26.02</b>
Cleanup	<b>42.58</b>	<b>7.49</b>	<b>27.18</b>	<b>13.60</b>	-4.71	1.12	<b>19.73</b>	<b>15.91</b>
Commentator	<i>33.85</i>	<i>-10.32</i>	<b>35.59</b>	<b>22.61</b>	-4.95	0.88	<b>33.42</b>	<b>29.97</b>
Critic	<i>35.29</i>	<i>-3.98</i>	<b>37.26</b>	<b>24.99</b>	-3.38	2.25	<b>36.05</b>	<b>32.78</b>
Disciplined	<b>50.34</b>	<b>26.82</b>	<b>30.12</b>	<b>17.43</b>	<i>-2.07</i>	<i>3.58</i>	<b>20.30</b>	<b>16.65</b>
Editor	<i>28.10</i>	<i>-9.12</i>	<b>30.91</b>	<b>19.99</b>	-2.55	2.38	<b>31.37</b>	<b>28.36</b>
Electorate	<b>21.84</b>	<b>5.61</b>	<i>7.45</i>	<i>0.25</i>	-1.87	1.03	<b>5.56</b>	<b>3.47</b>
Enlightened	<b>54.27</b>	<b>23.08</b>	<b>36.36</b>	<b>21.98</b>	<i>-2.78</i>	<i>3.73</i>	<b>30.95</b>	<b>27.15</b>
Enthusiast	<b>46.45</b>	<b>8.87</b>	<b>36.79</b>	<b>22.71</b>	-4.63	1.69	<b>32.32</b>	<b>28.61</b>
Epic	<b>102.21</b>	<b>58.32</b>	<i>21.77</i>	<i>-4.96</i>	<i>-2.80</i>	<i>8.35</i>	<i>8.41</i>	<i>0.83</i>
Famous Question	<b>12.80</b>	<b>10.26</b>	<b>7.47</b>	<b>4.78</b>	-0.61	0.58	<b>5.49</b>	<b>4.64</b>
Fanatic	<b>46.74</b>	<b>21.81</b>	<b>23.47</b>	<b>10.74</b>	<i>-2.66</i>	<i>2.80</i>	<b>17.63</b>	<b>14.02</b>
Favorite Question	<b>18.61</b>	<b>8.88</b>	<b>13.98</b>	<b>9.11</b>	-1.20	0.96	<b>7.75</b>	<b>6.20</b>
Good Answer	<b>42.54</b>	<b>20.05</b>	<b>31.59</b>	<b>20.66</b>	<i>-0.81</i>	<i>4.20</i>	<b>27.41</b>	<b>24.31</b>
Good Question	<b>21.24</b>	<b>8.92</b>	<b>16.13</b>	<b>10.35</b>	-1.10	1.46	<b>10.75</b>	<b>8.95</b>
Great Answer	<b>24.29</b>	<b>17.59</b>	<b>15.53</b>	<b>10.17</b>	-0.99	1.40	<b>11.41</b>	<b>9.74</b>
Great Question	<b>15.54</b>	<b>15.62</b>	<b>10.25</b>	<b>7.66</b>	0.42	1.65	<b>6.27</b>	<b>5.41</b>
Guru	<b>39.03</b>	<b>27.89</b>	<b>21.94</b>	<b>13.19</b>	<i>-0.12</i>	<i>3.77</i>	<b>16.64</b>	<b>14.03</b>
Legendary	<b>77.08</b>	<b>74.27</b>	<i>14.03</i>	<i>-1.23</i>	<i>-1.21</i>	<i>5.02</i>	<i>4.83</i>	<i>0.39</i>
Mortarboard	<b>60.99</b>	<b>18.13</b>	<b>43.30</b>	<b>26.17</b>	-5.58	2.20	<b>31.09</b>	<b>26.68</b>
Necromancer	<b>27.07</b>	<b>7.17</b>	<b>22.34</b>	<b>14.33</b>	-2.61	0.95	<b>17.58</b>	<b>15.17</b>
Nice Answer	<b>43.54</b>	<b>11.72</b>	<b>40.66</b>	<b>28.41</b>	<i>-1.04</i>	<i>4.75</i>	<b>38.20</b>	<b>34.94</b>
Nice Question	<i>23.76</i>	<i>1.48</i>	<b>19.17</b>	<b>11.18</b>	-1.34	2.16	<b>17.62</b>	<b>15.26</b>
Notable Question	<i>11.68</i>	<i>0.48</i>	<b>11.89</b>	<b>8.02</b>	-1.70	0.03	<b>10.39</b>	<b>9.17</b>
Organizer	<b>43.13</b>	<b>4.61</b>	<b>37.44</b>	<b>23.85</b>	<i>-3.57</i>	<i>2.61</i>	<b>32.12</b>	<b>28.49</b>
Peer Pressure	<i>28.93</i>	<i>-1.11</i>	<b>22.34</b>	<b>12.04</b>	-6.20	-1.83	<b>15.06</b>	<b>12.02</b>
Popular Question	<i>15.59</i>	<i>-2.70</i>	<b>17.49</b>	<b>11.80</b>	-0.76	1.81	<b>17.50</b>	<b>15.76</b>
Populist	<b>38.82</b>	<b>29.42</b>	<b>17.33</b>	<b>8.65</b>	-1.96	1.75	<b>10.66</b>	<b>8.06</b>
Pundit	<b>44.17</b>	<b>41.15</b>	<b>16.65</b>	<b>8.05</b>	-1.20	2.47	<b>10.31</b>	<b>7.74</b>
Reversal	<b>23.84</b>	<b>22.97</b>	<b>7.26</b>	<b>2.60</b>	-1.71	0.21	<b>5.46</b>	<b>4.07</b>
Scholar	<i>19.63</i>	<i>-14.50</i>	<b>23.95</b>	<b>14.86</b>	-1.57	2.48	<b>24.82</b>	<b>22.21</b>
Self-Learner	<b>19.44</b>	<b>3.31</b>	<b>17.69</b>	<b>11.65</b>	-0.97	1.74	<b>15.23</b>	<b>13.37</b>
Stellar Question	<b>16.40</b>	<b>16.37</b>	<b>11.79</b>	<b>9.10</b>	0.04	1.34	<b>6.18</b>	<b>5.26</b>
Strunk&White	<b>52.71</b>	<b>23.06</b>	<b>24.73</b>	<b>10.06</b>	-3.77	2.46	<b>15.80</b>	<b>11.64</b>
Student	<i>19.84</i>	<i>-14.03</i>	<b>23.59</b>	<b>14.48</b>	-1.64	2.40	<b>24.02</b>	<b>21.40</b>
Supporter	<i>26.36</i>	<i>-8.53</i>	<b>31.14</b>	<b>21.01</b>	<i>1.15</i>	<i>5.87</i>	<b>35.59</b>	<b>32.81</b>
Taxonomist	<i>7.59</i>	<i>-0.49</i>	<b>8.25</b>	<b>5.61</b>	0.45	1.65	<b>8.58</b>	<b>7.75</b>
Teacher	<i>26.80</i>	<i>-3.97</i>	<b>32.41</b>	<b>23.03</b>	<b>2.73</b>	<b>7.21</b>	<b>38.21</b>	<b>35.61</b>
Tumbleweed	<i>7.63</i>	<i>-19.77</i>	<b>10.37</b>	<b>4.15</b>	-6.26	-3.73	<b>5.00</b>	<b>3.13</b>
Yearling	<b>18.10</b>	<b>4.40</b>	<b>20.98</b>	<b>15.87</b>	<b>2.93</b>	<b>5.40</b>	<b>25.62</b>	<b>24.04</b>

## CHAPTER 5

# RELATIONAL BLOCKING

Conditional independence is a central concept for learning and reasoning with causal models [66, 85]. Explicit tests for conditional independence are the basic operators used in many algorithms for learning the structure of such models. These tests identify conditional independence by explicitly evaluating the impact of conditioning on specific sets of one or more observed variables.

In this section, we present *relational blocking*,<sup>1</sup> a fundamentally new algorithmic operator for learning conditional independence by exploiting relational structure among data entities [75]. Relational blocking behaves in ways that differ fundamentally from simple conditioning on observed variables. Specifically, it adjusts for sets of both observed and latent variables when they act as confounders. Yet it does not induce dependence when these variables are common effects.

Relational blocking formalizes approaches commonly used in the social sciences [89] and reveals statistical implications of these methods that are both surprising and useful. Despite the widespread use of blocking in other fields, it has not been used in algorithms for learning the structure of causal models such as PC [85] or RPC [55]. We describe relational blocking using DAPER models and ground graphs [35], and we use these formalisms to show how blocking is distinct from simple conditioning. We demonstrate the effectiveness of relational blocking by showing how it reduces

---

<sup>1</sup>The term blocking is overloaded in the statistical sciences. In this paper, blocking refers to instance grouping, and is distinct from the concept of path blocking found in the graphical models literature.

variability and adjusts for entire classes of observed and latent confounders. Finally, we examine the frequency with which relational blocking can be applied to discover causal dependencies in data describing social media systems.

## 5.1 Example

Consider the problem of understanding the operation of Wikipedia, a peer-produced encyclopedia of general knowledge.<sup>2</sup> Wikipedia articles, or pages, are produced collectively by thousands of volunteer users. Pages are created and modified by users, and users often organize themselves into groups called “projects,” each of which covers a general topic. Within a project, individual pages are assessed by editors for “quality,” an objective evaluation of key criteria.

One of the most persistent claims about Wikipedia is that its high quality stems from the large number of users that collaborate to write each article [48]. We call this the *many-eyes hypothesis*: The more users that revise an article, the higher the quality of that article. If we knew that this association was causal, then we could increase the quality of an article by directing more users to revise it. However, to determine that a causal dependence exists between editor count and article quality, we must eliminate other plausible alternative models that could explain an observed dependence.

A naive approach to this question would examine a large number of pages at a given point in time and estimate the dependence between the number of editors  $E$  and the quality of the page  $Q$ . This method tests the assumptions encoded in the graphical model shown in Figure 5.1a. Given this design, the variables are highly correlated: We sampled twenty random Wikipedia pages from ten projects, and found that a chi-square test yields  $\chi^2=101.83$  ( $n=189$ , since not all pages had  $Q$  and  $E$

---

<sup>2</sup><http://www.wikipedia.org>

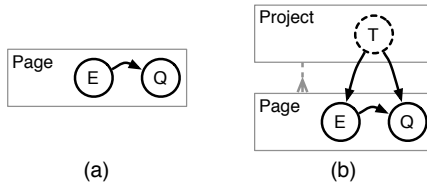


Figure 5.1: (a) A simple graphical model describes the dependence between the number of editors  $E$  and quality  $Q$  of an article, but it does not account for common causes. (b) A more complex graphical model incorporates latent common causes  $T$  associated with project.

values;  $\text{DOF}=12$ ;  $p = 2.44 \times 10^{-16}$ ), and approximately 66% of the variance of page quality could be attributed to the number of editors. This approach is quite similar to those conducted by many algorithms in machine learning—it identifies a statistical association between two variables, but that association is insufficient to establish a causal dependence. The observed dependence could stem from a common cause, such as the general topic area  $T$ . It is plausible that pages on topics of high interest to Wikipedians may be edited by a disproportionately large number of users (that is,  $T$  causes  $E$ ). Additionally, that same interest in topic could drive editors to exert special care when editing, thereby improving quality ( $T$  causes  $Q$ ). If  $T$  is a cause of both  $E$  and  $Q$ , then  $E$  and  $Q$  will be marginally dependent even if their dependence is not directly causal.

Unfortunately, since topic  $T$  is not a measured variable, we cannot account for its influence on  $E$  and  $Q$  through simple conditioning. However, since project structure is based on topic, we can adjust for this potential common cause by blocking. Projects govern pages that are thematically similar, so blocking on project can factor out the latent influence of topic. This alternative approach helps to differentiate between the graphical model shown in Figure 5.1a and the model in Figure 5.1b.

The DAPER model in Figure 5.1a contains only a single entity type and thus is equivalent to a conventional Bayesian network. However, the DAPER model in Figure 5.1b shows dependencies that span two entity types in which an instance of

one entity (Project) typically connects to more than one instance of a second entity type (Page). A given Project instance is related to several Page instances, each of which contains an instance of the  $E$  variable. Each of those  $E$  variables has the same parent variable  $T$  on the given Project instance.

When we use project relations to arrange pages into groups, we find that the average correlation between editor count and page quality decreases. A Cochran-Mantel-Haenszel test yields  $M^2=82.33$  ( $n=189$ ;  $\text{DOF}=12$ ;  $p = 1.48 \times 10^{-12}$ ). Although lower, this value is still highly significant, and roughly 53% of the variance would now be attributed to the number of editors. The effect size has dropped, but it is still statistically significant.

However, using this approach allows a stronger claim regarding the source of the association because we have plausibly factored out at least one potential (unmeasured) common cause. The ability to factor out multiple variables, observed or latent, is a key benefit of blocking. After ruling out several plausible common causes of variation, we now have much stronger evidence that the dependence between editor count and page quality is causal and that the many-eyes hypothesis is valid.

The example above highlights three concepts whose intersection forms the basis of this work. First, the Wikipedia data set is relational, made up of heterogeneous, interrelated data instances drawn from a relational network. Second, the question being investigated is causal. While there is a marginal association between editor count and quality, we are trying to establish a more powerful claim. Lastly, we are able to adjust for confounding factors (and draw a stronger causal conclusion) by using blocking as a complement to traditional conditioning.

## 5.2 Background

Most modern machine learning algorithms focus on identifying correlations in data. In this work, we are concerned with causal relationships between entities and their associated attributes in relational domains.

The traditional approach in machine learning is to statistically model all possible common cause variables (e.g., Bayesian network learning [85], RPC [55]). These techniques determine structure by finding dependencies among variables through statistical control of restricted sets of parent variables. However, even with a highly accurate model, algorithms that rely exclusively on conditioning can succumb to various problems related to the existence of latent or unmeasured variables and low statistical power.

At its core, blocking is a data grouping strategy used to reduce variation and factor out common causes. The block design, originating in the agricultural experimental design work of Fisher [20], divides data instances into disjoint groups, or blocks, according to the value of one or more blocking criteria. Within each block, confounding factors (often called “nuisance factors”) associated with the blocking variable are held constant, reducing any variability in the outcome (effect) variable that is due to these factors. For example, the analysis of a drug trial might block on the hospital where the treatment was administered, allowing experimenters to control for any environmental factors associated with the facility.

In a network setting, units can be blocked using network structure as well. Relational blocking groups entities that share relations with a common neighbor, called the *blocking entity*. Blocking in this manner can be used to facilitate causal discovery in network data sets consisting of entities (e.g., people, events, or places) that share some type of relationship or action among them. For example, papers written by common authors, or movies produced by the same studio, may form blocks.

The use of relational structure to block by entities rather than attributes can be thought of as a generalization of the classic twin design, in which pairs of twins are blocks. For more than a century, researchers have relied on twin data to account for whole classes of (often unmeasurable) attributes related to family environment and heredity [9].

Blocking is commonly used in experimental studies; for example, the Randomized Complete Block Design refers to a configuration where each possible value of the treatment (cause) variable is paired with each value of the blocking variable to form the blocks. In the multilevel modeling framework, the attributes of the blocking entity would be modeled as a “level” of regression parameters [26].

Note that block assignment should not be confused with the notion of experimental group assignment found in experimental design literature. Experimental groups are homogeneous with regard to treatment (or lack thereof). In contrast, experimental blocks contain instances with varying treatments and outcomes while homogenizing confounding factors that make detecting the relationship between treatment and outcome more difficult.

Blocking is used less commonly in observational, or quasi-experimental settings. In contrast to experimental domains, treatment is not explicitly assigned in non-experimental settings, so factors associated with each block may affect both treatment and outcome.

Previous work in relational learning provides strong evidence that blocking by network structure will have this effect. Relational autocorrelation, a commonly observed trait of network data sets, is indicative of an association between network structure and attributes such that entities sharing common neighbors often share similar attribute values as well [39]. Neville and Jensen exploited this property on unipartite data using Latent Group Models to model an unmeasured attribute on a “coordinating entities” [62]. Of course, autocorrelation may be the result of differing causal

mechanisms: When the existence of relationships stems from attributes, it is referred to as homophily; when the reverse is true, it is called network influence [3, 22]. In either case, blocks constructed from using network structure exhibit less variability than the population at large in terms of treatment, outcome, or both.

The benefit of relational blocking is twofold. The first is statistical: By organizing experimental units into groups such that variability within each block is reduced, we can improve statistical power. Relational blocks hold constant any attribute associated with the blocking entity. In this respect, blocking serves the same purpose as conditioning. However, unlike conditioning, blocking can simultaneously adjust for the influence of several (even latent) variables. When applied to hierarchical domains (such as the synthetic domains described in Section 5.3), relational blocking serves a similar purpose to multilevel modeling, where the influence of factors associated with a common group or entity is modeled within the appropriate regression equation associated with each level of the hierarchy [29].

The second benefit relates to causal reasoning. The *causal sufficiency assumption* [85] states that any possibly confounding variables are observed. When blocking, factors that are held constant within each block can be eliminated as possible common causes of treatment and outcome, enabling stronger claims of causal sufficiency and pruning the space of alternative causal models. By eliminating entire classes of potential common causes, including both measured and latent variables, the causal sufficiency assumption is relaxed, in that confounding factors can be accounted for even if they are unobserved, assuming that the *entities* they are associated with are observable.

In addition, blocking lacks one of the central disadvantages of standard conditioning — the potential to induce independence if the conditioning variable is a common effect, rather than a common cause or a mediating variable in a directed causal chain. This is a surprising and highly beneficial feature of blocking, and it is one that, to our



knowledge, is unrecognized in the literature on multi-level models and experimental design. The next section describes this effect in more detail.

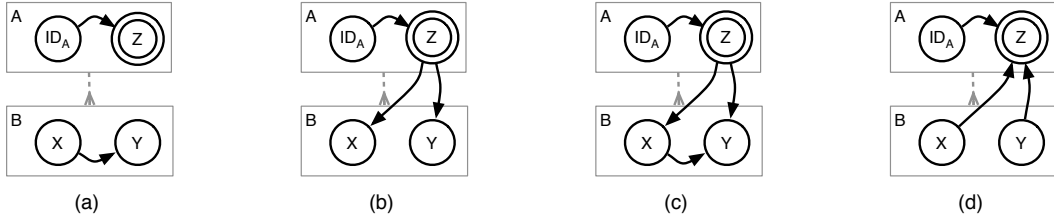


Figure 5.2: Different generative models for bipartite one-to-many data. In case (a),  $X$  directly influences  $Y$ . In (b),  $X$  and  $Y$  have a common cause ( $Z$ ), and blocking and conditioning will both render them conditionally independent. In (c), blocking and conditioning are able to factor out the influence of confounder  $Z$ , but the two remain conditionally dependent. Case (d) depicts  $Z$  as a common effect of  $X$  and  $Y$ ; here,  $X$  and  $Y$  are rendered dependent when conditioned on  $Z$  (Berkson’s paradox), yet remain independent when  $Z$  is held constant through blocking using entities of type  $A$ . In all models, the double circles represent the deterministic dependence between  $ID_A$  and  $Z$ .

### 5.3 Blocking versus conditioning

It may be tempting to view blocking merely as a form of conditioning. While the two serve common purposes—reducing variability and adjusting for common causes—they do not produce the same statistical results. To illustrate this point, we generate synthetic bipartite data and compare the results of blocking and conditioning for different generative models of attribute structure. Each data set consists of entities of two types,  $A$  and  $B$ , connected in a one-to-many manner. In all cases, there are 10,000  $B$  entities, with the number of  $A$  entities varying between different experiments. Each  $A$  entity carries two attributes,  $Z$  and  $H$ , with the former considered measured and the latter latent. The  $B$  entities also have two attributes,  $X$  and  $Y$ , both of which are measured.

In each experiment, the goal is to assess the dependence between  $X$  and  $Y$  while either blocking on entity  $A$  or conditioning on variable  $Z$ . Note that  $Z$  is generated

as a continuous variable; in each experiment it is discretized to a fixed number of levels in order to compare the results of blocking and conditioning using the same hypothesis test (we use Guo’s weighted Pearson’s  $r$  correlation [33]). While not presented here, we found that the results of experiments using partial correlation with an untransformed  $Z$  were qualitatively similar.

To represent blocking with a graphical model, we introduce an identity variable  $ID_A$  [74]. The models in Figure 5.2 depict bipartite, one-to-many models with the identifier variable included. With this framework, we can formally define relational blocking:

**Definition 7.** *Let  $A$  and  $B$  be two entity sets in a  $k$ -partite network. A block contains a set of  $B$  entities that link to a common  $A$  entity. Let  $ID$  be the unique identifier of a block, and let  $X$  and  $Y$  be two attributes of  $B$ . We define **Relational blocking** as the process that evaluates the conditional independence of  $X$  and  $Y$  given  $ID$  by grouping  $B$  entities into disjoint blocks.*

The directed edge connecting  $ID_A$  and  $Z$  denotes a *deterministic* dependence between the two. Certainly,  $ID_A$  determines  $Z$ , since the value of  $ID_A$  indicates the value of  $Z$  with a simple lookup. The reverse is not true, however, as several  $A$  entities may share the same value of  $Z$  while having different identifiers.

Despite being common in real data sets, the consequences of determinism in graphical models are rarely discussed in the machine learning literature. The presence of deterministic dependence slightly complicates the rules of  $d$ -separation.<sup>3</sup> The following definition is adapted from Spirtes et al. [85], and Geiger [25]:

**Definition 8.** *Let  $\mathbf{X}$ ,  $\mathbf{Y}$ , and  $\mathbf{W}$  be three disjoint sets of vertices in DAG  $G$ . Let  $Det(\mathbf{V})$  be the set of all variables determined by  $\mathbf{V}$ . Then,  $\mathbf{X}$  and  $\mathbf{Y}$  are  **$d$ -separated***

---

<sup>3</sup>Some authors use the term *D-separation* (with a capital D) to denote  $d$ -separation with determinism. In this work, we will not rely on this typographical convention.

by  $\mathbf{W}$  if and only if for all undirected paths  $P$  between  $\mathbf{X}$  and  $\mathbf{Y}$  either (1)  $\exists v \in \text{colliders}(P)$  such that  $v \wedge \text{descendants}(v) \notin \mathbf{W}$  or (2)  $\exists v \in \text{noncolliders}(P)$  such that  $v \in \text{Det}(\mathbf{W})$ .

### 5.3.1 Common causes

The first set of experiments simulate the scenario outlined in the introduction. Figures 5.2a and 5.2b represent two generative models where  $X$  and  $Y$  are marginally dependent, denoted  $X \not\perp Y$ . In the first case,  $X$  has direct influence on  $Y$ ; in the second, their marginal dependence is due to a common cause.

Under the framework of  $d$ -separation [65], this marginal dependence is evident from the existence of a collider-free path connecting  $X$  and  $Y$  in either case. From data, we can differentiate the two models with a conditional independence test. Conditioning on  $Z$  has no effect on the independence relationship between  $X$  and  $Y$  in model 5.2a, but interrupts the  $d$ -connecting path in model 5.2b, rendering  $X$  and  $Y$  conditionally independent:  $X \perp\!\!\!\perp Y \mid Z$ .

The data for model 5.2b are generated such that  $X, Y = \beta_Z Z + \epsilon$ . For all values of  $\beta_Z$ , blocking is comparable to conditioning in terms of Type I error, maintaining an error level of less than 7% for  $\alpha = 0.05$ , with conditioning less than 6%.

This similarity in performance can be explained by the semantics of  $d$ -separation and the observation that, as defined above, blocking is equivalent to conditioning on  $ID_A$ . When conditioning on variable  $Z$ , data are grouped such that the value of  $Z$  is held constant within each group. Similarly, blocking holds constant the entity  $A$  within each group. In model 5.2b,  $Z$  lies on the only  $d$ -connecting path between  $X$  and  $Y$ . Per the above definition, conditioning on  $Z$  or any set of variables that determines  $Z$  will render  $X$  and  $Y$  conditionally independent. Since  $ID_A$  fully determines  $Z$ , conditioning on it (that is, blocking) will  $d$ -separate  $X$  and  $Y$ .

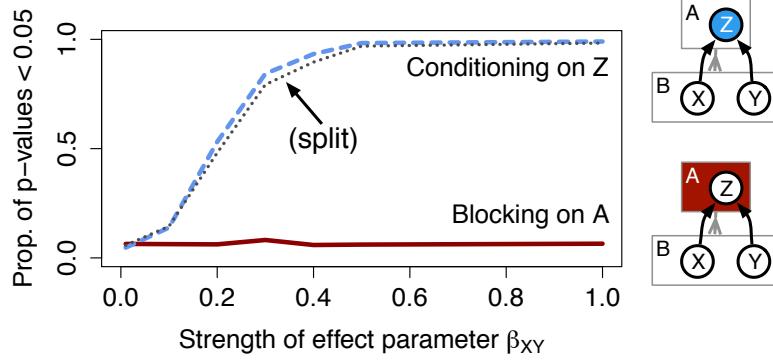


Figure 5.3: Unlike conditioning, blocking does not induce conditional dependence when holding constant a common effect of two marginally independent variables. The line labelled “split” indicates a conditioning analysis with statistical power identical to the blocking analysis.

### 5.3.2 Common Effects

An additional case is described by the model shown in Figure 5.2d. In this case,  $X$  and  $Y$  are marginally independent, while  $Z$  is generated such that  $Z = \beta X' + \beta Y' + \epsilon$ , where  $X'$  and  $Y'$  are the sums of the values of the  $X$  and  $Y$  values for each related  $B$  entity. This case presents an example of Berkson’s paradox [7], where conditioning on a common effect (i.e., collider) will induce dependence between marginally independent variables. Here, blocking and conditioning lead to opposite conclusions. As expected, conditioning on  $Z$  does indeed induce dependence between  $X$  and  $Y$ ; however, blocking on  $A$  does not, even though doing so effectively adjusts for variable  $Z$  as in the conditioning case.

These effects can be seen in Figure 5.3. Conditioning produces the expected result: As we increase the strength of effect parameter  $\beta$ , conditioning induces a dependence between  $X$  and  $Y$  more frequently. Blocking, on the other hand, does not produce any of the conditional dependence described by Berkson’s paradox. The  $d$ -separation criteria stated above agree with our empirical results—conditioning on the collider  $Z$  creates a  $d$ -connecting path, while blocking (conditioning on  $ID_A$ ) does not.

The differences between blocking and conditioning cannot be attributed to statistical power. For the case presented above, the block size (10 instances) is an order of magnitude smaller than the conditioning groups (100). To compensate for this difference, we randomly split each conditioning group into subgroups of 10 instances (labeled as “split” in Figure 5.3). Even with conditioning groups of equal size to the blocks, the proportion of significant  $p$ -values is unchanged.

These results clearly indicate that blocking and conditioning are fundamentally different operations. The difference between blocking and conditioning is illustrated in Figure 5.4. For a small dataset generated under the model in Figure 5.2d, the data have been stratified into contingency tables for both blocking and conditioning. Even for this illustrative example, the results of statistical tests can differ, as the  $p$ -value for the conditioning case is 0.009 (indicating significance at the 0.01 level), compared to 0.033 for blocking (not significant at 0.01).

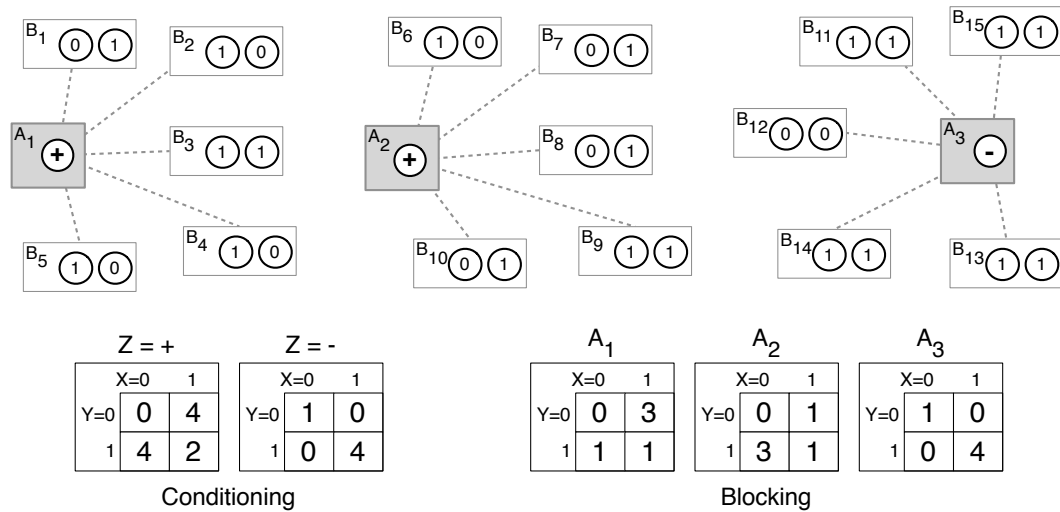


Figure 5.4: Blocking and conditioning are distinct operations, as they stratify the data in different ways. For the above relational data set, conditioning groups the data into two strata, yielding a combined  $\chi^2$  value of 9.44 ( $p=0.009$ ) while blocking groups the data into three strata, producing a  $\chi^2$  value of 8.75 ( $p=0.033$ ).

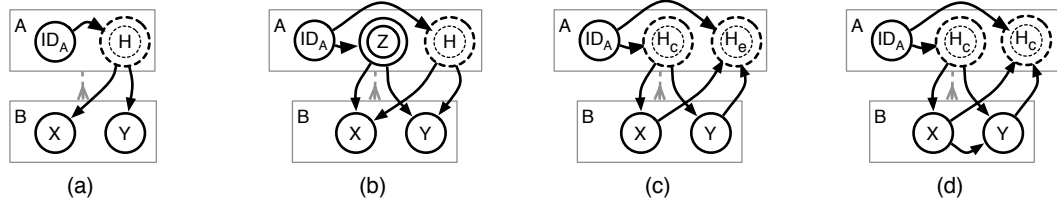


Figure 5.5: Models for bipartite data with latent variables. Models (a) and (b) depict cases where a latent common cause  $H$  exerts influence on  $X$  and  $Y$ . In these cases, blocking is able to render  $X$  and  $Y$  conditionally independent, while conditioning is not. In models (c) and (d),  $X$  and  $Y$  have both a latent common cause  $H_c$  and a latent common effect  $H_e$ . Here, blocking will distinguish between the two models.

### 5.3.3 Latent Confounders

Conditioning and blocking do not perform equivalently in the presence of latent variables. Figures 5.5a and 5.5b depict generative models for data with a latent variable  $H$  acting as a common cause of both  $X$  and  $Y$ . Since  $H$  is unobserved, conditioning on  $H$  is impossible for model 5.5a, while blocking performs as if it is controlling for an observable variable. In the case of model 5.5b, both a measured ( $Z$ ) and latent ( $H$ ) variable exert influence on  $X$  and  $Y$ , such that  $X, Y = \beta_Z Z + \beta_H H + \epsilon$ . The plot in Figure 5.6 depicts Type I error rate at the  $\alpha=0.05$  level with  $\beta_Z$  held constant at 0.5, and  $\beta_H$  varying from 0 to 0.5. Since blocking accounts for all confounders, it can be used to establish conditional independence in the presence of unmeasured factors. Thus, in cases where two variables are marginally dependent, conditioning alone is inadequate for ruling out alternative models such as those in models 5.5a or 5.5b.

The models depicted in 5.5c and 5.5d show cases where  $X$  and  $Y$  have both a latent common cause ( $H_c$ ) and latent common effect ( $H_e$ ). In both cases,  $X$  and  $Y$  are marginally dependent. Blocking renders  $X$  and  $Y$  conditionally independent for model 5.5c, but not 5.5d. As a result, any finding that  $X \not\perp Y \mid ID_A$  cannot be “explained away” by the presence of (latent) common effects when blocking (this property follows directly from the results detailed in Section 5.3.2). Thus, while

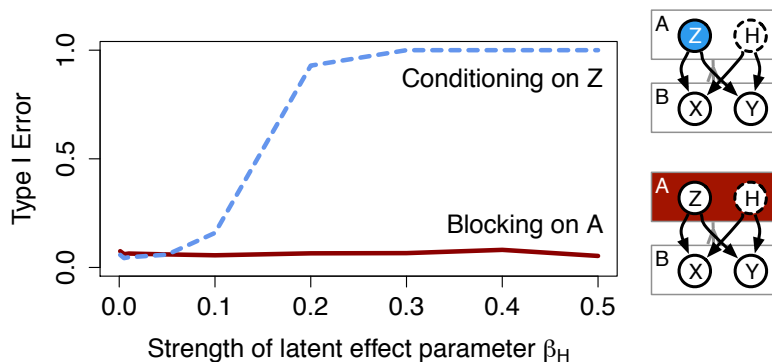


Figure 5.6: The effects of blocking and conditioning differ for data generated under the models shown in Figure 5.5b. Conditioning can only adjust for measured variable  $Z$ , and is susceptible to high rates of Type I error as the strength of the latent effect  $\beta_H$  increases. Blocking accounts for both  $H$  and  $Z$ ; it is not affected by  $\beta_H$ .

blocking can adjust for multiple latent confounders, it introduces no threat to causal conclusions in the presence of latent common effects.

### 5.3.4 Power

The small example in Figure 5.4 illustrates another distinction between blocking and conditioning: Since identifiers and variables are related in a non-injective manner, blocking necessarily stratifies the data into smaller groups. To investigate the effects of the smaller groupings on statistical power, we generated synthetic data using the model found in Figure 5.2a such that  $Y = \beta_X X + \epsilon$ . Figure 5.7 depicts statistical power as a function of effect size, sample size, and block size. In each case, blocking does slightly decrease statistical power, which is expected given the smaller strata. However, given the large size of many modern relational data sets such as Wikipedia, these effects of this decrease are minimal.

## 5.4 Blocking in Practice

To assess the practical utility of relational blocking, we analyzed two domains derived from the peer-production systems Wikipedia and Stack Overflow. Each data

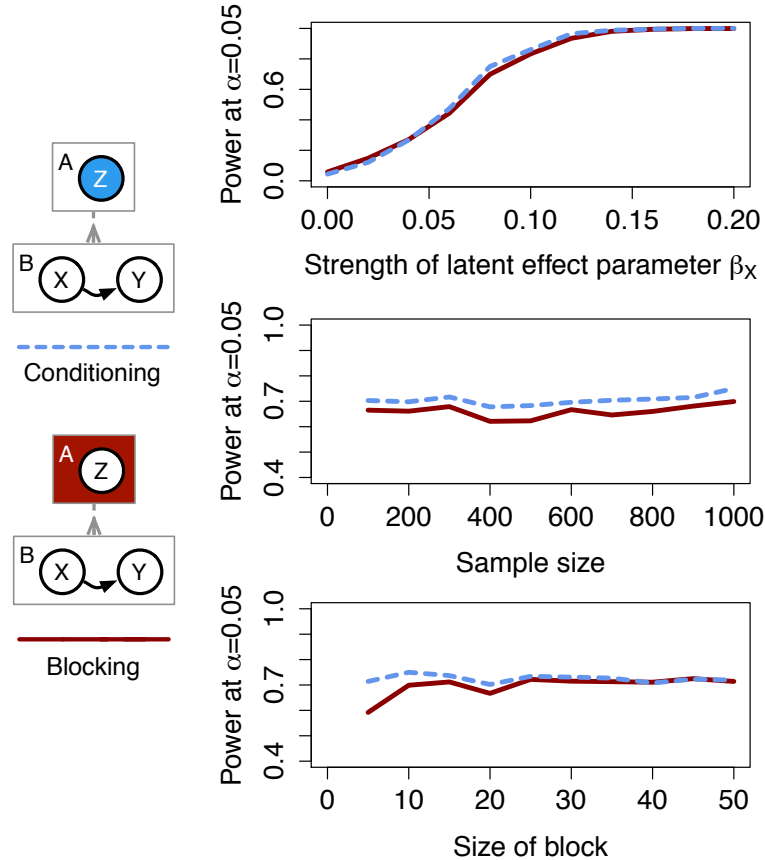


Figure 5.7: Although relational blocking groups the data into smaller strata than conditioning, there is little effect on statistical power.

set comprised multiple related entity types and attributes. The data schema for each can be found in Figure 5.8. Blocking was applicable to 80% of the questions identified by practitioners as the most interesting, and blocking produced substantial changes in results in 28% of the quantitative assessments of actual causal dependencies.

#### 5.4.1 Wikipedia

Although Wikipedia has been the subject of several recent studies (e.g., Kittur [48]), we know very little about how it functions, particularly from a causal standpoint. These aspects make Wikipedia an ideal candidate for studying the applicability and utility of relational blocking.



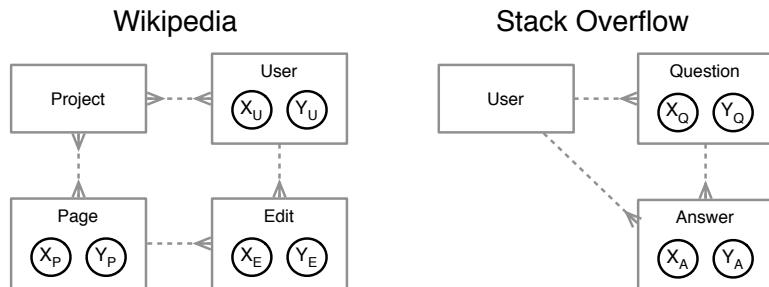


Figure 5.8: Data schemata for Wikipedia and Stack Overflow. Each pair of  $X$  and  $Y$  variables on the same entity can be tested for dependence, and related parent entities can be used for blocking. For example, Wikipedia Page.Quality and Page.Edits can be blocked through Project or User, while Stack Overflow Question.Score and Question.Length can only be blocked through User.

Table 5.1: Details of Wikipedia data

Entity	Attributes	Block Ents.
Page	Adopted by Project, Age, Assessment, Editors, Edits, Featured, Importance, Length Notice, Number of Links, Protected, Quality, Views	Project, User
User	Role, Edits, Membership in Project	Page, Project
Edit	Size, Vandalism, Minor, Reverted	Page, User

Our version of the data contained User entities and Edit events in addition to the Pages and Projects discussed in Section 5.1. The details of the entity types and associated attributes can be found in Table 5.1. In all, there are twenty attributes that are applied to three target entity types (Projects lack intrinsic attributes of their own, and are only used as blocking entities). This schema allows for 174 different relationships apropos to the bipartite models illustrated in Figure 5.2, for which 348 blocking schemes are available (each  $X$ ,  $Y$  attribute pair can be blocked with two different entities).

We took a qualitative approach to determining the applicability of relational blocking. We surveyed ten people, each with a bachelors or masters degree in Computer Science, to obtain a sample of interesting causal questions in the Wikipedia domain. Respondents were given a simple list of attributes and asked to indicate ten pairs of causes and effects they found compelling for study (attributes were presented in one of five random orderings to eliminate biases associated with presentation). The group generated a list of 99 causal discovery tasks (one respondent provided only 9 tasks), 71 of which were unique. Of these, 57 (80%) can be addressed with a simple relational blocking approach such as the one outlined in Section 5.1. While not definitive, these results indicate that relational blocking can be readily applied to the types of problems that interest practitioners.

Table 5.2: Details of Stack Overflow data

<b>Entity</b>	<b>Attributes</b>	<b>Block Ents.</b>
Question	Ans. Count, Mean Ans. Score, Mean Ans. Comment Count, Mean Ans. Length, Comment Count, Favorite Count, Has Accepted Ans., Length, Score, View Count	User
Answer	Accepted, Comment Count, Score, Length	Question, User

#### 5.4.2 Stack Overflow

In addition to the Wikipedia data set discussed in Section 5.1, we examined data from Stack Overflow, an online technical resource that allows users to pose questions as well as answer others' questions. For our study, we examined dependence between attributes on Questions (blocking on Users) as well as attributes on Answers (blocking on Users or Questions), and found a significant change in effect size in 28% of all cases. The complete list of attributes is found in Table 5.2.

For each of the 57 same-entity attribute pairs, we assessed their marginal and conditional independence using all available data for the month of March 2010. For pairs of continuous attributes (e.g., Score, Comment Count), we utilized a blocked Pearson’s  $r$  statistic [33]; for nominal attributes, we applied a Cochran-Mantel-Haenszel test. When one attribute was continuous and the other nominal, we discretized the continuous attribute to five levels using agglomerative clustering. In all cases, experiments involving Question entities had a sample size greater than 50k, while those involving Answer entities had samples of over 100k. Given these large samples,  $p$ -values for even the smallest effect sizes were significant, so we focused on associations with marginal effect sizes greater than 0.1 (the effect size for both statistics can be measured on a scale of 0.0–1.0).

Of the 57 attribute pairs, 20 exhibited a marginal association greater than 0.1. Of these, 16 (28%) demonstrated a strong reduction in the size of effect when blocking, suggesting a dependence structure similar to the model found in Figure 5.2c (albeit with a latent  $Z$ ). For instance, Question Score and View Count exhibit an effect size of  $r^2 = 0.51$  in the marginal case, but this drops to 0.12 in the conditional case (the associated  $z$ -scores are 214.16 and 59.49, respectively; both  $p$ -values are significant at the  $1 \times 10^{-8}$  level). This result suggests that while Score and View Count are associated, latent attributes on the Question author (e.g., expertise, writing style) are a common cause for both and explain most of the variation.

Four attribute pairs exhibited little change in effect size when blocking was applied, which provides evidence for the model in Figure 5.2a. For instance, the Score of a provided Answer is highly associated with Accepted status. Authors of Stack Overflow Questions can optionally “accept” a good Answer from among those provided; since many choose to accept the one with the highest score, this result is not surprising.

## 5.5 Blocking in many-to-many data

In the previous sections, blocking has been applied exclusively to one-to-many domains. While one-to-many relational data sets form natural, hierarchical blocks, the technique can be easily extended to many-to-many domains as well. In the many-to-many case, each block is determined by a *set* of parent entities rather than a single parent entity. Figure 5.9 illustrates the process for a bipartite domain consisting of four blocking entities and ten child entities. Since the set of parents is constant across each block, the influence of all attribute values (and interactions between them) can be controlled.

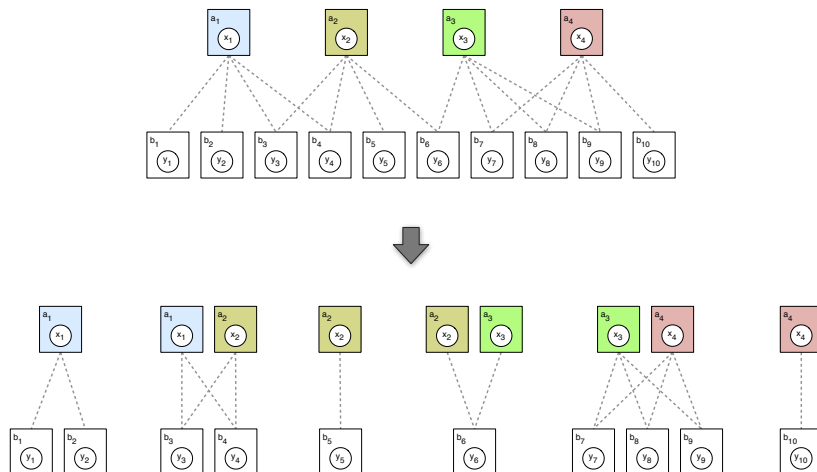


Figure 5.9: In many-to-many domains, relational blocks are determined by sets of parent entities rather than a single parent entity.

Figure 5.10 illustrates the effectiveness of relational blocking on synthetic many-to-many data. For these experiments, “small-world” bipartite graphs were generated using a version of the *stochastic copying model* [52] that was adapted for bipartite data. Each graph is made up of 10k nodes. Parent nodes have a degree uniformly chosen from  $[2, 10]$ , and child entities have a degree from  $[1, 3]$ . In the common case (a), many-to-many blocking is able to correctly adjust for the influence of parent attributes at levels comparable to traditional conditioning (albeit with slightly more

variability). However, since many-to-many blocking divides the data up in to even smaller blocks than single-parent blocking, statistical power does suffer slightly. The plot in Figure 5.10b depicts power curves for the case where child variable  $X$  has a direct influence on  $Y$  ( $Y = \beta X + \epsilon$ ). At lower strengths of effect ( $\beta < 0.2$ ), the risk of committing a Type II error is increased.

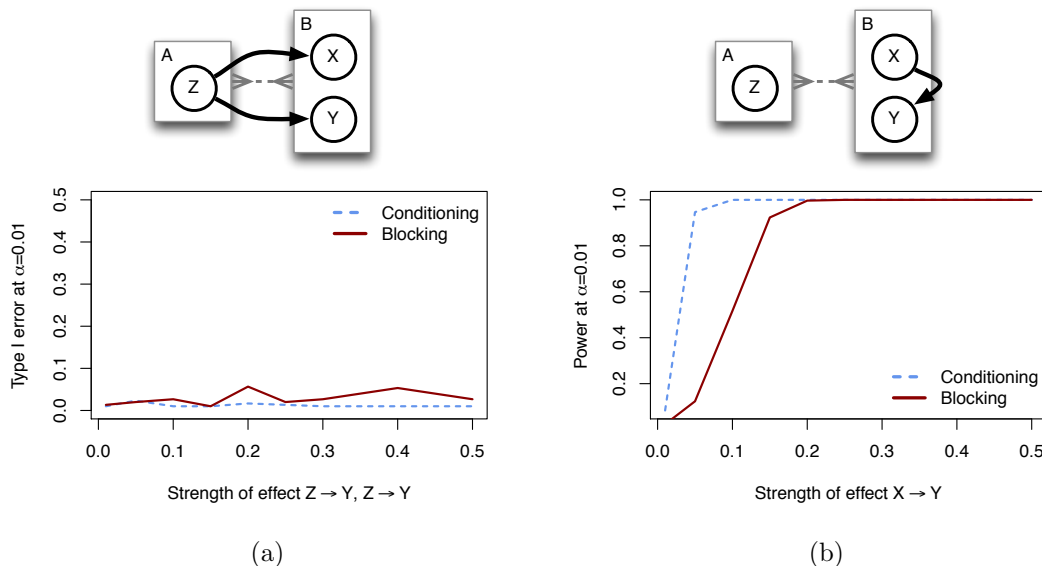


Figure 5.10: (a) Many-to-many blocking is able to adjust for common causes as effectively as traditional conditioning. (b) Since many-to-many blocking subdivides the data into small groups, statistical power decreases at low strengths of effect.

## 5.6 Less is more: Sampling for power

As defined here, relational blocking does not utilize all of the data instances. When calculating Guo’s weighted Pearson’s  $r$  or a 3D  $\chi^2$ , blocks with fewer than three instances are discarded. In this way, blocking can be thought of as a form of targeted sampling. Traditionally, there exists a tradeoff between sample size and statistical power [14]. In the relational domain, however, sampling (using fewer data

instances) may actually increase power. Below, we describe two scenarios where targeted sampling can be used in this way.

The first scenario is akin to the example presented involving Wikipedia in Section 5.1. We're given a bipartite data set connected in a one-to-many fashion. The child entities have two associated variables,  $X$  and  $Y$ , which are marginally associated. As discussed previously, relational blocking can be used to determine whether the relationship between  $X$  and  $Y$  is directly causal, or their association is due to some latent common cause  $Z$  associated with the parent entity. Graphical models describing these two situations are depicted in Figure 5.11a and b.

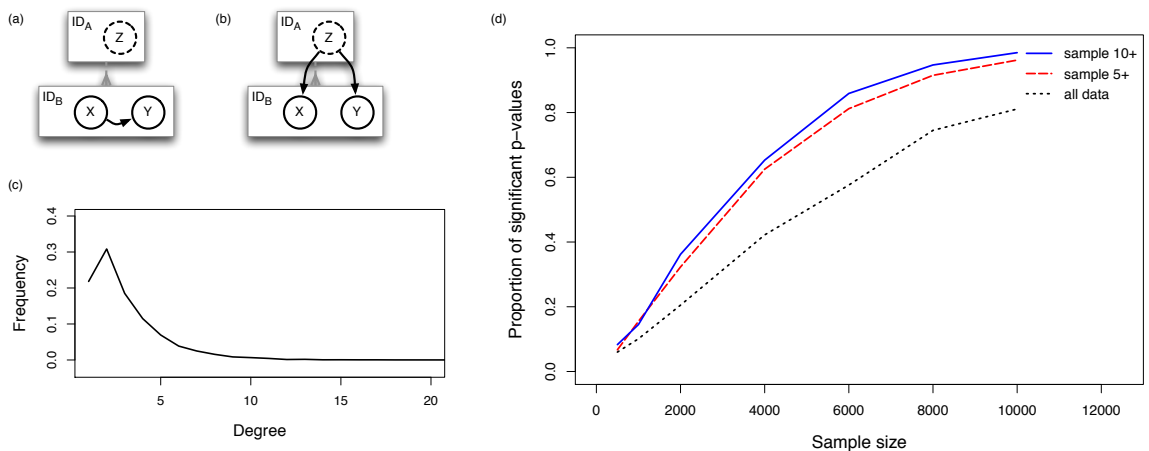


Figure 5.11: Targeted sampling can be used to increase statistical power. (a) and (b): Graphical models representing one-to-many relational data where  $X$  and  $Y$  are causally dependent (a), or marginally associated due to a common cause (b). For data with an exponential degree distribution (c), power can be increased by only considering large blocks when performing a conditional test (d).

In the previous sections, if  $X$  and  $Y$  were found to be conditionally independent when conditioned on parent entity using blocking, we used that as evidence to conclude that the model (b) was the correct model. However, it may be possible that the causal dependence between  $X$  and  $Y$  is direct but very weak, and that the blocking test was unable to disprove the null hypothesis due to insufficient statistical power.

By “slicing” the data too thinly, the association may be undetectable within any given block, and therefore undetectable overall.

Figure 5.11 shows the results of using blocking in such a scenario. Here, data are generated with the model (b), using an exponential degree distribution with a mean degree of 3, depicted in the degree distribution plot in (c). The dependence between  $X$  and  $Y$  is very weak:  $Y = 0.05X + \epsilon$ , where  $X$  and  $\epsilon$  are normally distributed with mean 0.0 and standard deviation of 1.0. Power curves for samples drawn from a one-to-many network composed of  $\sim 270\text{k}$  nodes are shown in Figure 5.11d. Since many of the blocks are small (less than three instances), blocking produces a false negative rate of over 50% for sample sizes of less than 5000 instances. If we selectively sample, however, by targeting only large subgraphs, we can increase power dramatically, as shown by the power curves corresponding to blocking using only blocks with more than 5 or 10 instances. Thus, by ignoring data instances that are part of small blocks, we can actually *increase* power.

A second scenario where targeted sampling can increase statistical power matches the one discussed in Chapter 3. Again, we have a bipartite, one-to-many relational data set; however, in this case, the variables of interest are associated with the parent and child entity, respectively. We generated synthetic graphs of 370 nodes using the models shown in 5.12a and b along with degree the distribution depicted in Figure 5.11c. For the networks generated under model a,  $Z = 0.05 * AVG(X) + \epsilon$ , for model b,  $X = 0.05 * Z + \epsilon$ , where  $\epsilon$  is distributed as  $N(0, 1)$ . In this scenario, both replication and aggregation are appropriate for propositionalization; results for both methods are shown in Figure 5.12.

For data generated under the model in Figure 5.12a, the effects of targeted sampling are dramatic for both aggregation (c) and replication (e). Here, each parent entity may be linked to several child entities. As a result, there may be competing influence among several  $X$  values on a single  $Z$ , rendering the association difficult to

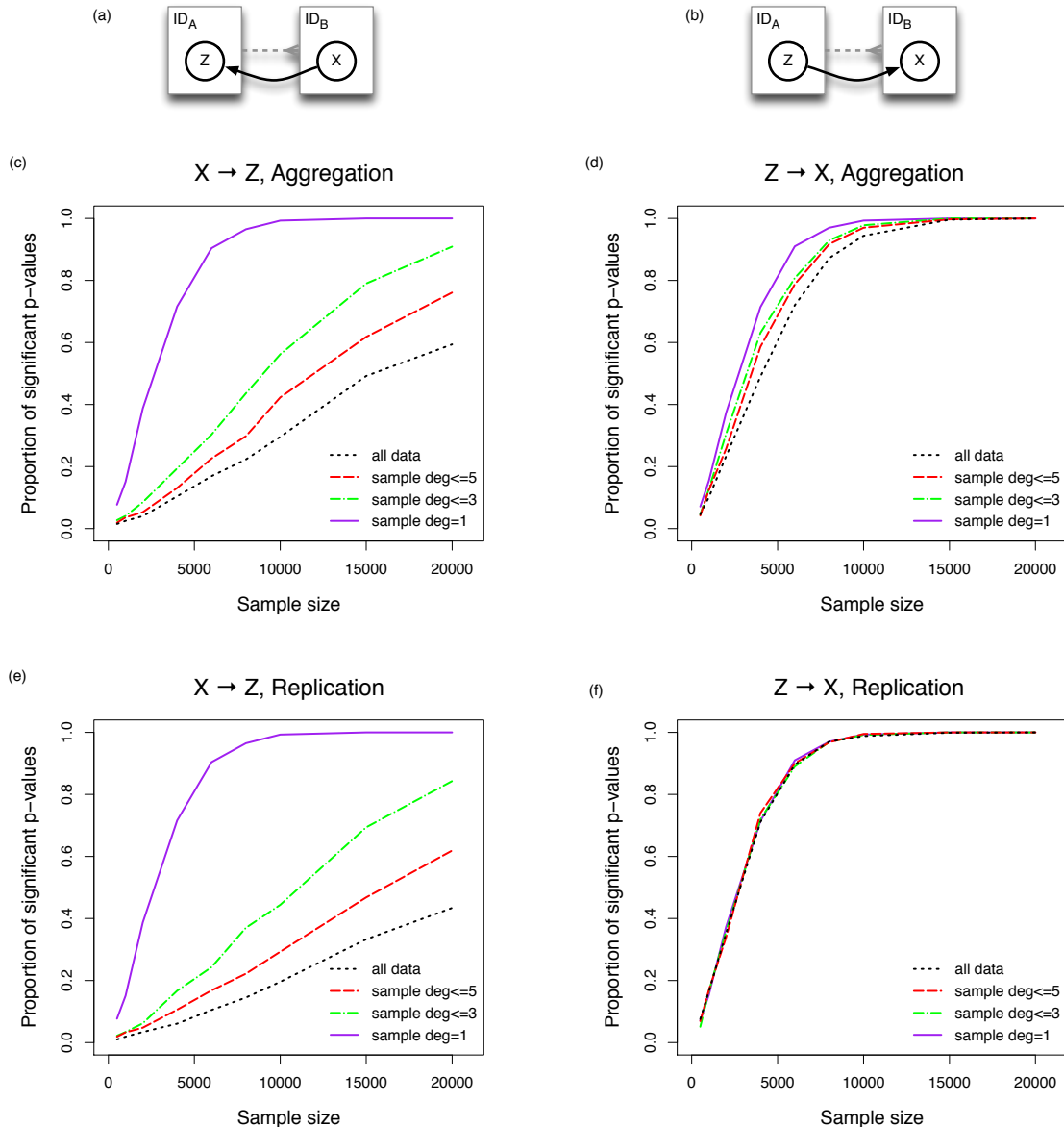


Figure 5.12: The effects of targeted sampling on parent-child attributes are quite pronounced for data generated under model (a) for both aggregation (c) and replication (e). Under model (b), the effects are greatly reduced for aggregation (d), and non-existent for replication (f).

detect. Unlike the scenario presented above, in this case, we can benefit by sampling only *small* subgraphs and ignoring larger ones. By doing so, the effects of competing influence are eliminated, resulting in a substantial increase in statistical power for equivalent sample sizes.



The power benefits of targeted sampling for data generated with model the model in Figure 5.12b are relatively slight in the aggregation case (d), and non-existent for replication (f). However, it is important to note that targeted sampling does not decrease power when compared to testing on all the data, so given a case where the causal direction cannot be established, the use of the technique has substantial upside with little downside.

Of course, the validity of this technique hinges on the assumption that there is no degree disparity present; that is, link existence is marginally independent of both variables of interest. Otherwise, sampling based on degree introduces a potential bias and conclusions that are invalid for the data as a whole.

## 5.7 Discussion

In the presence of common causes (such as in data described by the model in Figure 5.1), blocking serves essentially the same function as simple conditioning. In both cases, confounding factors are held constant within the block or conditioning group. Conditional hypothesis tests allow us to evaluate the dependence between two child entity variables while adjusting for any common causes associated with the parent entity.

As demonstrated experimentally in Section 5.3.2, the effects of blocking and conditioning are quite different for the common effect case. Conditioning on  $Z$  in model d may render (marginally independent) variables  $X$  and  $Y$  conditionally dependent. Blocking on  $A$  entities, however, will not result in any conditional dependence. This result may strike some as non-intuitive, especially given the fact that blocking, as defined above, is algebraically equivalent to conditioning on an *ID* variable. Below, we attempt to shed some light on the differences between blocking and conditioning and provide both an intuitive and theoretical explanation of why these operators behave differently in the presence of common effects.

## Berkson’s paradox, selection bias, explaining away

The concept of marginally independent events being rendered conditionally dependent given information about a common effect is relevant to data analysis in many fields, including statistics, artificial intelligence, and economics. The terms “Berkson’s Paradox”, “selection bias”, and “explaining away” describe essentially the same statistical phenomenon, although the context in which each is most commonly used varies.

Joseph Berkson formulated this idea in terms of binary treatments and outcomes in a medical setting (hence its subsequent naming) [7]. When performing inference in graphical models, AI practitioners refer to a decrease in a conditional probability of an independent causal factor due to information about a shared effect as “explaining away” [78]. For example, the presence of a cat burglar and an earthquake are marginally independent, but in the presence of an active security alarm, knowledge that an earthquake has not occurred dramatically increases the probability that a burglary has occurred. In statistics, “selection bias” refers to situations in which the population being studied does not accurately reflect the population at large [79]. Thus given a sample based on a common outcome (e.g., college students, cancer patients, self-selected survey participants), independent causal factors may appear dependent.

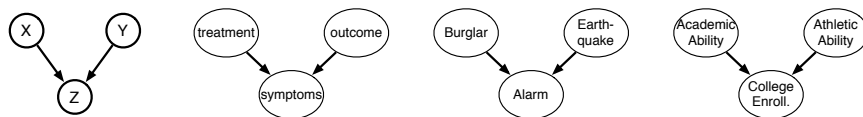


Figure 5.13: Examples of common effect cases in different domains. In each, two marginally independent factors ( $X$  and  $Y$ ) can be rendered conditionally dependent when conditioning on a common effect  $Z$ .

All of the above scenarios can be described by the simple v-shaped graphical model in Figure 5.13 below, and the semantics of d-separation agree with empirical results found in real data: In cases where independent factors  $X$  and  $Y$  are both causes of

a third variable  $Z$ , conditioning on  $Z$  (or sampling based on its value) will render  $X$  and  $Y$  (conditionally) dependent.

### Berkson's paradox in relational data

As described above, the effects of Berkson's paradox are found in relational domains as well. Consider a data set shown in Figure 5.14. In this scenario, conditioning on common effect  $Z$  may induce a conditional dependence between two (marginally independent) variables  $X$  and  $Y$ ; that is,  $X \perp\!\!\!\perp Y$ , yet  $X \not\perp\!\!\!\perp Y \mid Z$ . It should be noted that this effect is independent of the method we use for propositionalizing the data: In the replication case  $X \not\perp\!\!\!\perp Y \mid Z$ , while in the aggregation case  $f(X) \not\perp\!\!\!\perp g(Y) \mid Z$  for some aggregators  $f$  and  $g$ . Since blocking is meaningless when aggregations are used (each grouping will contain only one entity), we will focus exclusively on the replication case below.

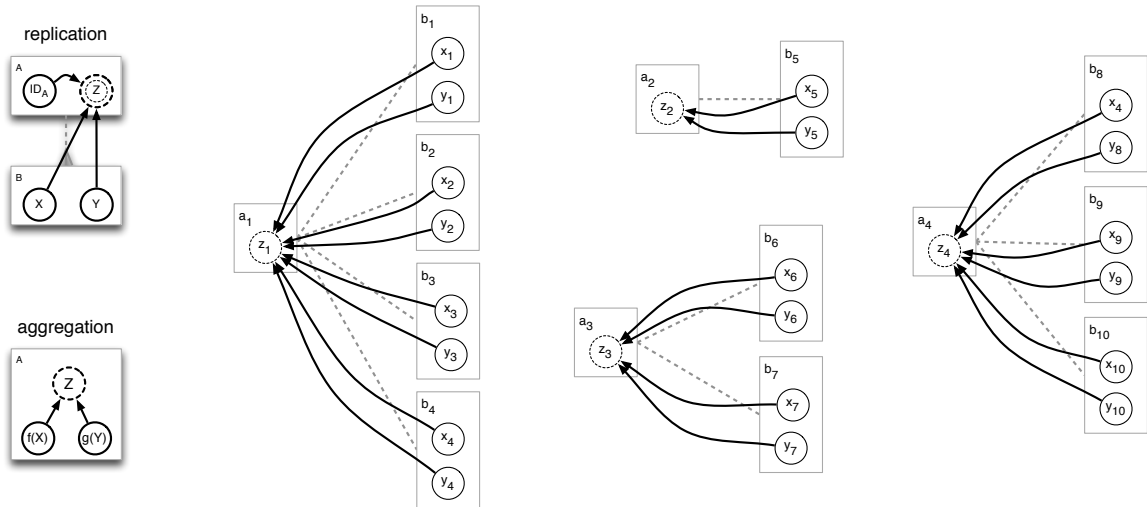


Figure 5.14: DAPER model graphs and ground graph for the common effect case. Although  $X$  and  $Y$  are marginally independent, conditioning on  $Z$  will activate a d-connecting path and may render them conditionally dependent for both replication or aggregation to propositionalize.

## Illustrative example

Consider a college lecture course consisting of exactly 2000 students (it is a rather large class). The lecture hall is divided up into 200 rows of 10 students each (it is a rather oddly shaped classroom). The students are seated randomly within the classroom, and each student has two associated binary attributes:  $S$ , which signifies whether he or she is “smart”, and  $W$ , whether he or she is “hard-working”. These two factors are considered independent.

The professor is a believer in the power of collaborative learning, and decides to divide the class into study groups by row. Examining the relationship between  $S$  and  $W$  conditioned on row can be used to factor out confounding factors associated with, for instance, the position of the row in the classroom or the proximity to distractions. However, since  $S$  and  $W$  are intrinsic, unchanging attributes, row membership has no effect on their values, and since  $S$  and  $W$  played no part in seating choice, within each row the values of  $S$  and  $W$  remain independent. Furthermore, each row is a random sample of the population at large, with (per the central limit theorem) means that are normally distributed around the population mean.

The professor decides that each row will complete all assignments as a group, and the group will receive a single grade  $G$ . The grade is a random variable collectively determined by the values of  $S$  and  $W$  for each student and follows a normal distribution (for example,  $G = C_s + C_w + \epsilon$ , where  $C_s$  and  $C_w$  are the counts of the students with  $S = True$  and  $W = True$ , respectively). The DAPER model in Figure 5.15 describes this scenario. Each (parent entity) row gets a grade  $G$  that is dependent on attributes  $S$  and  $W$  of the (child entity) students.

After the first assignment is completed, the professor decides to construct new study groups by combining different row-based groups. She (perhaps curiously) decides to combine all groups that received the same letter grades on their assignment. For example, all rows that received an “A” compose a group, all rows that got a “B”

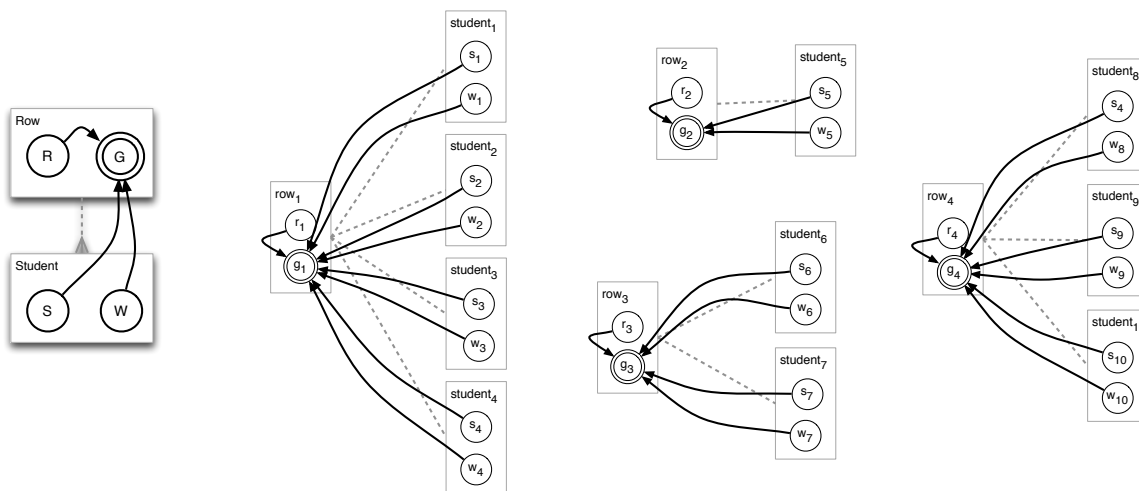


Figure 5.15: The rules of deterministic d-separation agree with empirically derived independence relationships. (left) DAPER model describing the classroom example. (right) Ground graph for the classroom data. While conditioning on  $G$  enables a path from  $S$  to  $W$ , blocking (conditioning on row  $R$ ) does not.

compose a new group, etc. After doing so, the professor performs a statistical test to ascertain the (in)dependence between  $S$  and  $W$  within each multi-row group, and finds that the two are in fact dependent!

Contingency tables for a synthetically-generated data set that matches the grade example are shown in Table 5.3. Note that while only one of the five grade groups (“F”) exhibits significance at the 0.05-level, the overall CMH test statistic is significant at the 0.001 level ( $CMH = 11.20, p = 0.0008$ ). Also note that since the grades for each row are normally distributed (with mean letter grade “C”), the grade-based groups have different sizes.

In the above example, conditioning by row-based groups corresponds to relational blocking, while the combined grade-based groups correspond to conditioning on the grade variable  $G$ . While conditioning induces a dependence between  $S$  and  $W$ , blocking by rows does not ( $CMH = 2.67, p = 0.1020$ ). This difference can be explained by the fact that in the case of blocking, the selection being performed (dividing randomly-

**Grade “A”**

	$S = +$	$S = -$	
$W = +$	0	7	7
$W = +$	24	119	143
	24	126	150

$\chi^2 = 1.40, p = 0.2370$

**Grade “B”**

	$S = +$	$S = -$	
$W = +$	9	47	56
$W = +$	99	295	394
	108	342	450

$\chi^2 = 2.20, p = 0.1376$

**Grade “C”**

	$S = +$	$S = -$	
$W = +$	28	73	101
$W = +$	237	392	429
	265	465	730

$\chi^2 = 3.73, p = 0.0534$

**Grade “D”**

	$S = +$	$S = -$	
$W = +$	36	57	93
$W = +$	150	187	337
	156	244	430

$\chi^2 = 1.00, p = 0.3175$

**Grade “F”**

	$S = +$	$S = -$	
$W = +$	27	33	60
$W = +$	112	68	180
	139	101	240

$\chi^2 = 5.48, p = 0.0193$

Table 5.3: Contingency tables and chi-square results for each grade group in the classroom example. While only one group (“F”) shows a significant dependence between  $W$  and  $S$ , the data set as a whole is significant when tested with a CMH statistic ( $CMH = 11.20, p = 0.0008$ ).

seated students up into rows) is free of bias. Since each row group was selected from the population without regard to variables  $S$  and  $W$ , they remain independent within each row (as they do in the population).

In the conditioning case, however, the grade-based groups are assembled using biased selection. Here, groups are determined by a row-based grade  $G$ , which (by virtue of the fact that  $S \rightarrow G$  and  $W \rightarrow G$ ) means that each group is a biased sample of the population with regards to  $S$  and  $W$ . Even though they are composed of unbiased subgroups (rows), the overall group is a biased sample of the population with regard to  $S$  and  $W$ . This fact is clearly illustrated in the bar graphs in Figure 5.16

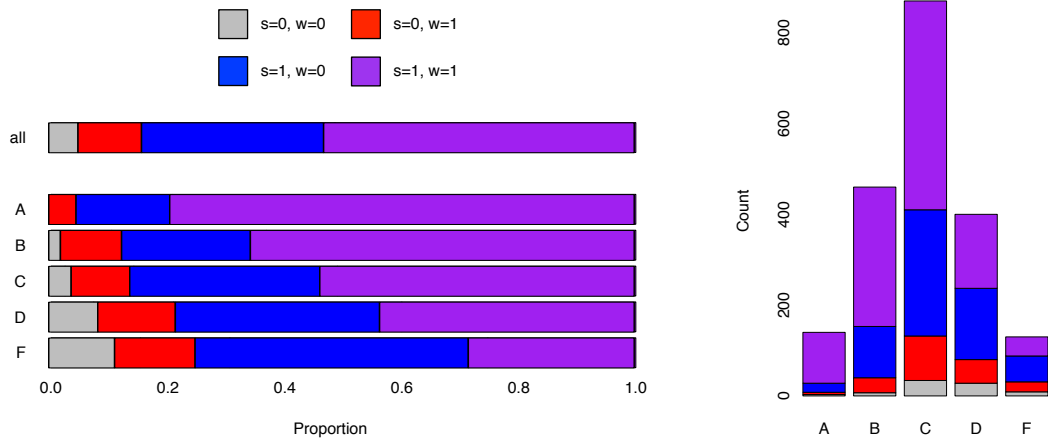


Figure 5.16: When row groups are combined according to grade, the new groups are no longer representative of the overall population. (left) Comparison of relative attribute distributions for each grade group. (right) Absolute population distributions for each grade group.

(left), which shows the proportion of students having each combination of attribute values for each grade group as compared to the overall population. Clearly, the attribute distribution of each grade-based group does not reflect the population at large. On the right, we show the absolute distribution within each grade group.

Lastly, the difference between these two operations is born out by the semantics of d-separation as well. Figure 5.15 depicts the DAPER model and ground graph for the classroom example. In the latter, conditioning on the collider  $Z$  will enable the path between  $X$  and  $Y$ . However, conditioning on row (blocking) does not, and  $S$  and  $W$  remain (conditionally) independent.

## 5.8 Conclusions

In this chapter, we have presented relational blocking as a technique to facilitate learning the structure of causal models. Blocking is similar in function to simple conditioning in its ability to reduce variability and increase statistical power. However,

unlike conditioning, blocking does not induce dependence when accounting for common effects. Blocking is able to adjust for whole classes of confounders simultaneously, whether observed or latent, effectively relaxing the causal sufficiency assumption and strengthening causal conclusions.

We have illustrated the use of blocking using synthetic data and found our approach to perform well in terms of Type I and Type II error. Furthermore, by explaining our results using the graphical models framework and d-separation criteria, we are able to provide a theoretic understanding of a commonly used technique employed in the social sciences. In addition, we have demonstrated the utility of blocking on two real world data sets.



## CHAPTER 6

### AUTOMATED IDENTIFICATION OF RELATIONAL MARKOV EQUIVALENCE CLASSES

The propositionalized models presented in prior sections often represent a subset of the possible causal structures for the data described. Other, possibly more complex models may also fit the data. For example, models  $H_1$  and  $H_2$  in Figure 6.1 depict two of the hypotheses for explaining dependence in replicated data as presented in Figure 3.7. Using a single test of conditional independence between  $ID$  and  $Y$  conditioned on  $X$ , we were able to differentiate between the two. Models  $H_3$  and  $H_4$  correspond to alternative hypotheses that also explain an association between  $X$  and  $Y$ . Model  $H_3$  is identical to model  $H_1$ , but the direction of causality between  $X$  and  $Y$  is reversed. In model  $H_4$ ,  $Y$  is causally determined by  $X$  as well as a latent variable  $Z$ .

As demonstrated previously, attributes derived from relational structure can be used to differentiate between causal models with conditional independence tests. Model  $H_3$  can be differentiated from model  $H_1$  by testing to see whether  $ID \perp\!\!\!\perp Y \mid X$  holds. However, since  $Z$  is a latent variable (and therefore cannot be explicitly included in any hypothesis test), models  $H_2$  and  $H_4$  cannot be differentiated through hypothesis testing, as the same set of testable conditional independence relationships hold in each:  $ID \not\perp\!\!\!\perp X$ ,  $ID \not\perp\!\!\!\perp X \mid Y$ ,  $ID \not\perp\!\!\!\perp Y$ ,  $ID \not\perp\!\!\!\perp Y \mid X$ ,  $X \not\perp\!\!\!\perp Y$ .

Previous work has demonstrated that the semantics of d-separation can be used to group graphical models that incorporate both latent and measured variables into equivalence classes, such that each class is defined by a set of conditional independence

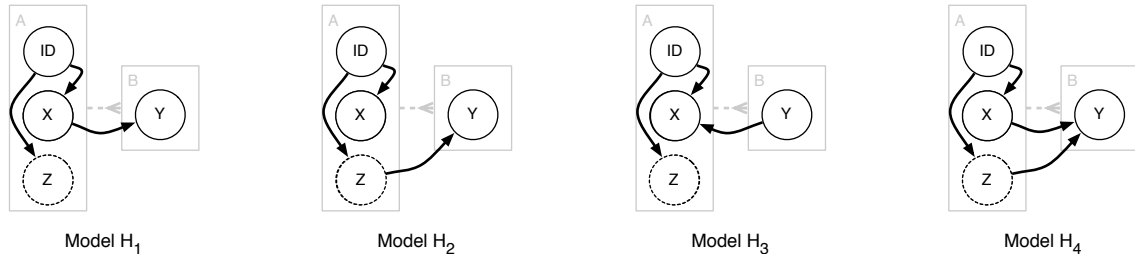


Figure 6.1: Alternative hypotheses ( $H_3$ ,  $H_4$ ) to those presented in Figure 3.7 ( $H_1$ ,  $H_2$ ). In all four,  $X$  and  $Y$  are marginally dependent, but the causal structures behind the associations differ. Given that  $Z$  is a latent variable, there are no conditional independence tests that can differentiate between  $H_2$  and  $H_4$ .

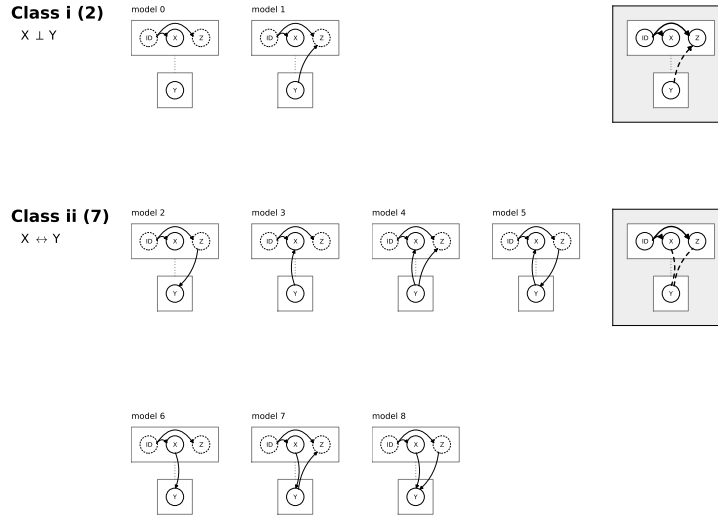
relationships among variables [87, 86, 81]. Models in the same class are said to be “Markov equivalent,” and cannot be distinguished through conditional independence tests alone. By examining these equivalence classes, we can clearly see the power of using relational data for causal learning.

Figure 6.2 depicts two sets of algorithmically-generated<sup>1</sup> Markov equivalence classes for the replicated one-to-many data discussed in Section 3.4. For simplicity, we restrict our discussion to models where  $X \perp\!\!\!\perp Z \mid ID$  and  $ID$  is not directly related to  $Y$ , along with the previous assumption that there is a causal dependence between  $ID$  to  $X$  and  $ID$  to  $Z$ . In addition, the  $ID$  variable cannot be used to condition, as the one-to-many structure of the data will necessarily zero out any test statistic (e.g., chi-square, Cochran-Mantel-Haenszel) calculated with  $X$  when conditioned on  $ID$ . Also, since  $Z$  is latent, it cannot be tested for dependence or used for conditioning.

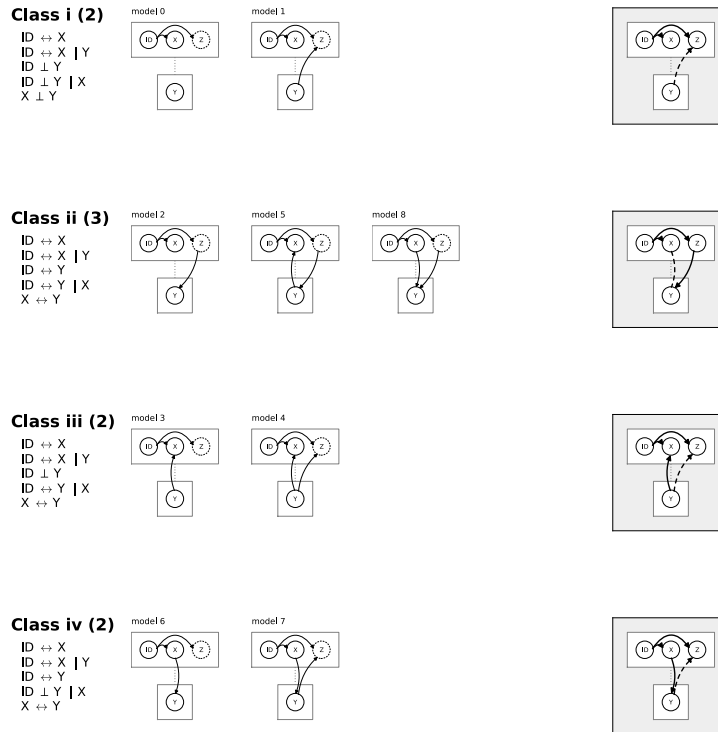
Given the above constraints, there are nine possible models, seven of which produce marginal dependence between  $X$  and  $Y$ . The shaded boxes depict a summary model for each class. For the summary models, solid directed arrows represent edges that are common to all models in that class. Undirected arrows represent edges that

---

<sup>1</sup>The code used to generate these examples if available from: <https://github.com/mattratt/CausalRelational>



(a)



(b)

Figure 6.2: D-separation equivalence classes for one-to-many data propositionalized through replication. Gray boxes contain summary graphs for each class, where solid lines represent edges shared by all models of the class, dashed lines represent edges shared by some models, and arrowheads are present where direction is consistent among the models having the edge. The  $\perp$  symbol stands for conditional independence, while the  $\leftrightarrow$  symbol stands for the converse.

exist in each class model, but may differ in direction. Dashed lines represent edges that exist in some class models but not others, and are directed if the direction is consistent when it does exist.

The first set of equivalence classes (a) illustrates the possible differentiations between models when relational structure (as captured by the *ID* variable) is not available for hypothesis testing. Here, there is only a single test available (assessing the marginal dependence of  $X$  and  $Y$ ), separating the models into two classes. When relational information is included, however, we can differentiate much more precisely (b). Here, the models separate into four equivalence classes.

Assuming both causal sufficiency and faithfulness, two aspects of the relationally-derived classes are worth noting. First, models 2 and 8 (identical to  $H_2$  and  $H_4$  mentioned above) are indistinguishable through conditional independence testing. That is, autocorrelation among values of  $Y$  on child entities will mask any causal effect flowing between  $X$  and  $Y$ . The task of differentiating homophily from influence in social networks is an ongoing challenge, these classes represent a first step toward distinguishing between the two effects as well as explaining why they are difficult to tease apart.

Second, Classes iii and iv constitute a new opportunity for causal edge orientation in relational data. Using the set of conditional independence relations, we can distinguish cases where  $X$  causes  $Y$  from those where  $Y$  causes  $X$ . Intuitively, a lack of autocorrelation among  $Y$  values (expressed as a marginal independence between *ID* and  $Y$ ) allows us to conclude that influence flows from  $Y$  to  $X$ , whereas cases where  $X$  causes  $Y$  will create autocorrelation.

For data propositionalized through aggregation, the results are similar. Figures 6.3 and 6.4 depict the classes derived from traditional and relational methods, respectively. For each, we assume that in all models,  $Y$  causes the aggregation  $f(Y)$ ,  $f(Y)$  does not cause any other variable,  $Y \perp\!\!\!\perp deg \mid X$ , and  $X \perp\!\!\!\perp f(Y) \mid deg$ . In Figure 6.3,

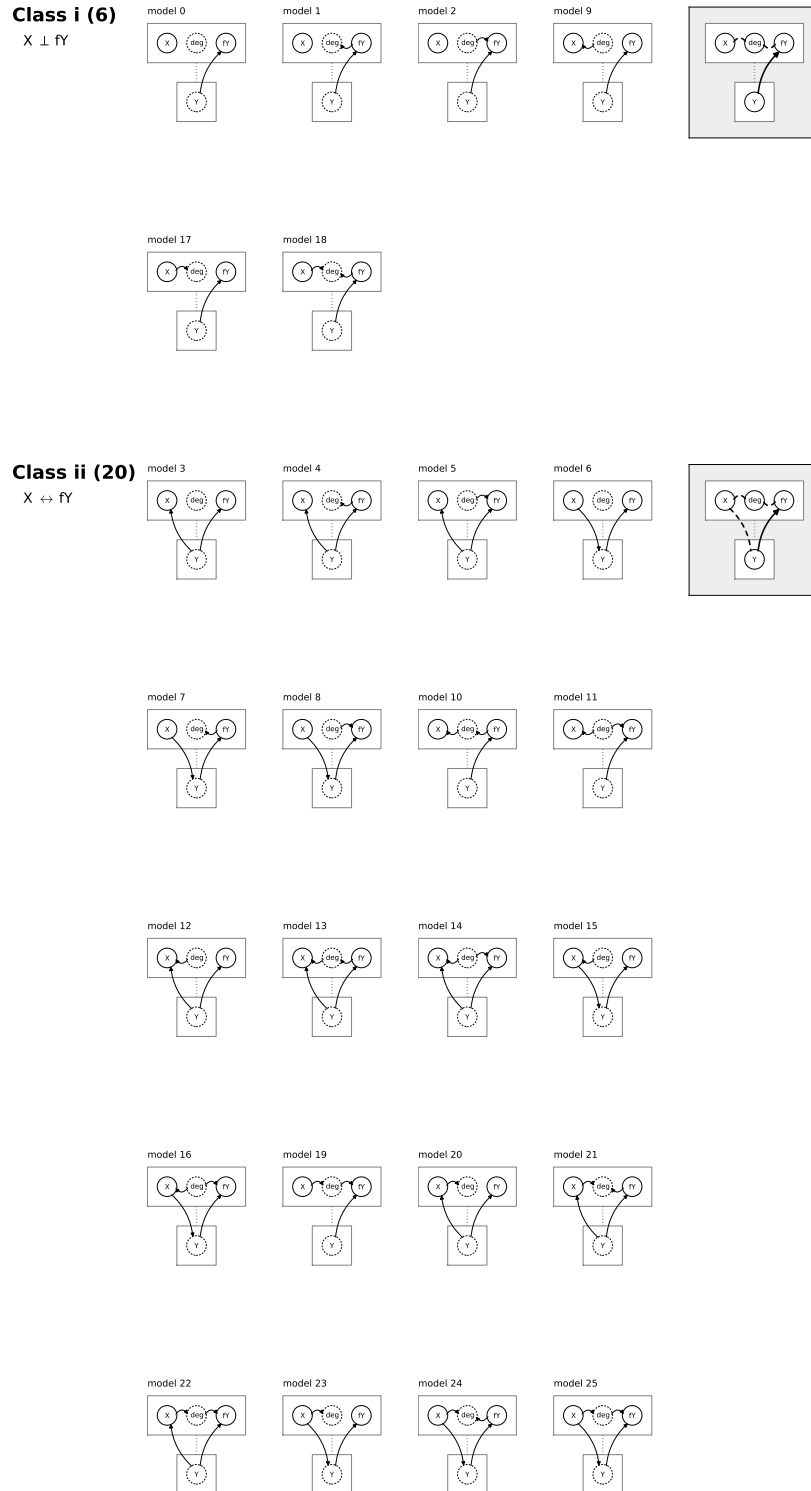


Figure 6.3: Markov equivalence classes for one-to-many data propositionalized through aggregation and separated without the use of relational degree information.

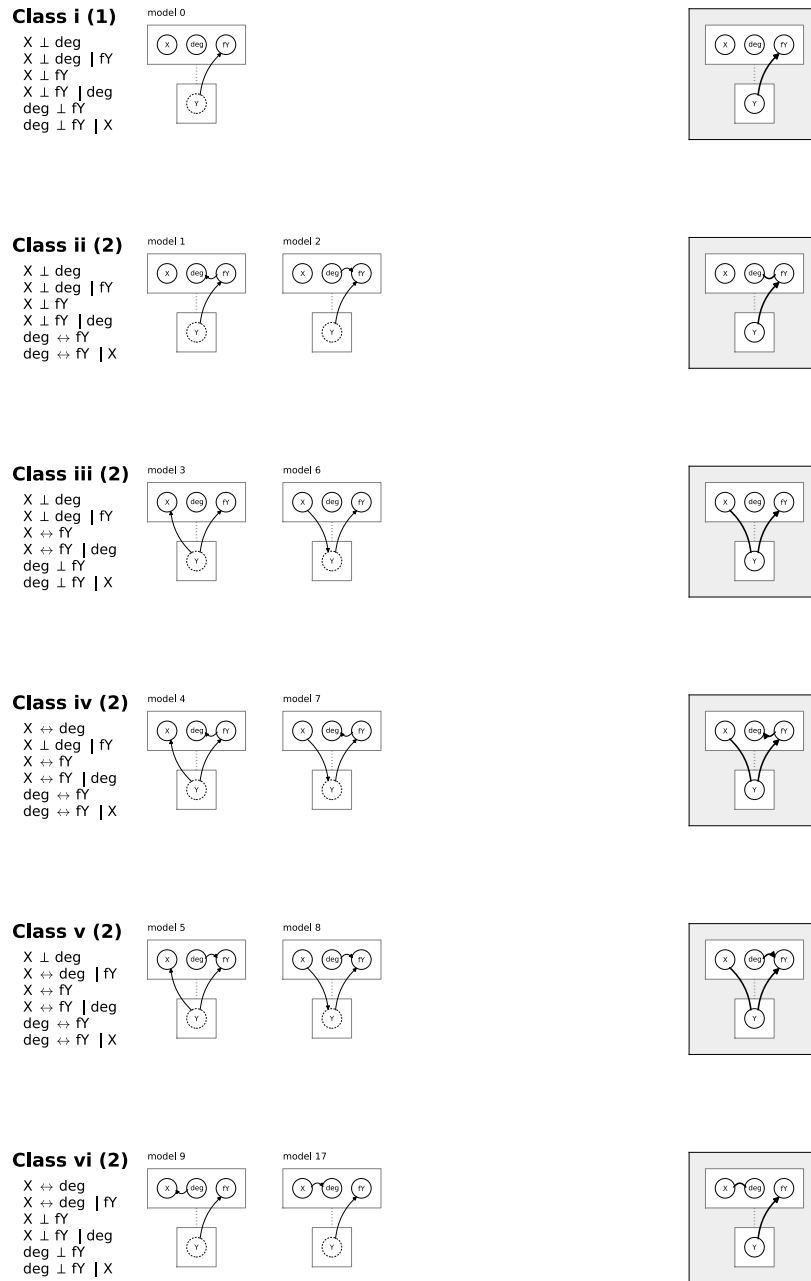


Figure 6.4: Markov equivalence classes for one-to-many data propositionalized through aggregation using degree information.

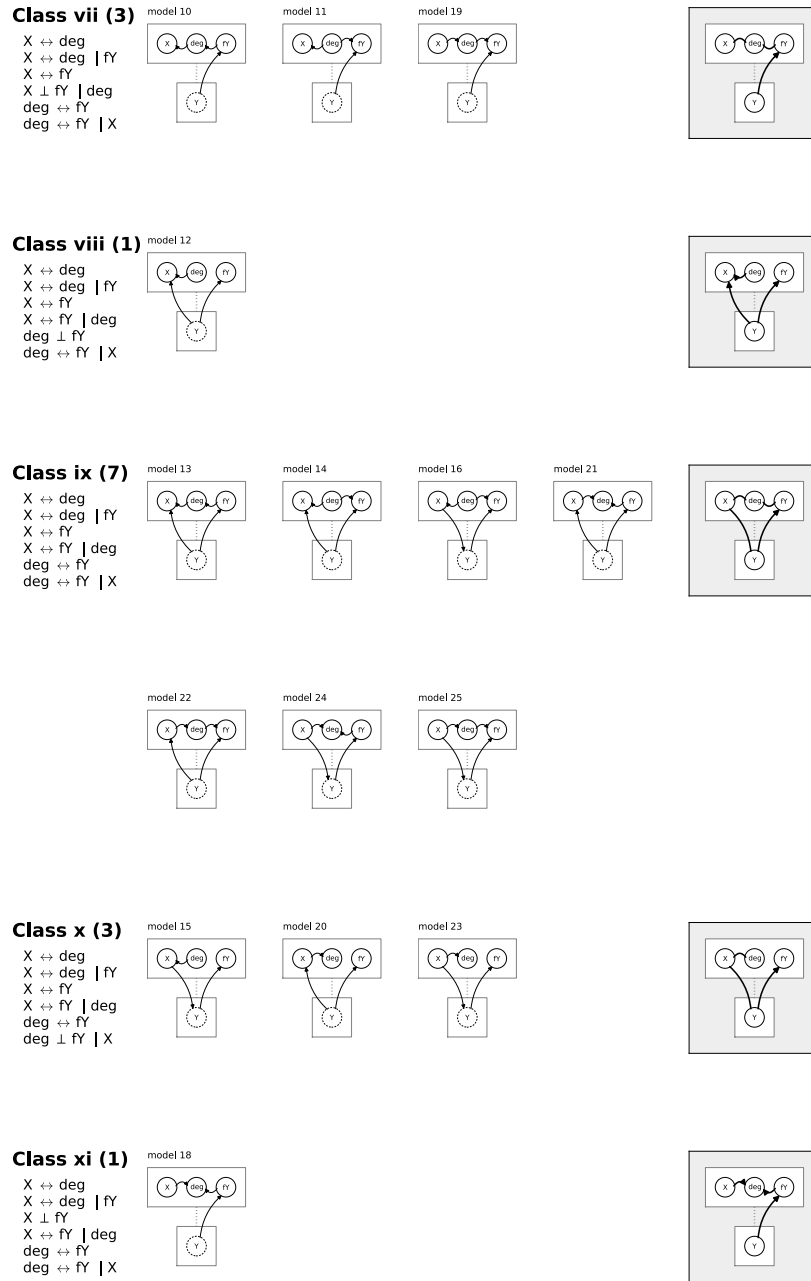


Figure 6.4: Markov equivalence classes for one-to-many data propositionalized through aggregation using degree information, cont'd.

we see that conventional, propositional methods are only able to separate the 26 possible models into two equivalence classes. By leveraging relational information in the form of a degree variable, however, we are able to further partition the model space into eleven separate classes. The relational classes are able to definitively identify the existence of a causal edge (in terms of existence, rather than direction), as indicated by the lack of dashed edges in the class summary graphs.

The relational Markov equivalence classes shown above were programmatically generated. Given a relational schema, each possible DAG is constructed and analyzed using the d-separation criteria. DAGs are then grouped according to their implied conditional independence relationships, and the class summary graphs are constructed by inspecting the edges present across the models of each class.

## 6.1 Discussion

Traditionally, the rules of d-separation are applied manually, allowing a practitioner to make decisions about the space of hypotheses through inspection alone. For the simple domains presented above, such automated schema inspection may not be necessary. However, the space of possible DAGs is exponential in the number of possible edges, which itself is  $O(V^2)$ . Thus, for more complex domains, manual construction is effectively impossible. For instance, the illustrative example schema involving journals, papers, and authors from Section 2.2.1 produces 1,656 valid DAGs that partition into 1,124 equivalence classes.

At a higher level, the automated schema analysis technique presented here is clear demonstration of benefits of relational representations. When relational information is made available to learning algorithms, they can model data with much greater specificity than algorithms that rely on attribute data alone. While the addition of relations adds some amount of complexity to analysis, it can ultimately lessen the difficulty of the learning task.



## CHAPTER 7

### CONCLUSIONS AND THE FUTURE

In the preceding chapters, we attempted to bridge the gap between the fields of causal discovery and relational learning. We have shown that by incorporating the formalism of Bayesian networks and d-separation, we can better understand the causal dynamics of non-iid data sets. In addition, we have demonstrated how to leverage the expressive power of relational representations to better perform causal discovery. Below, we review the main contributions of the work.

We have formalized propositionalization as a graphical sampling procedure, and grounded that in common practice. Traditional relational algebraic approaches and SQL tend to hide the information that is lost in the transformation process. In contrast, the graphical approach clearly highlights the mechanics of replication and aggregation. We use the formalisms of DAPER models and the ground graph to explain two previously identified pathologies in relational data analysis: instance dependence bias and degree disparity bias. In addition to identifying the circumstances that can produce two types of biased analysis, we detail the statistical consequences of each.

Using Bayesian networks augmented with variables to capture relational structure, we utilize the semantics of d-separation to create new classes of hypotheses tests that differentiate between models where instance dependence and degree disparity biases may be present. For replicated domains, we introduce a nominal *ID* variable that can be incorporated into conditional independence tests; for aggregated data tables, we condition on a constructed degree variable to adjust for the sensitivity of

different aggregators to cardinality. In both cases, we show how the graphical model formalism both explains the origins of and provides a solution for common pathologies in relational learning.

We introduce relational blocking, a design that can relax the causal sufficiency assumption for relational domains. We demonstrate the use of blocking empirically and explain the results theoretically. Our analysis includes a somewhat unexpected (but theoretically justified) result: While blocking is equivalent to traditional conditioning in terms of its ability to adjust for common causes, it does not induce conditional dependence in the presence of common effects.

In addition to clarifying problems in relational learning using causal reasoning and Bayesian networks, we also demonstrate how relational representations can produce new techniques for automated causal discovery. The conditional hypothesis tests presented in Chapters 3 and 4 can be utilized to programmatically construct relational Markov equivalence classes for relational domains, which in turn can be used to infer the existence and (in some cases) the direction of causal relationships from data.

## **Future directions**

Of course, the contributions listed above are all starting points for several new avenues of research. For example, much of the work in this thesis centers around the use of the ground graph for analysis. However, for some domains, constructing the ground graph in its entirety may be infeasible. DAPER models, on the other hand, provide a compact method for describing the relations and dependencies found in a given data set. However, the rules of d-separation cannot be directly applied to DAPER. Relational causal analysis could benefit greatly from some sort of hybrid representation that could be analyzed directly with d-separation and would maintain the compactness of DAPER. Alternatively, a new set of relational graphical criteria

could be formulated such that conditional dependence relationships could be read from a DAPER model directly.

As presented here, the propositionalization process is a somewhat necessary evil. In order to leverage modern statistical techniques, network data sets must be transformed into a single table format such that much information is lost. In Chapters 3 and 4, we present methods to account for this information loss and adapt traditional hypothesis testing techniques to relational domains. Rather than change the data to suit the statistical techniques, it may be possible to modify the latter to suit the former. For instance, there currently exists no closed-form description of the distribution of statistics such as  $\chi^2$  when applied to networks; if we could accurately calculate the p-values for a given network, many of the techniques described here would be obsolete.

The use of Markov equivalence classes for edge-orientation is an example of how the expressiveness of relational data representations can be utilized to differentiate between causal models. In the statistical relational learning community, much effort has been spent on differentiating *network influence* and *homophily* [3, 53]. It may be possible to formulate this task from a causal viewpoint, where the difference between two effects is expressed in terms of edge orientation between attribute and relationship formation.

Finally, most experimental and analysis design is centered around propositional representations. Hopefully, the use of relational data representations will proliferate and more truly relational designs such as blocking will be formulated. Doing so will enable research in both causal discovery and statistical relational learning to move forward.

## BIBLIOGRAPHY

- [1] Agrawal, R., and Srikant, R. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases* (1994), vol. 1215, pp. 487–499.
- [2] Angrist, J.D., and Pischke, J.S. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 2009.
- [3] Aral, S., Muchnik, L., and Sundararajan, A. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences* 106, 51 (2009), pp. 21544–21549.
- [4] Aronson, J., Dyer, M. E., Frieze, A. M., and Suen, S. Randomized greedy matching II. *Random Structure and Algorithms* 6, 1 (1995), pp. 55–74.
- [5] Barabási, A.L. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Penguin Group, 2002.
- [6] Batty, Ryan. Keys to 2011/12 - score first and the home wins will come. <http://www.coppernblue.com/2011/8/31/2396019/keys-to-2011-12-score-first-and-the-home-wins-will-come>, August 31, 2011.
- [7] Berkson, J. Limitations of the application of fourfold table analysis to hospital data. *Biometrics Bulletin* (1946), pp. 47–53.
- [8] Bliese, P.D., and Hanges, P.J. Being both too liberal and too conservative: The perils of treating grouped data as though they were independent. *Organizational Research Methods* 7, 4 (2004), pp. 400–417.
- [9] Boomsma, Dorret, Busjahn, Andreas, and Peltonen, Leena. Classical twin studies and beyond. *Nature Reviews Genetics* 3 (November 2002), pp. 872–882.
- [10] Buntine, W. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering* 8, 2 (1996), pp. 195–210.
- [11] Caldwell, Dave. Rangers' recipe for victory: Score first. *The New York Times* (February 17, 2012).
- [12] Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., and Kleinberg, J. Mining the web's link structure. *Computer* 32, 8 (1999), pp. 60–67.

- [13] Charniak, E. Bayesian networks without tears. *AI Magazine* 12, 4 (1991), pp. 50–63.
- [14] Cohen, P.R. *Empirical Methods For Artificial Intelligence*. MIT press Cambridge, Massachusetts, 1995.
- [15] Cooper, G.F. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery* 1, 2 (1997), pp. 203–224.
- [16] Cooper, G.F., and Herskovits, E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9, 4 (1992), pp. 309–347.
- [17] Doreian, P. Causality in social network analysis. *Sociological Methods & Research* 30, 1 (2001), pp. 81–114.
- [18] Dow, M.M., Burton, M.L., White, D.R., and Reitz, K.P. Galton’s problem as network autocorrelation. *American Ethnologist* (1984), pp. 754–770.
- [19] Fast, A. *Learning the Structure of Bayesian Networks With Constraint Satisfaction*. PhD thesis, University of Massachusetts Amherst, 2009.
- [20] Fisher, R.A. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.
- [21] Fitzmaurice, G.M., Laird, N.M., and Ware, J.H. *Applied Longitudinal Analysis*, vol. 745. Wiley, 2011.
- [22] Friedkin, N.E. *A Structural Theory of Social Influence*. Cambridge University Press, 1998.
- [23] Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. Learning probabilistic relational models. *Proceedings of the International Joint Conference on Artificial Intelligence* (1999), pp. 1300–1309.
- [24] Galton, Sir Francis. Discussion on ‘On a method of investigating the development of institutions applied to laws of marriage and descent’, E. Tylor. *Journal of the Anthropological Institute* 18, 270 (1889).
- [25] Geiger, D., Verma, T., and Pearl, J. *Identifying Independence in Bayesian Networks*. Wiley Online Library, 1989.
- [26] Gelman, A., and Hill, J. *Data Analysis Using Regression and Multi-level/Hierarchical Models*. Cambridge University Press New York, 2007.
- [27] Getoor, L., Koller, D., Taskar, B., and Friedman, N. *Learning Statistical Models From Relational Data*. PhD thesis, Stanford University, 2001.
- [28] Getoor, L., and Taskar, B. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

- [29] Goldstein, H. *Multilevel Statistical Models*. Arnold London, 1995.
- [30] Greenland, S. Principles of multilevel modelling. *International Journal of Epidemiology* 29, 1 (2000), pp. 158–167.
- [31] Greenland, S., and Brumback, B. An overview of relations among causal modelling methods. *International Journal of Epidemiology* 31, 5 (2002), p. 1030.
- [32] Grefen, P.W.P.J., and de By, R.A. A multi-set extended relational algebra: a formal approach to a practical issue. In *Data Engineering, 1994. Proceedings. 10th International Conference* (1994), IEEE, pp. 80–88.
- [33] Guo, J.H. Four correlation coefficients with a third blocking variable: Their efficacy, relative efficiency, and test statistics. *Communications in Statistics: Theory and Methods* 32, 9 (2003), pp. 1835–1858.
- [34] Heckerman, D, Geiger, D, and Chickering, D M. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20, 3 (1995), pp. 197–243.
- [35] Heckerman, D., Meek, C., and Koller, D. Probabilistic entity-relationship models, PRMs, and plate models. In *Introduction to Statistical Relational Learning*. The MIT Press, 2007, pp. 201–238.
- [36] Hill, A.B. The environment and disease: association or causation? *Bulletin of the World Health Organization* 83, 10 (2005), pp. 796–798.
- [37] Holland, P.W. Statistics and causal inference. *Journal of the American Statistical Association* 81, 396 (1986), pp. 945–960.
- [38] Jensen, D., and Neville, J. Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the International Conference on Machine Learning* (2002), Morgan Kaufmann, pp. 259–266.
- [39] Jensen, D, and Neville, J. Linkage and autocorrelation cause feature selection bias in relational learning. *Proceedings of the International Conference on Machine Learning* (2002), pp. 259–266.
- [40] Jensen, D., Neville, J., and Gallagher, B. Why collective inference improves relational classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), ACM, pp. 593–598.
- [41] Jensen, D., Neville, J., and Hay, M. Avoiding bias when aggregating relational data with degree disparity. In *Proceedings of the International Conference on Machine Learning* (2003), AAAI Press, pp. 274–281.
- [42] Jensen, D., Neville, J., and Rattigan, M. Randomization tests for relational learning. Tech. Rep. UM-CS-2003-05, University of Massachusetts, 2003.

- [43] Jensen, F.V. *An Introduction to Bayesian Networks*, vol. 74. Springer, 1996.
- [44] Kalisch, M., and Bühlmann, P. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *The Journal of Machine Learning Research* 8 (2007), pp. 613–636.
- [45] Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., and Ueda, N. Learning systems of concepts with an infinite relational model. In *Proceedings of the Twentieth National Conference on Artificial Intelligence* (2006), vol. 21, AAAI Press; MIT Press, pp. 381–88.
- [46] Kenny, D., and Judd, C. Consequences of violating the independence assumption in analysis of variance. *Psychological Bulletin* 99, 3 (1986), pp. 422–431.
- [47] Kenny, D., and La Voie, L. Separating individual and group effects. *Journal of Personality and Social Psychology* 48, 2 (1985), pp. 339–348.
- [48] Kittur, A., and Kraut, R.E. Harnessing the wisdom of crowds in Wikipedia: Quality through coordination. In *Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work* (2008), pp. 37–46.
- [49] Koerner, C., and Wrobel, S. Bias-free hypothesis evaluation in multirelational domains. In *Proceedings of the Tenth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, vol. 3918. Springer Verlag, 2006, pp. 668–672.
- [50] Kramer, S., Lavrač, N., and Falch, P. Propositionalization approaches to relational data mining. In *Relational Data Mining*, Sašo Džeroski and Nada Lavrač, Eds. Springer-Verlag, New York, NY, 2001, pp. 262–286.
- [51] Krogel, M.A., Rawles, S., Železný, F., Flach, P., Lavrač, N., and Wrobel, S. Comparative evaluation of approaches to propositionalization. *Inductive Logic Programming* (2003), pp. 197–214.
- [52] Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., and Upfal, E. Stochastic models for the web graph. In *Proceedings of the Forty-First Annual Symposium on Foundations of Computer Science* (2000), IEEE, pp. 57–65.
- [53] La Fond, T., and Neville, J. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the Nineteenth International Conference on World Wide Web* (2010), ACM, pp. 601–610.
- [54] Lavrač, N., Železný, F., and Flach, P. Rsd: Relational subgroup discovery through first-order feature construction. *Inductive Logic Programming* (2003), pp. 149–165.

- [55] Maier, M, Taylor, B, Oktay, H, and Jensen, D. Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence* (2010), pp. 531–538.
- [56] McCallum, A., Wang, X., and Corrada-Emmanuel, A. Topic and role discovery in social networks with experiments on Enron and academic email. *Journal of Artificial Intelligence Research* 30, 1 (2007), pp. 249–272.
- [57] McGovern, A., Friedland, L., Hay, M., Gallagher, B., Fast, A., Neville, J., and Jensen, D. Exploiting relational structure to understand publication patterns in high-energy physics. *ACM SIGKDD Explorations Newsletter* 5, 2 (2003), pp. 165–172.
- [58] Meek, C. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (1995), pp. 403–410.
- [59] Megiddo, N., and Srikant, R. Discovering predictive association rules. In *Proceedings of the Fourth International Conference on Knowledge Discovery in Databases* (1998), pp. 274–278.
- [60] Neapolitan, R.E. *Learning Bayesian Networks*. Pearson Prentice Hall, Upper Saddle River, NJ, 2004.
- [61] Neville, J., and Jensen, D. Iterative classification in relational data. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data* (2000), pp. 13–20.
- [62] Neville, J., and Jensen, D. Leveraging relational autocorrelation with latent group models. In *Proceedings of the Fourth International Workshop on Multi-Relational Data Mining* (2005), ACM, pp. 49–55.
- [63] Neville, J., Jensen, D., Friedland, L., and Hay, M. Learning relational probability trees. In *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining* (2003), ACM, pp. 625–630.
- [64] Neville, J., Jensen, D., Gallagher, B., and Fairgrieve, R. Simple estimators for relational Bayesian classifiers. In *ICDM 2003* (2003), pp. 609–612.
- [65] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [66] Pearl, Judea. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, 2000.
- [67] Perlich, C., and Provost, F. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning* 62, 1 (2006), pp. 65–105.



- [68] Popescul, A., and Ungar, L. Structural logistic regression for link analysis. In *Proceedings of the International Workshop on Multi-Relational Data Mining* (2003), pp. 92–106.
- [69] Rabiner, L., and Juang, B. An introduction to hidden markov models. *ASSP Magazine, IEEE* 3, 1 (1986), pp. 4–16.
- [70] Ramakrishnan, R., and Gehrke, J. *Database Management Systems*. Osborne/McGraw-Hill, 2000.
- [71] Rattigan, M. J., and Jensen, D. Hypothesis testing methods for relational data. Tech. Rep. UM-CS-2009-053, University of Massachusetts, 2009.
- [72] Rattigan, M. J., Maier, M., and Jensen, D. Exploiting network structure for active inference in collective classification. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops* (2007), IEEE, pp. 429–434.
- [73] Rattigan, M.J.H. Reidentification of artists and genres in KDD Cup 2011. In *Proceedings of the 2011 Workshop on Information in Networks* (2011).
- [74] Rattigan, M.J.H., and Jensen, D. Leveraging d-separation for relational data sets. In *2010 IEEE International Conference on Data Mining* (2010), IEEE, pp. 989–994.
- [75] Rattigan, M.J.H., Maier, M., and Jensen, D. Relational blocking for causal discovery. In *Proceedings of the Twenty-Fifth National Conference on Artificial Intelligence* (2011), pp. 145–151.
- [76] Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. The author-topic model for authors and documents. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence* (2004), pp. 487–494.
- [77] Rubin, D. B. Estimating causal effects of treatments in randomized and non-randomized studies. *Journal of Educational Psychology* 66, 5 (October 1974), pp. 688–701.
- [78] Russell, S.J., and Norvig, P. *Artificial intelligence: A Modern Approach*. Prentice Hall, 2010.
- [79] Sachs, L., and Reynarowych, Z. *Applied Statistics: A Handbook of Techniques*, vol. 707. Springer-Verlag, New York, 1984.
- [80] Scheines, R. An introduction to causal inference. *Causality in Crisis?* (1997), pp. 185–99.
- [81] Scheines, R. The similarity of causal inference in experimental and non-experimental studies. *Philosophy of Science* 72, 5 (2005), pp. 927–940.
- [82] Shadish, W., Cook, T., and Campbell, D. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, Boston, MA, 2002.

- [83] Silverstein, C., Brin, S., Motwani, R., and Ullman, J. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery* 4, 2-3 (2000), pp. 163–192.
- [84] Spirtes, P., and Glymour, C. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review* 9, 1 (1991), pp. 62–72.
- [85] Spirtes, P., Glymour, C.N., and Scheines, R. *Causation, Prediction, and Search*. The MIT Press, 2001.
- [86] Spirtes, P., and Richardson, T. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics* (1996), pp. 489–500.
- [87] Spirtes, P., and Verma, T. Equivalence of causal models with latent variables. Tech. Rep. CMU-Phil33, Carnegie Mellon University, 1992.
- [88] Susser, M. *Causal Thinking in the Health Sciences: Concepts and Strategies of Epidemiology*. Oxford University Press, 1973.
- [89] Trochim, W. The Research Methods Knowledge Base, 2nd Edition. <http://www.socialresearchmethods.net/kb/>, October 2006.
- [90] Tsamardinos, I., Brown, L.E., and Aliferis, C.F. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* 65, 1 (2006), pp. 31–78.
- [91] Watts, D.J. *Six Degrees: The Science of a Connected Age*. WW Norton & Company, 2004.
- [92] Xu, Z., Tresp, V., Yu, K., and Kriegel, H.P. Infinite hidden relational models. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence* (2006), pp. 544–551.
- [93] Zellner, A. Causality and causal laws in economics. *Journal of Econometrics* 39, 1-2 (1988), pp. 7–21.