UNIVERSITY OF CALIFORNIA

Los Angeles

Graphical Models for Inference with Missing Data

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Karthika Mohan

2017

ABSTRACT OF THE DISSERTATION

Graphical Models for Inference with Missing Data

by

Karthika Mohan
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2017
Professor Judea Pearl, Chair

We address inference problems associated with missing data using causal Bayesian networks to model the data generation process. We show that procedures based on graphical models can overcome limitations of conventional missing data methods and provide meaningful performance guarantees even when data are Missing Not At Random (MNAR). In particular, we identify conditions that guarantee consistent estimation of parameters of interest in broad categories of missing data problems, and derive procedures for implementing this estimation. We derive testable implications for missing data problems in both MAR (Missing At Random) and MNAR categories. Finally, we apply these techniques to develop a suite of algorithms for closed form estimation of Bayesian network parameters.

The dissertation of Karthika Mohan is approved.

David E Heckerman

Onyebuchi A Arah

Richard E Korf

Adnan Y Darwiche

Judea Pearl, Committee Chair

University of California, Los Angeles

2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

A heartfelt thank you to Judea Pearl, the best advisor anyone could have! You provided me with an exciting, productive and non-judgmental environment to air ideas, discuss research, and argue science. I enjoyed all our forays and tangents into the worlds of philosophers, Quantum Physics and everything else in between. Your gentle, patient guidance has taught me to be a better researcher at the end of this journey. It is an honor to have learnt from you.

Thank you committee members Adnan Darwiche, Richard Korf, Onyebuchi Arah and David Heckerman for your support and encouragement. Your ready cheer and helpful advice, Adnan, was always appreciated. Rich, thank you for letting me abuse your open door policy and for your support.

Thank you Jenn Vaughan for guiding my research early on and all the advice and mentoring subsequently.

Parts of this work have benefited from discussions and collaborations with Felix Thoemmes, Arthur Choi, Ilya Shpitser, Guy Van den Broeck and Jin Tian. I enjoyed working with all of you and look forward to more collaborations in the future.

Thank you Kaoru Mulvihill for assistance with all administrative matters, and for all the help.

Finally, thank you family and friends for the unstinting faith and encouragement.

BIOGRAPHICAL SKETCH

Karthika Mohan holds a Master of Science degree in Computer Science from University of California, Los Angeles (UCLA). Prior to this, she was a student at the International Institute of Information Technology, Hyderabad (IIIT-H), India, where she completed a Master of Science degree in Information Technology. Her undergraduate degree was a Bachelor of Technology in Computer Science and Engineering. The list of publications and presentations related to her doctoral research can be viewed online at www.karthikamohan.com

CHAPTER 1

Introduction

Missing data, also known as incomplete data, are data in which values of one or more variables are not recorded. All branches of empirical sciences are plagued by missing data - sensors don't work reliably; patients are unable to recall treatments, outcomes or episodes; and respondents do not answer all questions in questionnaires. Missing data analysis is an active research area, as attested by the huge volume of published works in this field. Google Scholar, for instance, returns over 2 million results on this topic. However, despite the plethora of literature, missing data is a thorny problem that leaves many questions unanswered.

The overarching goal of this thesis is to examine the missing data problem from a causal perspective, develop model-guided estimation procedures and devise tests for detecting model misspecifications. To this end, we use causal graphs as a tool to model and analyze the missing data generation processes, in order to reduce the damage due to missingness[1]. To remedy shortcomings in the traditional theoretical framework, we identify and answer open questions in this area. In order to place these questions in context and clarify basic notations, we exemplify a missing data problem below:

Example 1. *Let D be a dataset (shown in Table 1.1) comprising variables Gender, Age and Drug Use, where Gender and Age are fully observed and Drug Use is corrupted by missing values. Let M be the causal model encoding the assumption that missingness is caused by the reluctance of teenage respondents in revealing information about their drug use. Let Q denote the query we are interested in estimating. An example of a query is the probability of drug use among women, i.e. $Q = P(Drug\,Use = true|Gender = female)$.*

[1] In missing data literature, the word 'missingness' is commonly used as the noun form of missing.

We list below a set of open questions that are of interest to researchers burdened with missing data.

Q-1 Given missing data D, the missingness model M and a target query Q, can we determine whether or not Q is *consistently*[2] estimable?

Q-2 If the answer to Q-1 is in the affirmative, then what is the procedure for computing Q?

Q-3 If the answer to Q-1 is in the negative, then what additional assumptions are necessary for consistent estimation? For instance, will it help to make parametric assumptions about the model, such as linearity? In the worst case, can we compute informative bounds on Q?

Q-4 How can we efficiently compute Q in practice given finite samples?

Q-5 Finally, how can we detect mis-specifications in the model? This is important since the answers to all questions from Q-1 to Q-4 are highly model-sensitive.

Prior to deliberating over these questions, we explain the need to analyze missing data from a causal perspective. Additionally, we *informally* introduce the notion of *Missing At Random (MAR)* [Rub76], since the existing theoretical framework of missing data is almost exclusively built around it.

1.1 The Causal Element in Missing Data Problems

Re-examine the dataset in table 1.1 that depicts samples from a drug abuse study conducted in a school. When presented with this data, one cannot but wonder why data are missing - i.e., what *caused* missingness? Is the *cause* of missingness,

1. a random computer error that accidentally deleted some values, or

[2] Consistent estimate of a target quantity Q is the estimate produced by an estimator/procedure whose estimates/outputs converge to the true value of Q as the sample size increases indefinitely. Example: sample mean of a normal random variable.

2. a function of the fully observed variables Age and Gender - e.g., teenage boys that rebelled and decided not to participate in the study, or

3. the underlying true value of the variable - e.g., students who used drugs and refused to answer questions about drug use for fear of repercussions?

Of these plausible causes, data generated by (1) and (2) belong to *missing at random* (MAR)[3] while (3) belong to *missing not at random* (MNAR) category [Rub76]. Specifically, data are MNAR when missingness in variables is caused by the underlying true value of the variables and/or by other variables that are themselves afflicted by missingness; all other missing data are MAR.

While two missingness processes as distinct as (1) and (3) can produce data that are identical (proved in Chapter 7), we will show that the method needed to overcome missingness depends strongly on the causal story. In other words, causal assumptions are central to missing data analysis.

Table 1.1: Raw data in which Age and Gender are fully observed and Drug Use is partially observed. '?' indicates missing values in Drug Use.

#	Age	Gender	Drug Use
1	13	F	No
2	15	F	?
3	15	M	?
4	14	F	No
5	13	M	No
6	15	M	Yes
7	14	F	Yes

The following section briefly outlines the weaknesses of conventional, non-causal methods of overcoming missingness, and explicates the need to embrace causal assumptions and model-guided

[3] A reader well acquainted with missing data will identify (i) as *missing completely at random* (MCAR). While (1) is typically classified in literature as MCAR, it is not incorrect to call it MAR as MCAR implies MAR. This is further discussed in chapter 2.

analysis.

1.2 Deficiencies in the Conventional Treatment of Missing Data: An Overview

Essentially all the literature on missing data assumes that the data are missing at random ([LR02], page 22). Estimation procedures (such as multiple imputation), software packages (such as MICE in R) and books (such as [Gra12]) were developed, implemented and authored keeping MAR in mind. These developments have engendered a culture with a tendency to blindly assume MAR, with the consequence that the MNAR class of problems remains relatively unexplored. Rarely can we find procedures for handling MNAR data, and this poses a major problem because, in reality, data are more likely to be MNAR [RGP11, Ada07, Osb12, Osb14, Sve15, SK16].

A researcher handling missing data would find MAR appealing for a number of reasons. First, the widespread availability of tools to manage MAR data makes such an assumption convenient. Second, since MAR as defined in [Rub76] is untestable, it might seem tempting to assume that it holds - after all, it is impossible to prove otherwise. Third, assuming that data are MAR spares one the time and effort in determining why data are missing in the first place. Some experts advise not to waste valuable time building missingness models ([SG02], page 171).

Despite its popularity, MAR is rather unintuitive [New14]. It is a misnomer - the missingness process behind even legitimately MAR data is obviously not random (described in scenario (2) in section 1.1). Furthermore, MAR as originally defined in [Rub76] in terms of event-level conditional independence statements is cognitively formidable, making it very hard for a researcher to judge its plausibility in any given problem. We exemplify and elaborate further on this topic in chapter 2.

For all of these reasons, the traditional theory neglects the commonly-occurring MNAR category of missingness, the handling of which requires explicit modeling of the missingness process and developing model-guided procedures to compute queries of interest. This is a major deficiency

Yet another shortcoming of conventional methods is their overreliance on untestable assumptions. Assumptions about the missingness process are the building blocks of missing data theory. Therefore it is imperative that we test them whenever they are testable. Unfortunately, such tests are far and few between [All02]. The price one pays for making invalid assumptions is high, for they can completely distort and bias the outcome of research, making the whole endeavor fruitless.

Using causal assumptions in analysis, developing model-guided estimation procedures and devising tests to detect violation of assumptions are the measures to be taken to overcome deficiencies in this field.

The following section briefly summarizes answers to questions Q-1 to Q-6, and points to chapters that present full answers in the form of algorithms, theorems and lemmata.

1.3 Our Contributions

1.3.1 Missingness in the Language of Causal Graphs

Recent years have witnessed a growing interest in analysing missing data from a causal perspective using graphical models to encode assumptions about the causes of missingness. This development is natural since graphical models provide efficient representation of the independence conditions that are implied by causal assumptions ([Daw79, Lau96, CW96, Pea09]).

We demonstrate that graphical models depicting the data generating process play a critical role in analyzing missing data problems, determining if theoretical impediments exist to eliminating bias due to data missingness, finding procedures that will produce consistent estimates in the absence of impediments, and devising techniques to overcome impediments, if they exist.

1.3.2 Recoverability (Consistent Estimation) of Probabilistic and Causal Queries

Informally, *recoverability* is the ability to compute a consistent estimate of the query from data generated by the model.

We formalize the notion of recoverability and show that relations are always recoverable when data are MAR and, more importantly, that in many commonly occurring cases recoverability can be achieved even when data are MNAR. We further present sufficient graph-based conditions to ensure recoverability of joint, conditional distributions and causal effects in MNAR problems. Finally we identify graph structures that forbid recovery of probabilistic and causal queries. For a broad class of problems we also have necessary and sufficient conditions for recovering probabilistic queries.

The work described above provides answers to questions Q-1 and Q-2.

1.3.3 Overcoming Theoretical Impediments to Recoverability

Consider a dataset consisting of one variable, Income. Assume that both people with high income and people with low income are reluctant to reveal their income. Obviously we do not know if missing values are all high, all low or a combination thereof. This problem poses an impediment to consistent estimation of the average value of income. It is in fact impossible to correctly estimate the average value of income, even when given infinitely many samples.

We present three strategies for overcoming impediments to consistent estimation in problems similar to the example above. The first, based on matrix inversion, can recover joint distributions and is applicable to discrete variables with finite states. The second is applicable to variables governed by linear Gaussian models. Finally, for problems that cannot be handled by either of the preceding strategies, we compute bounds for the target queries.

The work described above answers question Q-3.

1.3.4 Testability under Missingness

It is well known that graphical models provide testable implications which can be detected using d-separation conditions in the graph [Pea09]. For example, a model with two nodes X and Y, with no edges between them, implies the independence of X and Y, which can be tested in the distribution $P(X,Y)$. If the independence claim holds in the data, we can verify that the model and data are compatible. Otherwise, this test can be used to refute the model.

However, when the distribution $P(X,Y)$ is corrupted by missingness we cannot verify an independence claim because the part of the data masked by missing values may contain information that defies the claim. Nevertheless there exist some independence claims that can be refuted by data and they are called *testable*.

We develop syntactic rules for identifying conditional independence claims that are testable. We further present conditions for non-testability of a conditional independence statement and discuss a general impediment to testability in missing data. We show that the popular class of models known as MAR are testable whenever there exist two or more partially observed variables in the dataset. Finally, we demonstrate sensitivity of missing data recovery procedures to structure of hypothesized models and prove that this sensitivity is inevitable in datasets classified as MNAR.

The work described above answers question Q-5.

1.3.5 Application: Robust Algorithms for Closed Form Estimation

We apply our recoverability results [MPT13] to the problem of estimating parameters of a Bayesian network. In particular, we propose a family of efficient and scalable algorithms for learning the parameters of Bayesian networks from MCAR and MAR datasets, and from some MNAR datasets [BMC15]. Our parameter estimates are asymptotically consistent, and further, they are obtained inference-free and in closed-form. Empirically, we demonstrate the practicality of our method, showing that it can scale to much larger datasets, and much larger Bayesian networks, than EM.

The work discussed above answers question Q-4.

1.4 Thesis Roadmap

In chapter 2, we discuss notations and technical preliminaries and formally introduce missingness graphs (m-graphs). Chapters 3 and 4 present algorithms for recovering probabilistic and causal queries. Methods to overcome theoretical impediments to recoverability are discussed in chapters 5 and 6. Testability results are discussed in chapter 7 and estimation algorithms given finite samples are detailed in chapter 8. We draw conclusions and discuss future research directions in the last chapter.

CHAPTER 2

Missingness in the Language of Graphs

In this chapter we describe modeling of missing data generation process using causal Bayesian networks and discuss the relevant notations and terminologies used in this thesis. This is followed by a discussion of graph based categorization of missing data with a focus on the category, 'Missing At Random'. Finally we wrap up with an overview of related work in this area.

2.1 Preliminaries

We use causal graphs to portray the underlying missingness process. Causal graphs are directed acyclic graphs where vertices correspond to variables and edges represent not just dependencies but also functional relationship between the variables they connect. Independencies embedded in a graphical model are read off it using the d-separation criterion described below.

Definition 1 (d-separation [Pea09])**.** *A path p is said to be d-separated by a set of nodes Z if and only if:*
(1) p contains a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle node m is in Z, or
(2) p contains an inverted fork (or collider) $i \rightarrow m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.

A set Z is said to d-separate X from Y if and only if Z blocks every path from a node in X to a node in Y and is denoted by $Y \perp\!\!\!\perp X|Z$.

Remark 1 (Minimal Separator)**.** *Given two sets of nodes X and Y in DAG and a set Z that d-separates X from Y, Z is a minimal separator if no proper subset of Z d-separates X from Y. There are polynomial time algorithms to find minimal separators [AD96, TPP98].*

2.2 Graphical Models for Missing Data: Missingness Graphs (m-graphs)

The following example, inspired by [LR02] (example-1.6, page 8), describes how graphical models can be used to explicitly model the missingness process and encode the underlying causal and statistical assumptions. Consider a study conducted in a school that measured three (discrete) variables: Age (A), Gender (G) and Obesity (O).

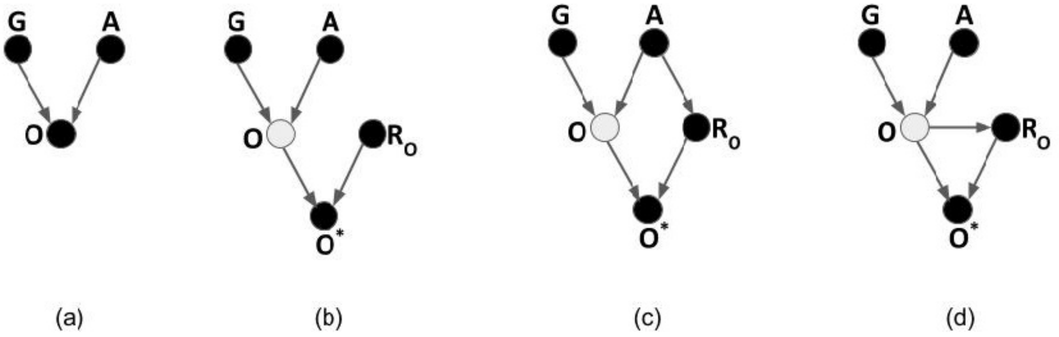


Figure 2.1: (a) Causal graph under no missingness (b), (c) & (d) m-graphs modeling distinct missingness processes.

No Missingness If all three variables are completely recorded, then there is no missingness. The causal graph[1] depicting the interrelations between variables is shown in Figure 2.1 (a). Nodes correspond to variables and edges indicate the existence of a relationship between pairs of nodes they connect. The value of a child node is a function of the values of its parent nodes. i.e. Obesity is a function of Age and Gender. The absence of an edge between Age and Gender indicates that A and G are independent, denoted by $A \perp\!\!\!\perp G$.

Representing Missingness Assume that Age and Gender are are fully observed since they can be obtained from school records. Obesity however is corrupted by missing values due to some students not revealing their weight. When the value of O is missing we get an empty measurement which we designate by m. Table 2.1 exemplifies a missing dataset. The missingness process can be modelled using a proxy variable Obesity$^*(O^*)$ whose values are determined by Obesity and its

[1] For a quick introduction to causal graphical models refer section 1.2 in [Pea09]

Table 2.1: Missing dataset in which Age and Gender are fully observed and Obesity is partially observed.

#	Age	Gender	Obesity*	R_O
1	11	F	Obese	R_O
2	15	F	m	1
3	15	M	m	1
4	14	F	Not Obese	0
5	13	M	Not Obese	0
6	15	M	Obese	0
7	14	F	Obese	0

missingness mechanism R_O.

$$O^* = f(R_O, O) = \begin{cases} O & \text{if } R_O = 0 \\ m & \text{if } R_O = 1 \end{cases}$$

R_o governs the masking and unmasking of Obesity. When $R_o = 1$ the value of obesity is concealed i.e. O^* assumes the values m as shown in samples 2 and 3 in table 2.1. When $R_o = 0$, the true value of obesity is revealed i.e. O^* assumes the underlying value of Obesity as shown in samples 1, 4, 5, 6 and 7 in table 2.1.

Choosing the correct estimation procedure is paramount to the outcome of any study involving missing data. Two identical datasets may require disparate estimation strategies which in turn are determinable only from the causes of missingness. Missingness can be caused by random processes or can depend on other variables in the dataset. An example of random missingness is students *forgetting* to return their questionnaires. This is depicted in figure 2.1 (b) by the absence of parent nodes for R_o. Teenagers rebelling and not reporting their weight is an example of missingness caused by a fully observed variable. This is depicted in figure 2.1 (c) by an edge between A and R_o. Partially observed variables can be causes of missingness as well. For instance consider obese students who are embarrassed of their obesity and hence reluctant to reveal their weight.

This is depicted in figure 2.1 (d) by an edge between O and R_o indicating the O is the cause of its own missingness.

The following subsection formally introduces missingness graphs (m-graphs) as discussed in [MPT13].

2.2.1 Missingness Graphs: Notations and Terminology

Let $G(\mathbf{V}, E)$ be the causal DAG where $\mathbf{V} = V_o \cup V_m \cup U \cup V^* \cup \mathbf{R}$. Nodes in the graph correspond to variables in the data set. U is the set of unobserved nodes (also called latent variables). E is the set of edges in the DAG. We use bi-directed edges as a shorthand notation to denote the existence of a U variable as common parent of two variables in $V_o \cup V_m \cup \mathbf{R}$. V_o is the set of variables that are observed in all records in the population and V_m is the set of variables that are missing in at least one record. Variable X is termed as *fully observed* if $X \in V_o$ and *partially observed* if $X \in V_m$. R_{v_i} and V_i^* are two variables associated with every partially observed variable, where V_i^* is a proxy variable that is actually observed, and R_{v_i} represents the status of the causal mechanism responsible for the missingness of V_i^*; formally,

$$v_i^* = f(r_{v_i}, v_i) = \begin{cases} v_i & \text{if } r_{v_i} = 0 \\ m & \text{if } r_{v_i} = 1 \end{cases} \tag{2.1}$$

V^* is the set of all proxy variables and $\mathbf{R}$ is the set of all causal mechanisms that are responsible for missingness. Unless stated otherwise it is assumed that no variable in $V \cup U$ is a child of an R variable. We call this graphical representation **Missingness Graph** (or m-graph). Figure 2.1 exemplifies three m-graphs in which $V_o = \{A, G\}$, $V_m = \{O\}$, $V^* = \{O^*\}$, $U = \emptyset$ and $R = \{R_o\}$. Proxy variables may not always be explicitly shown in m-graphs in order to keep the figures simple and clear. The missing data distribution, $P(V^*, V_o, R)$ is referred to as the *manifest distribution* and the distribution that we would have obtained had there been no missingness, $P(V_o, V_m, R)$ is called as the *underlying distribution*. Testing the compatibility of a manifest distribution with an underlying distribution is discussed in appendix A.1.1. Conditional Independencies are read off the graph using the d-separation[2] criterion ([Pea09]).

[2] For a quick introduction to d-separation see, http://www.dagitty.net/learn/dsep/index.html

2.3 Classification of Missing Data Problems based on Missingness Mechanism

[Rub76] classified missing data into three categories: Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR) based on the statistical dependencies between the missingness mechanisms (R variables) and the variables in the dataset (V_m, V_o). We capture the essence of this categorization in graphical terms below.

1. Data are MCAR if $V_m \cup V_o \perp\!\!\!\perp R$ holds in the m-graph. In words, missingness occurs at random and is entirely independent of both the observed and the partially observed variables. This condition can be easily identified in an m-graph by the absence of edges between the R variables and variables in $V_o \cup V_m$.

2. Data are MAR if $V_m \perp\!\!\!\perp R|V_o$ holds in the m-graph. In words, conditional on the fully observed variables V_o, missingness occurs at random. In graphical terms, MAR holds if (i) no edges exist between an R variable and any partially observed variable and (ii) no bidirected edge exists between an R variable and a fully observed variable. MCAR implies MAR, ergo all estimation techniques applicable to MAR can be safely applied to MCAR.

3. Data that are not MAR or MCAR fall under the MNAR category.

m-graphs in figure 2.1 (b), (c) and (d) are typical examples of MCAR, MAR and MNAR categories, respectively. Notice the ease with which the three categories can be identified. Once the user lays out the interrelationships between the variables in the problem, the classification is purely mechanical.

2.4 Missing At Random: A Brief Discussion

The original classification used in [Rub76] is very similar to the one defined in the preceding paragraphs; it is expressed however in terms of event-level conditional independencies as opposed to variable-level independencies. We will clarify the distinction between the former (which we call *Rubin-MAR*) and the latter (referred to as MAR) with an example. Consider a dataset with three variables such that two variables, A and B are partially observed and the third one C is fully observed. For data to be MAR, we require $(A, B) \perp\!\!\!\perp (R_A, R_B)|C$ to hold. On the other hand, Rubin-MAR requires that, " *missingness depends only on the components Y_{obs} of Y that are observed and not on the components that are missing*"([LR02]), where Y denotes the dataset. We exemplify Rubin-MAR in table 2.2. The primary difference between the two definitions is that MAR is a succinct statement comprising of a single conditional independence: $V_m \perp\!\!\!\perp R|V_o$, where as Rubin-MAR is a set of distinct conditional independencies of the form: $Y_{mis} \perp\!\!\!\perp R|Y_{obs}$, one for each subpopulation as described by the pattern of missingness[3]. Observe that both definitions coincide when $|V_m| = 1$.

Table 2.2: Rubin-MAR detailed for the dataset in which A and B are partially observed variables and C is a fully observed variable.

Missing Components Y_{mis}	Observed Components Y_{obs}	Rubin-MAR Conditions $Y_{mis} \perp\!\!\!\perp R|Y_{obs}$	Description of Samples
A	B, C	$A \perp\!\!\!\perp R|B, C$	Samples in which A is missing and B is observed
B	A, C	$B \perp\!\!\!\perp R|A, C$	Samples in which B is missing and A is observed
A, B	C	$(A, B) \perp\!\!\!\perp R|C$	Samples in which both A and B are missing
$-$	A, B, C	$-$	Samples in which all variables are observed.

Over the years the classification proposed in [Rub76] has been criticized both for its nomen-

[3] Each instantiation of R variables corresponds to a pattern of missingness. In the case of the ongoing example with $V_m = \{A, B\}$ and $V_o = \{C\}$, there are 4 patterns of missingness: $(R_A = 0, R_B = 0), (R_A = 0, R_B = 1), (R_A = 1, R_B = 0)$ and $(R_A = 1, R_B = 1)$ ([DMG08, CW15]).

clature and its opacity. Several authors noted that **MAR is a misnomer** ([Sch02, PE02, MGG06, Gra09]). What is currently defined as MCAR should have been called Missing At Random and as pointed out by Grace-Martin[4], what is currently defined as Missing At Random should have been called Missing Conditionally At Random.

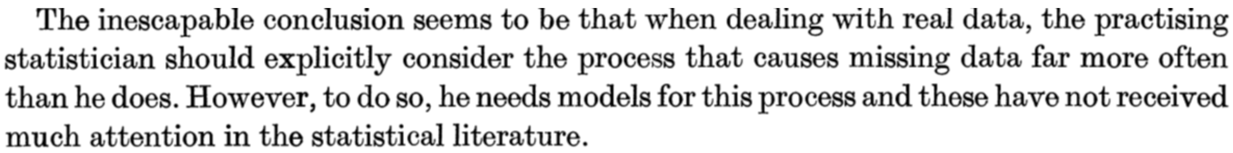
However, the **opacity of the assumptions** embedded in Rubin MAR presents a more serious problem. The number of conditional independence relations that need be verified is exponential in the number of partially observed variables. This is shown in Table 2.2, which displays the conditional independencies claimed by *Rubin-MAR* condition: $Y_{mis} \perp\!\!\!\perp R|Y_{obs}$. Clearly, a researcher would find it cognitively taxing, if not impossible to even decide if these assumptions are reasonable. This, together with the fact that Rubin-MAR is untestable ([All02]) motivates the variable-based taxonomy presented above.

Nonetheless, Rubin-MAR has an interesting theoretical property: It is the weakest simple condition under which the process that causes missingness can be ignored while still making correct inferences about the data ([Rub76]). It was probably this theoretical result that changed missing data practices in the 1970s. The popular practice prior to 1976 was to assume that missingness was caused totally at random ([GS75, Hai68]). With Rubin's identification of the MAR condition as sufficient for drawing correct inferences, MAR became the main focus of attention in the statistical literature.

Estimation procedures such as Multiple Imputation and Maximum Likelihood were developed and implemented with MAR assumptions in mind, and popular textbooks were authored exclusively on MAR ([Gra12]). These developments have engendered a culture with a tendency to blindly assume MAR, with the consequence that the more commonly occurring MNAR class of problems remains relatively unexplored ([RGP11, Ada07, Osb12, Osb14, Sve15, SK16])).

Need for Model Guided Estimation Procedures To overcome these limitations one must explicitly model the missingness process and [Rub76] made similar recommendations. In his words,

[4] http://www.theanalysisfactor.com/mar-and-mcar-missing-data/



> The inescapable conclusion seems to be that when dealing with real data, the practising statistician should explicitly consider the process that causes missing data far more often than he does. However, to do so, he needs models for this process and these have not received much attention in the statistical literature.

The graphical tools described in this paper provide a flexible way of modelling the missingness process and thus overcome the limitations identified in blindly assuming MAR. These tools enable researchers to both encode assumptions about the type of missingness that may occur in their data, and to extend the analysis of estimation techniques to the vast class of MNAR problems.

2.5 Related Work

For detailed discussion of missing data theory and practice we direct readers to the books ([All02, End10, LR02, MMS07]). Among all methods used for handling missing data, listwise deletion and pairwise deletion are the easiest to implement and have been found to be popular among practitioners ([PE04]) even though estimates produced by these methods are guaranteed to converge only under MCAR.

Listwise deletion or (complete case analysis) refers to the simple technique in which samples with missing values are deleted ([BGN08]). Unless data are missing completely at random, listwise deletion can bias the outcome ([Wot00]). Evidently this technique results in wastage of data.

Pairwise deletion (or available case analysis) is a deletion method that drastically reduces data loss by operating on all samples in which the variables of interest are observed [SBC10]. For example, to compute the covariance of variables X and Y, all samples in which both X and Y are observed are used, regardless of whether other variables in the dataset have missing values.

Another approach to handling missing data is imputation: substituting a reasonable guess for each missing value [All02]. A simple example is *mean Substitution*, in which all missing observations of variable X are substituted with the mean of all observed values of X. Hot-deck imputation, cold-deck imputation [MMS07], regression imputation [Sch02] and Multiple Imputation [Rub87, Rub96] are examples of popular imputation procedures. Among these techniques, regression imputation guarantees consistent estimates for MAR data [PE04]. While many other imputation techniques are attractive in practice, performance guarantees (eg: convergence and unbiasedness) are based primarily on simulation experiments.

Whenever data are Missing At Random, Maximum Likelihood (ML) based methods can be used for computing consistent estimates of parameters of interest [LR02]. Recent increase in the popularity of ML based procedures can be attributed to its quick and easy availability in the form of software packages. The expectation-maximization (EM) algorithm [DLR77] is a general technique for finding maximum likelihood (ML) estimates from MAR data.

Weighting procedures for missing data are based on creating weighted copies of complete cases and are succinctly summarized in [LSL13]. These procedures that are primarily based on [HT52] and have been generalized to address missing data problems in [RRZ94], [RRZ95] and [RHB00].

The handling of MNAR data is more or less limited to performing sensitivity analysis [RGP11]. Methods of performing sensitivity analysis have been suggested in research publications such as [RRS98, MKG01] and [TMM02]. Special handling of MNAR problems based on use of selection models [Hec77] and pattern mixture models is discussed in [End11].

The use of graphical models for handling missing data is a relatively new development. [DKC12] discussed sufficient criteria under which consistent estimates can be computed exclusively from complete cases (i.e. samples in which all variables are fully observed). [TR13] (and later on [TM15]) developed techniques that guide the selection of auxiliary variables to improve estimability from incomplete data. In machine learning, particularly while estimating parameters of Bayesian Networks, graphical models have long been used primarily as a pedagogical tool when dealing with missing data ([Dar09, KF09]).

Missing data discussed so far is a special case of *coarse data*, namely data that contains observations made in the power set rather than the sample space of variables of interest [HR91]. The notion of coarsening at random (CAR) was introduced in [HR91] and identifies the condition under which coarsening mechanism can be ignored while drawing inferences on the distribution of variables of interest [GVR97]. The notion of sequential CAR has been discussed in [GR97]. For a detailed discussion on coarsened data refer to [LR03].

CHAPTER 3

Recoverability of Probabilistic and Interventional Distribution

Recoverability refers to the task of determining from an assumed model whether any method exists that produces a consistent estimate of a target quantity for all data generated by the model, and if so, how. If the answer is negative, then an inconsistent estimate should be expected even with large samples. On the other hand, if the answer is affirmative then there exists a procedure that can exploit the features of the problem and produces consistent estimates.

If the problem is MAR or MCAR, joint distribution and hence all probabilistic relations and all identifiable causal effects are recoverable. But if a problem is MNAR, some queries of interest cannot be estimated by any method whatsoever while others can. We will show that MNAR problems exhibit this dichotomy, and more importantly that estimable parameters can often be identified directly from the structure of the graph. In this chapter we present several methods of deriving consistent estimators for both statistical and causal parameters.

Chapter Outline: In section 3.1 we define and exemplify recoverability and present conditions for recoverability when data are MCAR and MAR. In sections 3.2 and 3.4 we present graphical conditions for recoverability of MNAR problems. Non-recoverability is discussed and formalized in section 3.3. Recoverability of causal effects is detailed in 3.5.

3.1 Recoverability

Recoverability addresses the question of whether a quantity/parameter of interest can be estimated from incomplete data *as if* the data were complete.

Definition 2 (Recoverability of target quantity Q)**.** *Given m-graph G and the manifest distribution P^*, Q is recoverable if there exists an algorithm that can compute a consistent estimate of Q for* *all strictly positive data distributions $P(V^*, V_o, R)$ that G can generate.*

Typical target quantities Q that shall be considered are conditional/joint distributions and conditional causal effects. Note that for a given target Q, recoverability is a property of the m-graph G, and not of the data. The reason for restricting the definition above to strictly positive manifest distributions, $P(V^*, V_o, R)$, is mainly technical, to avoid division by zero. We allow however instances of zero probabilities as specified in equation 2.1. We note that recoverability is sometimes feasible even when strict positivity does not hold. We exemplify such an instance in appendix A.2.1.

Corollary 1. *A relation Q is recoverable in G if Q can be expressed in terms of the probability $P(O)$ where $O = \{R, V^*, V_o\}$ is the set of observable variables in G. In other words, for any two models M_1 and M_2 inducing distributions P^{M_1} and P^{M_2} respectively, if $P^{M_1}(O) = P^{M_2}(O) > 0$ then $Q^{M_1} = Q^{M_2}$.*

Proof: (sketch) The corollary merely rephrases the requirement of obtaining a consistent estimate to that of expressibility in terms of observables.

Practically, what recoverability means is that if the data D are generated by any process compatible with G, a procedure exists that computes an estimator $\hat{Q}(D)$ such that, in the limit of large samples, $\hat{Q}(D)$ converges to Q. Such a procedure is called a "consistent estimator." Thus, recoverability is the sole property of G and Q, not of the data available, or of any routine chosen to analyze or process the data.

3.1.1 Recoverability when data are MAR

When data are MAR, we have $R \perp\!\!\!\perp V_m | V_o$. Therefore $P(V) = P(V_m|V_o)P(V_o) = P(V_m|V_o, R = 0)P(V_o)$. Hence the joint distribution $P(V)$ is recoverable.

Example 2. *Consider the problem of recovering the joint distribution given the m-graph in Fig. 2.1 (c) and dataset in table 3.1. Let it be the case that 15-18 year olds were reluctant to reveal their weight, thereby making O a partially observed variable i.e. $V_m = \{O\}$ and $V_o = \{G, A\}$. This is a typical case of MAR missingness, since the cause of missingness is the fully observed variable:*

Age. The following three steps detail the recovery procedure.

1. Factorization: The joint distribution may be factorized as:

$$P(G, O, A) = P(G, O|A)P(A)$$

2. Transformation into observables: G embeds the conditional independence $(G, O) \perp\!\!\!\perp R_o|A$ which is read using d-separation criterion. Thus,

$$P(G, O, A) = P(G, O|A, R_o = 0)P(A)$$

3. Conversion of partially observed variables into proxy variables: When $R_o = 0$, $O^* = O$ (by eq 2.1). Hence,

$$P(G, O, A) = P(G, O^*|A, R_o = 0)P(A) \tag{3.1}$$

The RHS of equation 3.1 is in terms of variables in the manifest distribution. Therefore, $P(G, A, O)$ can be consistently estimated (i.e. recovered) from the available data. The recovered joint distribution is shown in table 3.2.

Remark 2. *Notice that in equation 3.1, factors are estimated from different subsets of data. For instance, the factor $P(G, O^*|A, R_o = 0)$ is computed exclusively from samples in which O is observed where as the factor, $P(A)$, is computed from all samples, regardless of the missingness status of O.*

Remark 3. *Furthermore, in the preceding example $P(G, O, A)$ was factorized as $P(G, O|A)P(A)$ as opposed to other options such as, $P(A|G, O)P(G, O)$. The former facilitates recovery, while the latter does not. The initial order of factorization thus plays a pivotal role in the recovery procedure.*

3.1.2 Recoverability when data are MCAR

When data are MCAR we have $R \perp\!\!\!\perp (V_o \cup V_m)$. Therefore, we can write $P(V) = P(V|R) = P(V_o, V^*|R = 0)$. Since both R and V^* are observables, the joint probability $P(V)$ is consistently

Table 3.1: Manifest Distribution $P(GAO^*R_o)$ where Gender (G) and Age (A) are fully observed and Obesity's proxy (O^*) is observed in its place. Age is partitioned into three groups: $[10-13), [13-15), [15-18)$. Gender and Obesity are binary variables and can take values Male (M) and Female (F), and Yes (Y) and No (N), respectively.

G	A	O^*	R_O	$P(G, A, O^*, R_O)$
M	$10-13$	Y	0	p_1
M	$13-15$	Y	0	p_2
M	$15-18$	Y	0	p_3
M	$10-13$	N	0	p_4
M	$13-15$	N	0	p_5
M	$15-18$	N	0	p_6
F	$10-13$	Y	0	p_7
F	$13-15$	Y	0	p_8
F	$15-18$	Y	0	p_9
F	$10-13$	N	0	p_{10}
F	$13-15$	N	0	p_{11}
F	$15-18$	N	0	p_{12}
M	$10-13$	m	1	p_{13}
M	$13-15$	m	1	p_{14}
M	$15-18$	m	1	p_{15}
F	$10-13$	m	1	p_{16}
F	$13-15$	m	1	p_{17}
F	$15-18$	m	1	p_{18}

estimable (hence recoverable) by considering complete cases only (listwise deletion), as shown in the following example.

Example 3. *Let X be the treatment and Y be the outcome as depicted in the m-graph in Fig. 3.1 (a). Let it be the case that we accidentally deleted the values of Y for a handful of samples, hence $Y \in V_m$. Can we recover $P(X, Y)$?*

From D, we can compute $P(X, Y^*, R_y)$. From the m-graph G, we know that Y^* is a collider and hence by d-separation, $(X \cup Y) \perp\!\!\!\perp R_y$. Thus $P(X, Y) = P(X, Y|R_y)$. In particular, $P(X, Y) = P(X, Y|R_y = 0)$. When $R_y = 0$, by eq. (2.1), $Y^* = Y$. Hence,

$$P(X, Y) = P(X, Y^*|R_y = 0) \tag{3.2}$$

The RHS of Eq. 3.2 is consistently estimable from D; hence $P(X, Y)$ is recoverable.

The assumption of MCAR allows an estimation procedure that amounts (asymptotically) to list-

Table 3.2: Recovered joint distribution corresponding to dataset in table 3.1 and m-graph in figure 2.1(c)

G	A	O	$P(G,A,O)$
M	$10-13$	Y	$\frac{p_1*(p_1+p_4+p_7+p_{10}+p_{13}+p_{16})}{p_1+p_4+p_7+p_{10}}$
M	$13-15$	Y	$\frac{p_2*(p_2+p_5+p_8+p_{11}+p_{14}+p_{17})}{p_2+p_5+p_8+p_{11}}$
M	$15-18$	Y	$\frac{p_3*(p_3+p_6+p_9+p_{12}+p_{15}+p_{18})}{p_3+p_6+p_9+p_{12}}$
M	$10-13$	N	$\frac{p_4*(p_1+p_4+p_7+p_{10}+p_{13}+p_{16})}{p_1+p_4+p_7+p_{10}}$
M	$13-15$	N	$\frac{p_5*(p_2+p_5+p_8+p_{11}+p_{14}+p_{17})}{p_2+p_5+p_8+p_{11}}$
M	$15-18$	N	$\frac{p_6*(p_3+p_6+p_9+p_{12}+p_{15}+p_{18})}{p_3+p_6+p_9+p_{12}}$

G	A	O	$P(G,A,O)$
F	$10-13$	Y	$\frac{p_7*(p_1+p_4+p_7+p_{10}+p_{13}+p_{16})}{p_1+p_4+p_7+p_{10}}$
F	$13-15$	Y	$\frac{p_8*(p_2+p_5+p_8+p_{11}+p_{14}+p_{17})}{p_2+p_5+p_8+p_{11}}$
F	$15-18$	Y	$\frac{p_9*(p_3+p_6+p_9+p_{12}+p_{15}+p_{18})}{p_3+p_6+p_9+p_{12}}$
F	$10-13$	N	$\frac{p_{10}*(p_1+p_4+p_7+p_{10}+p_{13}+p_{16})}{p_1+p_4+p_7+p_{10}}$
F	$13-15$	N	$\frac{p_{11}*(p_2+p_5+p_8+p_{11}+p_{14}+p_{17})}{p_2+p_5+p_8+p_{11}}$
F	$15-18$	N	$\frac{p_{12}*(p_3+p_6+p_9+p_{12}+p_{15}+p_{18})}{p_3+p_6+p_9+p_{12}}$

wise deletion, while MAR dictates a procedure that amounts to listwise deletion in every stratum of V_o. Applying MAR procedure to MCAR problem is safe, because all conditional independencies required for recoverability under the MAR assumption also hold in an MCAR problem, i.e. $R \perp\!\!\!\perp (V_o, V_m) \Rightarrow R \perp\!\!\!\perp V_m | V_o$. The converse, however, does not hold, as can be seen in Fig. 3.1 (b). Applying listwise deletion is likely to result in bias, because the necessary condition $R \perp\!\!\!\perp (V_o, V_m)$ is violated in the graph.

3.1.3 Recoverability when data are MNAR

Data that are neither MAR nor MCAR are termed MNAR.

Example 4. *Fig. 3.1 (d) depicts a study where (i) some units who underwent treatment ($X = 1$) did not report the outcome (Y) and (ii) we accidentally deleted the values of treatment for a handful of cases. Thus we have missing values for both X and Y which renders the dataset MNAR. We shall show that $P(X,Y)$ is recoverable. From D, we can compute $P(X^*, Y^*, R_x, R_y)$. From the m-graph G, we see that $X \perp\!\!\!\perp R_x$ and $Y \perp\!\!\!\perp (R_x \cup R_y) | X$. Thus $P(X,Y) = P(Y|X)P(X) = P(Y|X, R_y = 0, R_x = 0)P(X|R_x = 0)$. When $R_y = 0$ and $R_x = 0$ we have (by Equation (2.1)), $Y^* = Y$ and $X^* = X$. Hence,*

$$P(X,Y) = P(Y^*|X^*, R_x = 0, R_y = 0)P(X^*|R_x = 0) \tag{3.3}$$

Therefore, $P(X, Y)$ is recoverable.

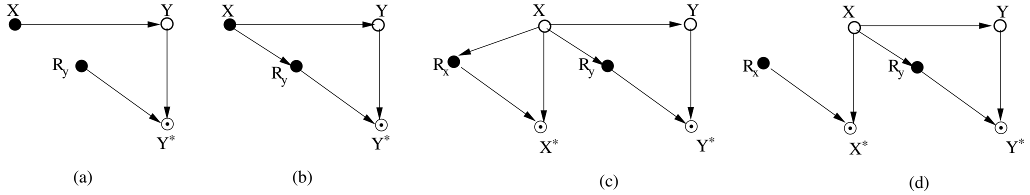


Figure 3.1: m-graphs that depict: (a) MCAR, (b) MAR, (c) & (d) MNAR.

An interesting property which evolves from this discussion is that recoverability of certain relations does not require $R_{V_i} \perp\!\!\!\perp V_i | V_o$; a subset of V_o would suffice as shown below.

Property 1. *$P(V_i)$ is recoverable if $\exists W \subseteq V_o$ such that $R_{V_i} \perp\!\!\!\perp V | W$.*

Proof: $P(V_i)$ may be decomposed as: $P(V_i) = \sum_w P(V_i^* | R_{v_i} = 0, W) P(W)$ since $V_i \perp\!\!\!\perp R_{V_i} | W$ and $W \subseteq V_o$. Hence $P(V_i)$ is recoverable.

Our next question is: how can we determine if a given relation is recoverable? The following theorem provides a sufficient condition for recoverability.

3.1.4 Conditions for Recoverability

Theorem 1. *A query Q defined over variables in $V_o \cup V_m$ is recoverable if it is decomposable into terms of the form $Q_j = P(S_j | T_j)$ such that T_j contains the missingness mechanism $R_v = 0$ of every partially observed variable V that appears in Q_j.*

Proof: If such a decomposition exists, every Q_j is estimable from the data, hence the entire expression for Q is recoverable.

Example 5. *Equation (3.3) demonstrates a decomposition of $Q = P(X, Y)$ into a product of two terms $Q_1 = P(Y | X, R_x = 0, R_y = 0)$ and $Q_2 = P(X | R_x = 0)$ that satisfy the condition of Theorem 1. Hence Q is recoverable.*

In the following section we define the notion of Ordered factorization which leads to a criterion for sequentially recovering conditional probability distributions.

3.2 Recovery by Sequential Factorization

Definition 3 (Ordered factorization of $P(Y|Z)$)**.** *Let $Y_1 < Y_2 < \ldots < Y_k$ be an ordered set of all variables in Y, $1 \leq i \leq |Y| = k$ and $X_i \subseteq \{Y_{i+1}, \ldots, Y_n\} \cup Z$. Ordered factorization of $P(Y|Z)$ is the product of conditional probabilities i.e. $P(Y|Z) = \prod_i P(Y_i|X_i)$, such that X_i is a minimal set for which $Y_i \perp\!\!\!\perp (\{Y_{i+1}, \ldots, Y_n\} \setminus X_i)|X_i$ holds.*

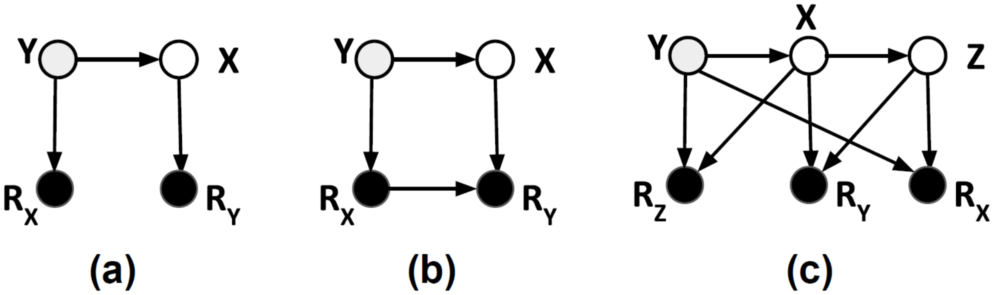


Figure 3.2: (a) & (c) m-graphs in which joint distribution is recoverable. (b) m-graph in which conditional distribution $P(X|Y)$ is recoverable.

The following theorem presents a sufficient condition for recovering conditional distributions of the form $P(Y|X)$ where $\{Y, X\} \subseteq V_m \cup V_o$.

Theorem 2. *Given an m-graph G and a manifest distribution $P(V^*, V_o, R)$, a target quantity Q is recoverable if Q can be decomposed into an ordered factorization, or a sum of such factorizations, such that every factor $Q_i = P(Y_i|X_i)$ satisfies $Y_i \perp\!\!\!\perp (R_{y_i}, R_{x_i})|X_i$. Then, each Q_i may be recovered as $P(Y_i^*|X_i^*, R_{Y_i} = 0, R_{X_i} = 0)$.*

Proof of the preceding theorem is presented in appendix A.2.3.

An ordered factorization that satisfies theorem 2 is called as an *admissible factorization*. Heuristics for finding admissible factorizations is discussed in appendix A.2.2.

Example 6. *Let G be the m-graph obtained by removing the edge between X and R_y in figure 3.2 (a). G depicts an MNAR problem since missingness in X is caused by the partially observed variable Y. Let the query of interest be $P(X, Y)$. The factorization $P(X|Y)P(Y)$ is admissible* *since both $X \perp\!\!\!\perp R_x, R_y|Y$ and $Y \perp\!\!\!\perp R_y$ hold in G. $P(X,Y)$ can thus be recovered using theorem 2 as $P(X^*|Y^*, R_x = 0, R_y = 0)P(Y^*|R_y = 0)$.*

The following theorem gives a sufficient condition for recovering the joint distribution in a Markovian model, without resorting to the use of an admissible factorization.

Theorem 3. *Given a m-graph with no latent variables (i.e., Markovian) the joint distribution $P(V)$ is recoverable if no missingness mechanism R_X is a descendant of its corresponding variable X. Moreover, if recoverable, then $P(V)$ is given by*

$$P(v) = \prod_{i,V_i \in V_o} P(v_i|pa_i^o, pa_i^m, R_{Pa_i^m} = 0) \prod_{j,V_j \in V_m} P(v_j|pa_j^o, pa_j^m, R_{V_j} = 0, R_{Pa_j^m} = 0), \tag{3.4}$$

where $Pa_i^o \subseteq V_o$ and $Pa_i^m \subseteq V_m$ are the parents of V_i.

Proof: Refer Appendix-A.2.4

3.3 Non-recoverability Criteria for Joint and Conditional Distributions

Up until now, we dealt with sufficient conditions for recoverability. It is important however to supplement these results with criteria for non-recoverability in order to alert the user to the fact that the available assumptions are insufficient to produce a consistent estimate of the target query. Such criteria have not been treated formally in the literature thus far. In the following theorem we introduce two graphical conditions that preclude recoverability.

Theorem 4 (Non-recoverability of $P(V)$)**.** *Given a semi-Markovian model G, the following conditions are necessary for recoverability of the joint distribution:*
(i) $\forall X \in V_m$, X and R_x are not neighbors and
(ii) $\forall X \in V_m$, there does not exist a path from X to R_x in which every intermediate node is both a collider and a substantive variable.

Proof Refer appendix A.2.5

While models satisfying (i) in theorem 4 are called **self-masking**, those satisfying (ii) are called **collider-induced**.

In the following corollary, we leverage theorem 4 to yield necessary conditions for recovering conditional distributions.

Corollary 2. *[Non-recoverability of $P(Y|X)$] Let X and Y be disjoint subsets of substantive variables. $P(Y|X)$ is non-recoverable in m-graph G if one of the following conditions is true:*
(1) Y and R_y are neighbors
(2) G contains a collider path p connecting Y and R_y such that all intermediate nodes in p are in X.

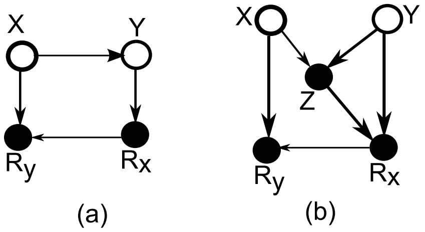


Figure 3.3: m-graphs depicting entangled models in which joint distribution is non-recoverable.

Figure 3.3 exemplifies other models, called **entangled**, in which joint distribution is non-recoverable. Proofs pertaining to non-recoverability of these models are presented in appendix A.2.6.

3.4 Recovery by R Factorization

Example 7. *Consider the problem of recovering $Q = P(X,Y)$ from the m-graph of Fig. 3.2 (a). Attempts to decompose Q by the chain rule, as was done in Eqs. (3.1) and (3.3) would not satisfy the conditions of Theorem 2. To witness we write $P(X,Y) = P(Y|X)P(X)$ and note that the graph does not permit us to augment any of the two terms with the necessary R_x or R_y terms; X is independent of R_x only if we condition on Y, which is partially observed, and Y is independent of R_y only if we condition on X which is also partially observed. This deadlock can be disentangled however using a non-conventional decomposition:*

$$\begin{aligned} Q = P(X,Y) &= P(X,Y)\frac{P(R_x,R_y|X,Y)}{P(R_x,R_y|X,Y)} \\ &= \frac{P(R_x,R_y)P(X,Y|R_x,R_y)}{P(R_x|Y,R_y)P(R_y|X,R_x)} \end{aligned} \tag{3.5}$$

where the denominator was obtained using the independencies $R_x \perp\!\!\!\perp (X, R_y)|Y$ and $R_y \perp\!\!\!\perp (Y, R_x)|X$ shown in the graph. The final expression above satisfies Theorem 2 and renders $P(X, Y)$ recoverable. This example again shows that recovery is feasible even when data are MNAR.

The following theorem formalizes the recoverability scheme exemplified above.

Theorem 5 (Recoverability of the Joint $P(V)$)**.** *Given a m-graph G with no edges between R variables the necessary and sufficient condition for recovering the joint distribution $P(V)$ is the absence of any variable $X \in V_m$ such that:*
1. X and R_x are neighbors
2. X and R_x are connected by a path in which all intermediate nodes are colliders and elements of $V_m \cup V_o$. When recoverable, $P(V)$ is given by

$$P(v) = \frac{P(R = 0, v)}{\prod_i P(R_i = 0|Mb^o_{r_i}, Mb^m_{r_i}, R_{Mb^m_{r_i}} = 0)}, \tag{3.6}$$

where $Mb^o_{r_i} \subseteq V_o$ and $Mb^m_{r_i} \subseteq V_m$ are the Markov blanket of R_i.

Proof: Refer appendix A.2.7

The preceding theorem can be applied to immediately yield an estimand for joint distribution. For instance, given the m-graphs in figure 3.2 (c), joint distribution can be recovered in one shot as:

$$P(X, Y, Z) = \frac{P(X,Y,Z,R_x=0,R_y=0,R_z=0)}{P(R_x=0|Y,R_y=0,Z,R_z=0)P(R_y=0|X,R_x=0,Z,R_z=0)P(R_z=0|Y,R_y=0,X,R_x=0)}$$

3.5 Recovering Causal Queries

Given a causal query and a causal Bayesian network a complete algorithm exists for deciding whether the query is identifiable or not ([SP06]). Obviously, a query that is not identifiable in the substantive model is not recoverable from missing data. Therefore, a necessary condition for recoverability of a causal query is its identifiability which we will assume in the rest of our discussion.

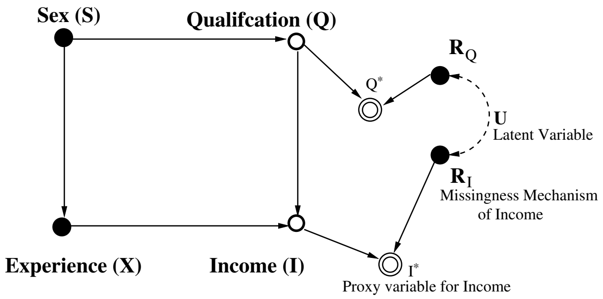


Figure 3.4: m-graph associated with example 8 where $V_o = \{S, X\}$, $V_m = \{I, Q\}$, $V^* = \{I^*, Q^*\}$, $R = \{R_i, R_q\}$ and U is the latent common cause.

Definition 4 (Trivially Recoverable Query). *A causal query Q is said to be trivially recoverable given an m-graph G if it has an estimand (in terms of substantive variables) in which every factor is recoverable.*

Classes of problems that fall into the MCAR (Missing Completely At Random) and MAR (Missing At Random) category are much discussed in the literature (([Rub76])) because in such categories probabilistic queries are recoverable by graph-blind algorithms. An immediate but important implication of trivial recoverability is that if data are MAR or MCAR and the query is identifiable, then it is also recoverable by model-blind algorithms.

Example 8. *In the gender wage-gap study example in Figure 3.4, the effect of sex on income, $P(I|do(S))$, is identifiable and is given by $P(I|S)$. By theorem 2, $P(S, X, Q, I)$ is recoverable. Hence $P(I|do(S))$ is recoverable.*

3.5.1 Recovering $P(y|do(z))$ when Y and R_y are inseparable

The recoverability of $P(V)$ hinges on the separability of a partially observed variable from its missingness mechanism (a condition established in theorem 4). Remarkably, causal queries may circumvent this requirement. The following example demonstrates that $P(y|do(z))$ is recoverable even when Y and R_y are not separable.

Example 9. *Examine Figure 3.5. By backdoor criterion, $P(y|do(z)) = \sum_w P(y|z, w)P(w)$. One might be tempted to conclude that the causal relation is non-recoverable because $P(w, z, y)$ is non-recoverable (by theorem 4) and $P(y|z, w)$ is not recoverable (by corollary 2). However, $P(y|do(z))$*

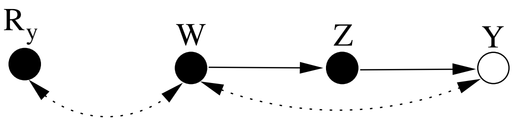


Figure 3.5: m-graph in which Y and R_y are not separable but still $P(Y|do(Z))$ is recoverable.

is recoverable as demonstrated below:

$$P(y|do(z)) = P(y|do(z), R'_y) = \sum_w P(y|do(z), w, R'_y)P(w|do(z), R'_y) \tag{3.7}$$

$$P(y|do(z), w, R'_y) = P(y|z, w, R'_y) \textit{ (by Rule-2 of do-calculus ([Pea09]))} \tag{3.8}$$

$$P(w|do(z), R'_y) = P(w|R'_y) \textit{ (by Rule-3 of do-calculus) }) \tag{3.9}$$

Substituting (3.8) and (3.9) in (3.7) we get:

$$P(y|do(z)) = \sum_w P(y|z, w, R'_y)P(w|R'_y) = \sum_w P(y^*|z, w, R'_y)P(w|R'_y)$$

The recoverability of $P(y|do(z))$ in the previous example follows from the notion of d*-separability and dormant independence ([SP08]).

Definition 5 (d^*-separation ([SP08])). *Let G be a causal diagram. Variable sets X, Y are d^*-separated in G given Z, W (written $X \perp_w Y|Z$), if we can find sets Z, W, such that $X \perp Y|Z$ in $G_{\overline{w}}$, and $P(y, x|z, do(w))$ is identifiable.*

Definition 6 (Inducing path ([VP91])). *An path p between X and Y is called inducing path if every node on the path is a collider and an ancestor of either X or Y.*

Theorem 6. *Given an m-graph in which $|V_m| = 1$ and Y and R_y are connected by an inducing path, $P(y|do(x))$ is recoverable if there exists Z, W such that $Y \perp_w R_y|Z$ and for $W = W \setminus X$, the following conditions hold:*
(1) $Y \perp\!\!\!\perp W_1|X, Z$ in $G_{\overline{X},\underline{W_1}}$ and
(2) $P(W_1, Z|do(X))$ and $P(Y|do(W_1), do(X), Z, R'y)$ are identifiable.
Moreover, if recoverable then,

$$P(y|do(x)) = \textstyle\sum_{W_1,Z} P(Y|do(W), do(X), Z, R'_y)P(Z, W_1|do(X))$$

We can now quickly glance at the m-graph in figure 3.5 and conclude that $P(y|do(z))$ is recoverable by verifying that the conditions in theorem 6 hold in the m-graph.

3.6 Attrition

Attrition (i.e. participants dropping out from a study/experiment), is a ubiquitous phenomenon, especially in longitudinal studies. In this section, we shall discuss a special case of attrition called 'Simple Attrition' (for an in-depth treatment see [Gar13]). In this problem, a researcher conducts a randomized trial, measures a set of variables (X,Y,Z) and obtains a dataset where outcome (Y) is corrupted by missing values (due to attrition). Clearly, due to randomization, the effect of treatment (X) on outcome (Y), $P(y|do(x))$, is identifiable and is given by $P(Y|X)$. We shall now demonstrate the usefulness of our previous discussion in recovering $P(y|do(x))$. Typical attrition problems are depicted in figure 3.6. In Figure 3.6 (b) we can apply theorem 2 to recover $P(y|do(x))$ as given below: $P(Y|X) = \sum_Z P(Y^*|X, Z, R'_y)P(Z|X)$. In Figure 3.6 (a), we observe that Y and R_y are connected by a collider path. Therefore by corollary 2, $P(Y|X)$ is not recoverable; hence $P(y|do(x))$ is also not recoverable.

3.6.1 Recovering Joint Distributions under simple attrition

The following theorem yields the *necessary and sufficient* condition for recovering joint distributions from semi-Markovian models with a single partially observed variable i.e. $|V_m| = 1$ which includes models afflicted by simple attrition.

Theorem 7. *Let $Y \in V_m$ and $|V_m| = 1$. $P(V)$ is recoverable in m-graph G if and only if Y and R_y are not neighbors and Y and R_y are not connected by a path in which all intermediate nodes are colliders. If both conditions are satisfied, then $P(V)$ is given by, $P(V) = P(Y|V_O, R_y = 0)P(V_O)$*

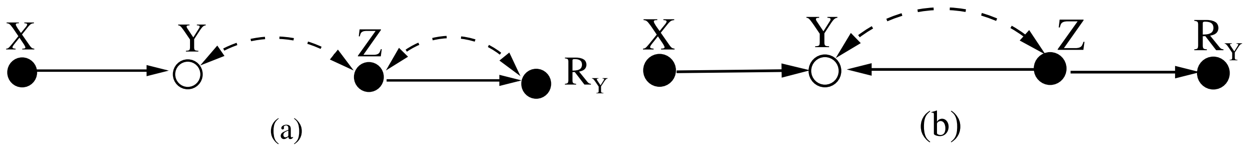


Figure 3.6: (a) m-graphs in which $P(y|do(x))$ is not recoverable (b) m-graphs in which $P(y|do(x))$ is recoverable.

3.6.2 Recovering Causal Effects under Simple Attrition

Theorem 8. *$P(y|do(x))$ is recoverable in the simple attrition case (with one partially observed variable) if Y and R_y are neither neighbors nor connected by an inducing path. Moreover, if recoverable,*

$$P(Y|X) = \sum_{z} P(Y^*|X, Z, R_y')P(Z|X) \tag{3.10}$$

where Z is the separating set that d-separates Y from R_y.

3.7 Summary

We presented graphical conditions for recovering joint and conditional distributions and sufficient conditions for recovering causal queries. We further identified conditions that forbid recovery of joint and conditional distributions. We exemplified the recoverability of causal queries of the form $P(y|do(x))$ despite the existence of an inseparable path between Y and R_y. We applied our results to problems of attrition and presented necessary and sufficient graphical conditions for recovering causal effects in such problems.

CHAPTER 4

Advanced Algorithms for Recoverability

The recoverability procedures presented thus far relied entirely on conditional independencies that are read off the m-graph using the d-separation criterion. Interestingly, recoverability can sometimes be accomplished by graphical patterns other than conditional independencies. These patterns represent distributional constraints which can be detected using mutilated versions of the m-graph. In this chapter we develop techniques for constraint based recovery assisted by do-calculus.

This chapter builds on our previous work [SMP15], that developed an algorithm to recover joint distributions in Markovian models. However, the algorithm in [SMP15] is sufficient but not complete as evidenced by example 10 (section 4.2.1), in which recoverability of joint distribution in a Markovian model cannot be established using results in [SMP15].

This chapter presents a unified approach to recovering causal and pribabilistic queries. To this end, we develop a general algorithm that can recover conditional probability distributions and conditional causal effects in Semi-markovian models.

Chapter Outline: In Section 4.1 we present the definition of *inducing path* and rules of do-calculus. Examples of recoverability of probabilistic queries using do-calculus are in section 4.2. Sections 4.3, 4.4 and 4.5 present an overview of this approach, define the factorization scheme and detail a general algorithm for recovering conditional probabilistic/interventional distributions. Finally, section 4.6 summarizes the results.

4.1 Preliminaries: Inducing Path, do-calculus

Definition 7 (Inducing Path [VP91]). *A path p between nodes A and B is called an inducing path if all the intermediate nodes on p are colliders and ancestors of A and/or B.*

In the presence of such an inducing path, $\nexists C$ such that $A \perp\!\!\!\perp B|C$. In this chapter we detail how $P(X)$ can be recovered when R_x and X are connected by inducing path(s).

The following description of do-calculus has been modified from [Pea09], Theorem 3.4.1.

4.1.1 do-calculus [Pea09]

Let G be a causal Bayesian network and and let $P(\cdot)$ stand for the probability distribution induced by that model. $G_{\overline{X}}$ is the graph obtained by deleting from G all arrows pointing to nodes in X. Likewise, we denote by $G_{\underline{X}}$ the graph obtained by deleting from G all arrows emerging from nodes in X. For any disjoint subsets of variables X, Y, Z, and W, we have the following rules.

Rule 1 (Insertion/deletion of observations):
$P(y|\hat{x}, z, w) = P(y|\hat{x}, w)$ if $Y \perp\!\!\!\perp Z|X, W$ in $G_{\overline{X}}$.

Rule 2 (Action/observation exchange):
$P(y|\hat{x}, \hat{z}, w) = P(y|\hat{x}, z, w)$ if $Y \perp\!\!\!\perp Z|X, W$ in $G_{\overline{X}\underline{Z}}$.

Rule 3 (Insertion/deletion of actions):
$P(y|\hat{x}, \hat{z}, w) = P(y|\hat{x}, w)$ if $Y \perp\!\!\!\perp Z|X, W$ in $G_{\overline{X}\,\overline{Z(W)}}$.
where Z(W) is the set of Z-nodes that are not ancestors of any W-node in $G_{\overline{X}}$.

4.2 Constraint based Recoverability: Examples

In this section we present two examples of recoverability in the presence of inducing paths. In the first one the inducing path is explicitly visible in the m-graph whereas in the second one, the inducing path is hidden and surfaces later on during the analysis.

Example 10. *Let G be the m-graph in figure 4.1 (a) and let the query of interest be $P(X)$. The absence of a set that d-separates X from R_x, makes it impossible to apply any of the techniques discussed previously. While it may be tempting to conclude that $P(X)$ is not recoverable, we prove otherwise, using Verma constraints ([VP91, TP02b]). The proof presented below is based on the*

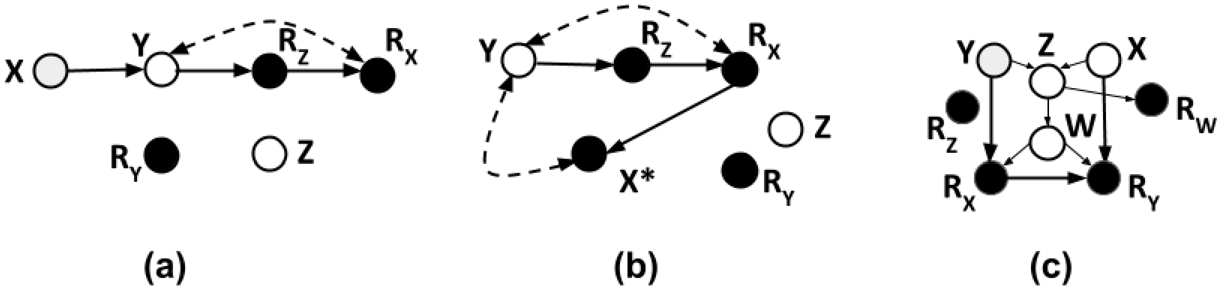


Figure 4.1: (a) & (c) m-graphs from which conditional distributions can be recovered aided by intervention, (b) latent structure [Pea09, Chapter 2] corresponding to m-graph in (a) when X is treated as a latent variable.

rules of do-calculus.

$$\begin{aligned} P(X) &= P(X|do(R_z=0)) \textit{ (Rule-3 of do-calculus)} \\ &= P(X|do(R_z=0), R_x=0) \textit{ (Rule-1 of do-calculus)} \\ &= P(X^*|do(R_z=0), R_x=0) \textit{ (using equation 2.1)} \end{aligned} \tag{4.1}$$

Note that the query of interest is now a function of X^* and not X. Therefore the problem now amounts to identifying a conditional interventional distribution using the m-graph in figure 4.1(b). A complete analysis of such problems is available in [SP06] and [HV12]. The causal effect in eq 4.1 is identified as:

$$P(X) = \sum_Y P(X^*|Y, R_x=0, R_z=0)\frac{P(R_x=0|Y,R_z=0)P(Y)}{\sum_Y P(R_x=0|Y,R_z=0)P(Y)}$$

The above equaltion is not in its recovered form because it contains Y, a partially observed variable. However, since $X^* \perp\!\!\!\perp R_y|R_x, Y, R_z$ and $(Y, R_x, R_z) \perp\!\!\!\perp R_y$, we have the license to insert $R_y = 0$ as shown below,

$$P(X) = \sum_Y P(X^*|Y^*, R_x=0, R_z=0, R_y=0)\frac{P(R_x=0|Y^*,R_z=0,R_y=0)P(Y^*|R_y=0)}{\sum_Y P(R_x=0|Y^*,R_z=0,R_y=0)P(Y^*|R_y=0)} \tag{4.2}$$

In addition to $P(X)$, this graph also allows recovery of $P(X, Z)$ as shown below.

$$\begin{aligned}
&P(X, Z) = P(X)P(Z) \\
&= \sum_Y P(X^*|Y^*, R_x = 0, R_z = 0, R_y = 0)\frac{P(R_x = 0|Y^*, R_z = 0, R_y = 0)P(Y^*|R_y = 0)}{\sum_Y P(R_x = 0|Y^*, R_z = 0, R_y = 0)P(Y^*|R_y = 0)}P(Z) \\
&= \sum_Y P(X^*|Y^*, R_x = 0, R_z = 0, R_y = 0)\frac{P(R_x = 0|Y^*, R_z = 0, R_y = 0)P(Y^*|R_y = 0)}{\sum_Y P(R_x = 0|Y^*, R_z = 0, R_y = 0)P(Y^*|R_y = 0)}P(Z^*|R_z = 0)
\end{aligned}$$

The decomposition in the first line uses $X \perp\!\!\!\perp Z$ while recoverability of $P(XY)$ in the last line follows from theorem 2, since $Z \perp\!\!\!\perp R_z$.

Remark 4. *In the preceding example we were able to recover $P(X)$ despite the fact that X and R_x were not independent. The ability to exploit such cases further underscores the need for graph based analysis.*

A more complex example detailing recoverability of joint distribution from the m-graph in figure 4.1 (c), where the inducing path is not immediately visible, is presented below.

4.2.1 A Complex Example of Recoverability

We use $R = 0$ as a shorthand for the event where all variables are observed i.e. $R_{V_m} = 0$.

Example 11. *Given the m-graph in figure 4.1 (c), we will now recover the joint distribution.*

$$P(W, X, Y, Z) = P(W, X, Y, Z)\frac{P(W, X, Y, Z, R = 0)}{P(W, X, Y, Z, R = 0)} = \frac{P(W, X, Y, Z, R = 0)}{P(R = 0|W, X, Y, Z)}$$

Factorization of the denominator based on topological ordering of R variables yields:

$$P(W, X, Y, Z) = \frac{P(W, X, Y, Z, R = 0)}{P(R_y = 0|W, X, Y, Z, R_x = 0, R_w = 0, R_z = 0)P(R_x = 0|W, X, Y, Z, R_w = 0, R_z = 0)}$$
$$\frac{1}{P(R_w = 0|W, X, Y, Z, R_z = 0)P(R_z = 0|W, X, Y, Z)}$$

On simplifying each factor of the form: $P(R_a = 0|B)$, by removing from it all $B_1 \in B$ such that $R_a \perp\!\!\!\perp B_1|B - B_1$, we get:

$$P(W, X, Y, Z) = \frac{P(W, X, Y, Z, R = 0)}{P(R_z = 0)P(R_w = 0|Z)P(R_y = 0|X, W, R_x = 0)P(R_x = 0|Y, W)} \tag{4.3}$$

$P(WXYZ)$ is recoverable if all factors in the preceding equation are recoverable. Examining each factor one by one we get:

- $P(W, X, Y, Z, R=0)$*: Recoverable as* $P(W^*, X^*, Y^*, Z^*, R=0)$ *using equation 2.1.*

- $P(R_z = 0)$*: Directly estimable from the manifest distribution.*

- $P(R_w = 0|Z)$*: Recoverable as* $P(R_w = 0|Z^*, R_z = 0)$*, using* $R_w \perp\!\!\!\perp R_z|Z$ *and equation 2.1.*

- $P(R_y = 0|X, W, R_x = 0)$*: Recoverable as* $P(R_y = 0|X^*, W^*, R_x = 0, R_w = 0)$*, using* $R_y \perp\!\!\!\perp R_w|X, W, R_x$ *and equation 2.1.*

- $P(R_x = 0|Y, W)$*: The procedure for recovering* $P(R_x = 0|Y, W)$ *is rather involved and requires converting the probabilistic sub-query to a causal one as detailed below.*

$$\begin{aligned} P(R_x = 0|Y, W = w) &= P(R_x = 0|Y, do(W = w))\textit{(Rule-2 of do calculus)} \\ &= \frac{P(R_x = 0|Y, R_y = 0, do(w))}{P(R_x = 0|Y, R_y = 0, do(w))} P(R_x = 0|Y, do(W = w)) \\ &= P(R_x = 0|Y, R_y = 0, do(w)) \frac{P(R_y = 0|Y, do(w))}{P(R_y = 0|Y, do(w), R_x = 0)} \end{aligned} \tag{4.4}$$

To prove recoverability of $P(R_x = 0|Y, W = w)$*, we have to show that all factors in equation 4.4 are recoverable.*

Recovering $\mathbf{P(R_y = 0|Y, do(w), R_x = 0)}$: *Observe that* $P(R_y = 0|Y, do(w), R_x = 0) = P(R_y = 0|do(w), R_x = 0)$ *by Rule-1 of do calculus. Recoverability of* $P(R_y = 0|do(w), R_x = 0)$ *follows from recoverability of* $P(X^*, Y^*, R_x, R_y, Z|do(w))$ *in* G'*, the m-graph corresponding to* G *in which* X *and* Y *are treated as latent variables.*

$$\begin{aligned} P(X^*, Y^*, R_x, R_y, Z|do(w)) &= P(X^*, Y^*, R_x, R_y|Z, do(w))P(Z|do(w)) \\ &= P(X^*, Y^*, R_x, R_y|Z, w)P(Z|do(w)) \textit{ (Rule-2 of do-calculus)} \\ &= P(X^*, Y^*, R_x, R_y|Z, w)P(Z) \textit{ (Rule-3 of do-calculus)} \end{aligned}$$

Using $(X^*, Y^*, R_x, R_y) \perp\!\!\!\perp (R_z, R_w)|(Z, W)$*, equation 2.1 and* $Z \perp\!\!\!\perp R_z$ *we show that the causal effect is recoverable as:*

$$P(X^*, Y^*, R_x, R_y, Z|do(w)) = P(X^*, Y^*, R_x, R_y|Z^*, w^*, R_w = 0, R_z = 0)P(Z^*|R_z = 0) \tag{4.5}$$

Recovering $\mathbf{P(R_x = 0|Y, do(w), R_y = 0)}$: *Using equation 2.1, we can rewrite* $P(R_x = 0|Y, do(w), R_y = 0)$ *as* $P(R_x = 0|Y^*, do(w), R_y = 0)$. *Its recoverability follows from equation 4.5.*

Recovering $\mathbf{P(R_y = 0|Y, do(w))}$:

$$\begin{aligned} P(R_y = 0|Y, do(w)) &= \frac{P(R_y = 0, Y|do(w))}{\sum_{R_x} P(R_y = 0, Y, R_x|do(w)) + P(R_y = 1, Y, R_x|do(w))} \\ &= \frac{P(R_y = 0, Y^*|do(w))}{\sum_{R_x} P(R_y = 0, Y^*, R_x|do(w)) + P(R_y = 1, Y, R_x|do(w))} \textit{(using eq 2.1)} \end{aligned}$$

$P(R_y = 0, Y^*|do(w))$ *and* $P(R_y = 0, Y^*, R_x|do(w))$ *are recoverable from equation 4.5. We will now show that* $P(R_y = 1, Y^*, R_x|do(w))$ *is recoverable as well.*

$$P(R_y = 1, Y, R_x|do(w)) = \frac{P(R_y = 0, Y, R_x|do(w))}{P(R_y = 0|R_x, Y|do(w))} - P(R_y = 0, R_x, Y|do(w))$$

Using equation 2.1 and Rule-1 of do-calculus we get,

$$= \frac{P(R_y = 0, Y^*, R_x|do(w))}{P(R_y = 0|R_x, do(w))} - P(R_y = 0, R_x, Y^*|do(w))$$

Each factor in the preceding equation is estimable from equation 4.5. Hence $P(R_y = 1, Y, R_x, do(w))$ *and therefore,* $P(R_y = 0|Y, do(w))$ *is recoverable. Since all factors in equation 4.4 are recoverable, the joint distribution is recoverable.*

We formalize a general procedure for recoverability in the following sections.

4.3 Unified Approach to Recoverability of Causal and Probabilistic Queries: An Overview

Previous examples clearly show that general procedures for recovering conditional probability distributions should be able to recover conditional causal effects as well. Similarly, our discussions on recovering causal effects in chapter 3 confirm the need for procedures recovering conditional causal effects to be able to handle conditional probability distributions. Our solution to recovering both probabilistic and causal queries relies on the notion of **partial-recoverability**. A query of the form $P(A|B, do(C))$ is said to be *partially-recovered* if $(A \cup B) \cap V_m = \emptyset$. In words, while A and B can contain proxy variables, fully observed variables or even R variables, they must not contain partially observed variables.

We shall now outline the steps for recovering causal effects of the form $P(X|Y, do(D))$. First using rules of algebra, probability theory and do-calculus, factorize $P(X|Y, do(D))$, so that each factor is partially recoverable. Second apply causal identification algorithm to obtain an estimand in terms of conditional probability distributions, $P(A_i|B_i)$. Finally if $P(A_i|B_i)$ is recoverable for all i, output the recovered estimand. On the other hand, if the query of interest is a conditional probability distribution of the form $P(X|Y)$, we proceed in a similar manner, with the exception that we will skip the second step if all the factors are purely probabilistic.

The following section presents a general factorization scheme inspired by theorem 5, that can be applied to recover probabilistic and causal queries from missing data.

4.4 General Missingness Factorization

Definition 8 (General Missingness Factorization)**.** *Let X and Y be sets of variables. Let $D \subseteq X_m \cup Y_m$ be a maximal set such that $\forall D_1 \in D$, $R_{D_1} \notin X \cup Y$. The factorization:*

$$P(X|Y, \hat{w}) = \frac{P(X|Y, R_D = 0, \hat{w})P(R_D = 0|Y, \hat{w})}{P(R_D = 0|X, Y, \hat{w})} \tag{4.6}$$

is called General Missingness Factorization.

Remark 5. *X and Y in the preceding definition are not restricted to being subsets of $V_m \cup V_o$. Furthermore, the above factorization is applicable to both conditional interventional and probabilistic distributions.*

Let $A_1 < A_2 < ... < A_n$ denote a topological ordering of variables in set A such that all child nodes are ordered before their respective parent nodes. $A^{(i)}$ denotes the set $\{A_i, A_{i+1}, ..., A_n\}$ and $A^{(0)} = \emptyset$. For example, in figure 4.1(c), a valid topological ordering of R variables is $R_Z < R_W < R_X < R_Y$.

Lemma 1. *Given eq 4.6, $P(R_D = 0|Y, \hat{w})$ and $P(R_D = 0|X, Y, \hat{w})$ can be factorized as:*

$$P(X|Y, \hat{w}) = \frac{P(X|Y, R_D = 0, \hat{w}) \prod_{R_i \in (R_D)} P(R_i = 0|Y, R^{(i-1)}, \hat{w})}{\prod_{R_i \in R_D} P(R_i = 0|Y, X, R^{(i-1)}, \hat{w})} \tag{4.7}$$

Let $Z_i = \{Y, R^{(i-1)}\}$. Then the factors in the numerator and denominator are compactly denoted as $P(R_i = 0|Z_i)$ and $P(R_i = 0|Z_i, X)$, respectively.

Lemma 2. *If $R_i \perp\!\!\!\perp X|Z_i$ in $G_{\overline{W}}$, equation 4.7 can be further simplified by removing from it both $P(R_i = 0|Z_i, \hat{w})$ and $P(R_i = 0|X, Z_i, \hat{w})$.*

Observe that there exists a term $P(R_i = 0|Z_i, \hat{w})$ in the numerator corresponding to every term $P(R_i = 0|Z_i, X, \hat{w})$ in the denominator and the following lemma details a procedure for recoverability of $P(R_i = 0|Z_i, \hat{w})$ when $P(R_i = 0|Z_i, X, \hat{w})$ is recoverable.

Lemma 3 (Recoverability of $P(R_i = 0|Z_i, \hat{w})$ from $P(R_i = 0|Z_i, X, \hat{w})$)**.** *If $P(R_i = 0, Z_i|\hat{w})$ and $\forall X = x$, $P(R_i = 0|X, Z_i, \hat{w})$ and $P(R_i = 0, X, Z_i|\hat{w})$ are recoverable, then $P(R_i = 0|Z_i, \hat{w})$ can be recovered as, $\frac{P(R_i=0,Z_i|\hat{w})}{\sum_X P(R_i=0,X,Z_i|\hat{w}) + \frac{P(R_i=0,X,Z_i|\hat{w})}{P(R_i=0|X,Z_i,\hat{w})} - P(R_i=0,X,Z_i|\hat{w})}$.*

Proof. $P(R_i = 1, X, Z_i|\hat{w}) = \frac{P(R_i=0,X,Z_i|\hat{w})}{P(R_i=0|X,Z_i,\hat{w})} - P(R_i = 0, X, Z_i|\hat{w})$ (a)

$P(R_i = 0|Z_i, \hat{w})$= $\frac{P(R_i=0,Z_i|\hat{w})}{\sum_X P(R_i=0,X,Z_i|\hat{w}) + P(R_i=1,X,Z_i|\hat{w})}$ (b)

Substituting (a) in (b), we get the estimand for $P(R_i = 0|Z_i, \hat{w})$.

□

4.5 Algorithm for Recovering $P(X|Y)$

The following is an algorithm to recover both conditional causal and probabilistic distributions.

Algorithm 1: Recover($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{P}, \mathbf{G}, \mathbf{History}$)

```
if querySeenBefore($History, P(X|Y, \hat{D}), G$) then
    return FAIL
$Y \leftarrow$ pruneQuery($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{G}$)
```

```
$G \leftarrow$ getAncestralGraph($\mathbf{X} \cup \mathbf{Y} \cup \mathbf{D}, \mathbf{G}$)
if $E \leftarrow$ getRecoveredEstimand($P(X|Y, \hat{D}), G$) ! = FAIL then
    if $D \neq \emptyset$ then
        $E \leftarrow$ recoverCausal($E, P, G, History \cup \{E, G\}$)
    return $E$
$S \leftarrow$ addVariables($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{G}$)
$\frac{F_1}{F_2} \leftarrow$ gmf($P(X \cup S|Y, \hat{D}), G$)
if $S == \emptyset$ & $F_2 == \emptyset$ & $F_1 == P(X|Y, \hat{D})$ then return FAIL
$A \leftarrow \emptyset$
$(B, Latent, flag) \leftarrow recoverFactors(P(X \cup S|Y, \hat{D}), F_2, P, G, History)$
if flag then
    return $\sum_S B$
if $B == FAIL$ then
    if $Latent == \emptyset$ then
        return FAIL
    $G^* \leftarrow getLatentGraph(Latent, G)$
    return Recover$(P(X|Y, \hat{D}), P, G^*, History \cup \{P(X|Y, \hat{D}), G^*\})$
for every factor $P(R_{v_i} = 0|Z_i)$ in $F_1$ do
    if  $P(R_{v_i} = 0|Z_i)$ can be recovered using lemma 3 then
        Let $E$ be the estimand returned by lemma 3
        $A \leftarrow A \cup E$
Let $F$ be the factors of the form $P(R_{v_i} = 0|Z_i)$ in $F_1$ that were not recoverable using lemma
    3 in the previous step
$(C, Latent, flag) \leftarrow recoverFactors(P(X|Y, \hat{D}), F, P, G, History)$
if flag then
    return $\sum_S C$
if $C == FAIL$ then
    if $Latent == \emptyset$ then
```

```
        **return** FAIL
    $G^* \leftarrow getLatentGraph(Latent, G)$
    **return** Recover($P(X|Y, \hat{D}), P, G^*, History \cup \{P(X|Y, \hat{D}), G^*\}$)
Let $f$ be the factor in $F_1$ corresponding to $P(X|Y, R_W = 0, , \hat{D})$ in equation 4.7
$E \leftarrow$ Recover($f, P, G, History \cup \{f, G\}$)
**if** $E ==$ FAIL **then return** FAIL
**return** $\sum_S E \frac{\prod_i A[i] \prod_j C[j]}{\prod_k B[k]}$
```

Following subsection briefly describes each function that the algorithm invokes.

4.5.1 Description of Functions Invoked by Algorithm 1

The following function checks if the current query has been processed in the current context (call stack). The goal is to prevent infinite recursion of the recover procedure.

Function querySeenBefore($History, Q, G$)

1. if $(Q, G) \in History$ then **return** TRUE
2. if $History == \emptyset$ then $History \leftarrow History \cup (Q, G)$
3. **return** FALSE

The following function removes redundant variables from the query. For instance given $G: X \rightarrow Y \rightarrow Z \rightarrow R_x$ and the query $P(X|Y, Z)$ as input, the function would prune the set $\{Y, Z\}$ to $\{Y\}$. This is permissible since $X \perp\!\!\!\perp Z|Y$ and recovering $P(X|Y)$ is equivalent to recovering $P(X|Y, Z)$.

Function pruneQuery($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{G}$)

1. $\forall Y_1 \in Y$, if $X \perp\!\!\!\perp Y_1|Y - \{Y_1\}, D$ in $G_{\overline{D}}$, then $Y \leftarrow Y - \{Y_1\}$
2. return Y

The following function recursively removes variables that are not pertinent to the recovery procedure. License for this operation follows from the d-separation criterion. As stated in step-9, in cases where a partially observed variable X is removed in the resultant graph while its mechanism R_x is retained, R_x will henceforth be treated as a fully observed variable as opposed to an R variable.

Function getAncestralGraph($\mathbf{X}, \mathbf{G}$)

```
1. $\mathbf{Y} \leftarrow \mathbf{X}$
2. $\forall \mathbf{x} \in \mathbf{X}$, if $\exists \mathbf{R_x}$, then add $\mathbf{R_x}$ to $\mathbf{Y}$
3. Mark all $\mathbf{y} \in \mathbf{Y}$ in $\mathbf{G}$
4. $\mathbf{A} = \emptyset$
5. $\forall \mathbf{y} \in \mathbf{Y}$ add $parent(\mathbf{y})$ to $\mathbf{A}$, as long as $parent(\mathbf{y}) \notin \mathbf{Y}$
6. if $\mathbf{A} \neq \emptyset$ then return getAncestralGraph($\mathbf{A}, \mathbf{G}$)
7. Let $G^*$ be the sub-graph of $\mathbf{G}$ comprising of all marked nodes in $\mathbf{G}$.
8. $\forall$ partially observed variables $X$ in $G^*$, add to $G^*$ proxy variable $X^*$, and the edges $R_x \rightarrow X^*$
   and $X \rightarrow X^*$
9. $\forall R_x \in G^*$ such that $X \notin G^*$
   $R \leftarrow R - \{R_x\}$
   $V_o \leftarrow V_o \cup \{R_x\}$
10. return $G^*$
```

The following function checks if $P(X|Y, \hat{D})$ is partially recoverable.

Function getRecoveredEstimand($P(X|Y, \hat{D}), G$)

```
1. if $X == \emptyset$, return $\emptyset$
2. $P(X|Y, \hat{D}) \leftarrow$ toProxy($P(X|Y, \hat{D})$)
3. Let $Z \leftarrow (X \cup Y) \cap V_m$
```

```
4. if $Z == \emptyset$ return $P(X|Y)$
5. if $X \not\perp\!\!\!\perp R_z|Y, D$ in $G_{\overline{D}}$ return FAIL
6. return $toProxy(P(X|Y \cup \{R_Z = 0\}, \hat{D}))$
```

The following function applies equation 2.1 to the input query, $P(X|Y, \hat{D})$ to convert partially observed variables in the query to proxy variables.

Function toProxy($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}})$)

```
1. $\forall Z \in Y \cap V_m$ such that $R_z \in Y$ and $R_z$ assumes the value 0, $Y \leftarrow (Y - \{Z\}) \cup \{Z^*\}$
2. $\forall Z \in X \cap V_m$ such that $R_z \in Y \cup X$ and $R_z$ assumes the value 0, $X \leftarrow (X - \{Z\}) \cup \{Z^*\}$
3. return $P(X|Y, \hat{D})$
```

The following function adds variables that could possibly aid the recovery procedure, while taking care not to add variables (such as colliders and their descendants) that can open the path between any $X_i \in X$ and R_{X_i}. For instance, to recover $P(X)$ given $G : X \rightarrow Y \rightarrow R_x$, it is vital to include Y in the analysis, whereas given $G : X \rightarrow Y <-- > R_x$, it is important to not include Y.

Function addVariables($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{G}$)

```
1. $S \leftarrow \emptyset$
2. $\forall Z \in V_m \cup V_o - \{X, Y, D\}$
    (a) if there does not exist $X_1 \in X \cap V_m$ such that $Z$ is a collider or descendant of a collider on any path between $X_1$ and $R_{X_1}$, then $S \leftarrow S \cup \{Z\}$
3. return $S$
```

The following function performs general missingness factorization on the input query $P(A|B, \hat{w})$ and further simplifies the estimand by invoking lemmata 1 and 2, before returning the resulting estimand.

Function gmf($\mathbf{P(A|B, \hat{w}), G}$)

1. Let $\frac{F_1}{F_2}$ correspond to the fraction returned by applying definition 8 and lemmata 1 and 2 respectively on $P(A|B, \hat{w})$
2. return $\frac{F_1}{F_2}$

The following function constructs a latent projection [VP91, Pea09, SMP15] and returns the resulting graph G_l in which all variables in X will be treated as latent and not explicitly portrayed in the graph. An example is shown in figure 4.1(b).

Function getLatentGraph($\mathbf{X}, \mathbf{G}$)

1. Let G_l be the latent projection corresponding to X
2. Return G_l

The following function handles inducing paths by first identifying variables to be intervened upon and then converting the probabilistic query to a causal one, using rules of do-calculus. If the causal effect is recoverable, the recovered estimand is returned, else a FAIL message is returned.

C-component of X (also known as district of X) is the set of variables (including X) that is connected to X by a path comprising bi-directed edges.

Function handleInducingPath($P(Y|X, \hat{W}), Z, P, G, H$)

1. Let C denote the set of all C-components of R_Z
2. $D \leftarrow (Parents(C) - C) \cap Ancestors(R_Z)$
3. $\forall$ minimal $D_1 \subseteq D$ such that no inducing paths exist between Z_i and $R_{Z_i}, \forall i$ in $G_{\overline{D_1}}$

   (a) Let $(D_2 \leftarrow D_1 \cap X)$

   (b) if $P(Y|X, \hat{W})! = P(Y|X - D_2, \hat{D_2}, \hat{W})$ as per Rule-2 of do-calculus
   continue

   (c) if $P(Y|X - D_2, \hat{D_2}, \hat{W})! = P(Y|X - D_2, \hat{D_1}, \hat{W})$ as per Rule-3 of do-calculus
   continue

   (d) if $(E \leftarrow Recover(P(Y|X - D_2, \hat{D_1}, \hat{W}), P, G, H \cup \{P(Y|X - D_2, \hat{D_1}, G\}))! = FAIL$
   return E

4. return FAIL

The following function recovers factors of the form $P(R_i = 0|Z_i)$ contained in set F. The function performs the following tasks: (i) Partitioning the factors into 'recovered' and 'failed to recover' groups stored in lists $SuccessList$ and $FailList$ respectively (lines 8-11). (ii) Identifying variables which when removed from analysis (i.e. treated as latent) will most likely result in recovery of the query of interest (lines 14-20). (iii) Throwing a FAIL when FailList is not empty and ensuring that the variables to be removed are not part of the query of interest (lines 21,22).

Function recoverFactors($\mathbf{P}(\mathbf{X}|\mathbf{Y}, \hat{\mathbf{D}}), \mathbf{F}, \mathbf{P}, \mathbf{G}, \mathbf{History}$)

```
1: $SuccessList \leftarrow \emptyset$, FailList $\leftarrow \emptyset$, $G^* \leftarrow G$
2: **for** every factor $P(R_{v_i} = 0|Z_i, \hat{W})$ in $F$ **do**
3:     **if** $V_i \notin Z$ **then** $G^* \leftarrow$ getLatentGraph $(V_i, G)$
4:     **if** $(Z \leftarrow detectInducingPath(P(X|Y, \hat{D}), G^*))! = \emptyset$ **then**
5:         $E \leftarrow$ handleInducingPath($P(X|Y, \hat{D}), Z, P, G, History$)
6:         **if** $E! = FAIL$ **then**
7:             **return** $(E, \emptyset, true)$
8:     **else**
9:         $E \leftarrow$ Recover($P(R_{v_i} = 0|Z_i, \hat{W}), P, G^*, History \cup \{P(R_{v_i} = 0|Z_i, \hat{W}), G^*\}$)
10:     **if** $E \neq$ FAIL **then**
11:         $SuccessList \leftarrow SuccessList \cup E$
```

```
12:     else
13:         FailList ← $FailList \cup P(R_{v_i} = 0|Z_i, \hat{W})$
14: if FailList $!= \emptyset$ then
15:     Latent ← $\emptyset$
16:     for every factor $P(R_{v_i} = 0|Z_i, \hat{W})$ in FailList  do
17:         $t \leftarrow true$
18:         for every $V_j \in Z_i \cap V_m$  do
19:             if $P(R_{v_j} = 0|Z_j, \hat{W}) \in$ FailList  then
20:                 $t \leftarrow false$
21:         if $t$ then
22:             $Latent \leftarrow Latent \cup V_i$
23:     if $Latent \cap \{X, Y\} \neq \emptyset$ then return (FAIL, $\emptyset, false$)
24:         return (FAIL,Latent, false)
25: return $(SuccessList, \emptyset, false)$
```

To better understand how this function works, consider the following example.

Example 12. *Let G be $X \rightarrow Z \rightarrow W \rightarrow R_x$ and $Z \rightarrow R_Z$, and the query of interest be $P(X)$. The function addVariables will modify the query as,*

$$P(X) = \sum_{Z,W} P(X, Z, W)$$

On applying gmf function we get,

$$= \frac{P(X, Z, W|R_X = 0, R_Z = 0)P(R_x = 0|R_Z = 0)P(R_Z = 0)}{P(R_x = 0|Z, X, W, R_z = 0)P(R_z = 0|Z, X, W)}$$

While processing the factors in the denominator of the above equation, FailList will contain $P(R_z = 0|Z, X, W)$ and SuccessList will contain all other factors. Lines 16-20 will add Z to 'Latent'.

Notice that by treating Z as a latent variable and removing it entirely from the analysis, $P(X)$ can be recovered as $\sum_W P(XW) = \frac{P(X^*,W|R_x=0)P(R_x=0)}{P(R_x=0|W)}$. On the other hand, if instead of $P(X)$, the query was $P(X, Z)$, we could not have removed Z from the analysis.

The following function uses the function *identify* to invoke an appropriate algorithm [SP06, HV12, TP02a] to recover causal effect. E as returned by this algorithm will comprise of summations and conditional probability distributions and is assumed to be in its simplified form. In other words, recoverability of E is guaranteed by recoverability of constituent conditional distributions of the form $P(A_i|B_i)$.

```
Function recoverCausal($\mathbf{P(X|Y, \hat{D}), P, G, History}$)
if $\{X, Y\} \cap V_m \neq \emptyset$ then return FAIL
$Z^* \leftarrow \{X, Y\} \cap V^*$
$G^* \leftarrow getLatentGraph(Z, G)$
Let $P^*$ be the manifest distribution corresponding to m-graph $G^*$, computed from $P$.
$E \leftarrow identify(P(X|Y, \hat{D}), P^*, G^*)$
if $E == FAIL$ then return FAIL
for every $P(A_i|B_i) \in E$ do
    if $t \leftarrow recover(P(A_i|B_i), P, G, History \cup \{P(A_i|B_i), G\})$ then
        if $t == FAIL$ then return FAIL
        Replace $P(A_i|B_i)$ in $E$ with the recovered estimand $t$
return $E$
```

The following function returns the list of all $X_i \in X$ that are connected to R_{X_1} by inducing path(s).

```
Function detectInducingPath($P(X|Y, \hat{D}), G$)
$IP \leftarrow \emptyset$
$P(X|Y, \hat{D}) \leftarrow$ toProxy ($P(X|Y, \hat{D})$)
for every $X_1 \in X \cap V_m$ do
    if there exists inducing paths between $X_1$ and $R_{X_1}$ in $G$ then
        $IP \leftarrow IP \cup X_1$
for every $R_w \in X \cap R$ do
```

```
    **if** $W \in Y$ and there exists inducing paths between $W$ and $R_w$ in $G$ **then**
        $IP \leftarrow IP \cup W$
**return** $IP$
```

4.5.2 Description of Algorithm 1

The algorithm initially pre-processes the input query and graph, then applies general missingness factorization and finally establishes recoverability (or not) of the input query by recursively invoking itself on the individual factors. The soundness of the algorithm follows from the soundness of general missingness factorization and the soundness of individual functions. The algorithm modifies the graph via *getLatentGraph* and *getAncestralGraph* functions, and the query using *pruneQuery* and *addVariables* functions. Furthermore, if the query or graph remains unchanged between consecutive iterations, gmf function will return input query itself as the output. This is checked and when it happens the algorithm returns a FAIL message. Finally, the function *querySeenBefore* ensures that a query seen before in the same call stack is not processed again on the same graph.

4.6 Summary and Discussion

In this chapter we exemplified recoverability in cases where a variable and its mechanism are connected by an inducing path. We presented a new factorization scheme that is general and applicable to both probabilistic and causal queries. Using this factorization scheme, we developed a general algorithm to recover conditional probability and interventional distributions.

A pertinent question at this juncture is whether or not the algorithm, *recover*, is complete, and our answer is that we do not know. The hardness of proving non-recoverability is in generalizing the proof over all non-recoverable models, as opposed to proving non-recoverability given a specific model. We devoted our efforts in developing procedures that can, using additional information, recover queries in non-recoverable models. This is covered in the next two chapters.

CHAPTER 5

Overcoming Theoretical Impediments to Recoverability

The most common type of missingness that one encounters in the real world is that of a variable causing its own missingness. In other words, missingness R_X in a variable X is a function of its underlying true value, represented graphically as $X \rightarrow R_x$. Examples of such missingness are: smokers not answering questions about their smoking behavior in insurance applications, longitudinal studies with attrition [Lit95], people with high income not revealing their income and a general reluctance to answer questions about sensitive topics such as religion, sexual habits and abortion. We call missingness problems such as these that are characterized by non-recoverability of joint distribution as *hard-MNAR* problems.

This chapter develops an algorithm for recovering joint distribution in *hard-MNAR* problems by exploiting properties of the data. In the event of non-recoverability despite these new assumptions, the chapter exemplifies computation of informative bounds.

Chapter Outline: In section 5.1, we exemplify recoverability of joint distribution in self-masking models, formalize the recoverability procedure as a theorem, and develop an algorithm based on the theorem for handling *hard-MNAR* problems. In section 5.2 we compute informative upper and lower bounds for queries of interest.

5.1 Restricted Recoverability in Non-Recoverable Models

Recoverability of a target quantity Q, as defined in chapter 3, is a strong criterion in the sense that it requires Q to be consistently estimable *for all* data that the m-graph generates. Therefore, if there exists even one dataset D generated by the graph for which Q cannot be consistently estimated, Q is termed non-recoverable. In this chapter we relax this strong requirement and decide recoverability by examining properties of both m-graph G and manifest distribution P^* as opposed to relying exclusively on the m-graph. In other words, we view recoverability as a property of both data and m-graph. Therefore, it is quite possible that given m-graph G and data distribution P_1^*, query Q may be recoverable, but given the same m-graph G but a different distribution P_2^*, Q may be non-recoverable.

Interestingly, for some m-graphs such as the one discussed in example 13 below, queries can be non-recoverable for all data generated by them.

Example 13. *Consider a missing dataset comprising of a single variable, Income (I), obtained from a population in which the very rich and the very poor were reluctant to reveal their income. Obviously, under these circumstances the true distribution over income, $P(I)$, cannot be computed even given infinitely many samples, for we are neither given nor able to compute the fraction of rich (or poor) who refused to disclose their income.*

However, upon embellishing the m-graph with more variables as illustrated in the ensuing example, queries become recoverable for a large number of datasets.

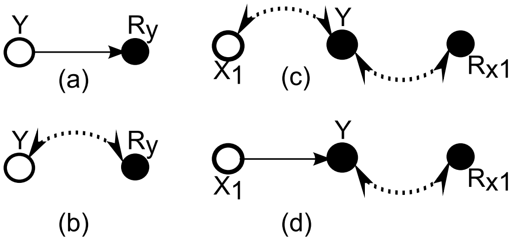


Figure 5.1: m-graphs depicting self masking model in (a) and collider induced models in (b), (c) and (d).

5.1.1 Recoverability by Matrix Inversion

Example 14. *Consider the m-graph $X \to I \to R_I$. The query of interest is $P(X, Y)$. Let X and I be binary variables. Both $P(X)$ and $P(X|I)$ may be recovered using $X \perp\!\!\!\perp R_x$ and $X \perp\!\!\!\perp (R_x, R_i)|I$*

respectively, as shown below:

$$P(X) = P(X|r'_x)) = P(X^* = X|r'_x)$$
$$P(X|I) = P(X|I, r'_x, r'_i) = P(X^* = X|I^* = I, r'_x, r'_i)$$

Expressing $P(X) = \sum_y P(X|I)P(I)$ *in matrix form, we get:*

$$\begin{pmatrix} P(x') \\ P(x) \end{pmatrix} = \begin{pmatrix} P(x'|y') & P(x'|y) \\ P(x|y') & P(x|y) \end{pmatrix} \begin{pmatrix} P(y') \\ P(y) \end{pmatrix}$$

Assuming that the square matrix on R.H.S is invertible, $P(I)$ *can be estimated as:*

$$\begin{pmatrix} P(x'|y') & P(x'|y) \\ P(x|y') & P(x|y) \end{pmatrix}^{-1} \begin{pmatrix} P(x') \\ P(x) \end{pmatrix}$$

Having recovered $P(I)$*, the query* $P(X, I)$ *may be recovered as* $P(X|I)P(I)$*.*

Notice that recoverability is incumbent upon the invertibility of the matrix. If X *and* I *are independent, then matrix won't be invertible and recovery will not be feasible. Therefore to facilitate recovery it is important to choose an* X *that is strongly correlated with* I*. For instance in the example if* I *denotes Income then* X *denoting years of work experience, is a promising candidate.*

The following theorem presents sufficient conditions under which the procedure described in example 14 yields a consistent estimate. $M_{WY} = P(W|I)$ denotes a square matrix with non-negative entries such that entries in each column sum to one. For example, for binary variables W and I, $M_{WY} = \begin{pmatrix} P(w=0|y=0) & P(w=0|y=1) \\ P(w=1|y=0) & P(w=1|y=1) \end{pmatrix}$. For any variable W, $|W|$ denotes the the cardinality of W. Given a set $W = \{W_1, W_2, ...W_n\}$, we define $|W| = \prod_i |W_i|$

Theorem 9 (Sufficiency)**.** *Let G be an m-graph,* $V = V_o \cup V_m$ *and* W *be a set of variables in the super graph of G but not in G such that*

1. $P(W)$ *and* $P(W|V)$ *are recoverable*

2. $|W| = |V|$

Given G and W, $P(V)$ and $P(WV)$, are recoverable if M_{WY} is invertible.

Proof. $P(W) = \sum_V P(W|V)P(V)$. By re-writing this equation in the matrix format and then inverting M_{WV} we recover $P(V)$ as: $P(V) = M_{WV}^{-1}P(W)$.
$P(WV) = P(W|V) * P(V)$ □

Variables in set W that satisfy conditions 1 and 2 in the preceding theorem are referred to as **ancillary variables**. They can be any set of variables that meets the criteria in theorem 9. In fact, they can even include R variables so long as the corresponding partially observed variables are not part of the recovery procedure.

5.1.2 Algorithm for recovering joint distribution in hard-MNAR problems

Theorem 9 relies on the inversion of M_{WV} and hence requires $|W| = |V|$ so that M_{WV} is a square matrix. While cardinality mismatch is an impediment for theorem 9, it is not for recoverability. Algorithm *recoverHardMNAR* describes how to proceed with recovery even when $|W| \neq |V|$. Algorithm 2 handles cardinality mismatch by clustering values of X if $|X| > |W|$ and W if $|X| < |W|$. However, clustering or state space abstraction is known to create dependencies between the neighbors of the clustered variable [WL94, CF13]. For example, in the graph $A \rightarrow C \rightarrow B$, if variable C is clustered then in the post-clustered distribution, $A \perp\!\!\!\perp B|C$, may no longer hold. To account for such dependencies we define the function *getClusteredGraph* that constructs and returns a graph G^c compatible with the new distribution over clustered variables.

Algorithm 2: recoverHardMNAR($\mathbf{W}, \mathbf{X}, \mathbf{P}, \mathbf{G}$)

```
if $|W| == |X|$ then
    return $invert\&recover(\mathbf{W}, \mathbf{X}, \mathbf{P}, \mathbf{G})$
if $|W| > |X|$ then
    Cluster $|W| - |X| + 1$ values of $W$ into a single value $w_+$ to form the clustered variable $W^c$ whose state space comprises of $w_1, w_2, ...w_{|X|-1}, w_+$.
    $G^c \leftarrow getClusteredGraph(W, G)$
    Let $P^c$ be the new data distribution compatible with the clustering process
    return $invert\&recover(\mathbf{W^c}, \mathbf{X}, \mathbf{P^c}, \mathbf{G^c})$
```

```
$G^c \leftarrow getClusteredGraph(X, G)$
Create $|W|$ sized partitions $Parts$ by picking $|W| - 1$ values of $X$ and clustering the rest to form new value $x_+$.
Let $Vector_X$ be a vector indexed by values of $X$.
for all $Parts[i] \in$ Parts do
    $X_i^c \leftarrow Parts[i]$, where $x_+^i$ is the clustered value
    Let $P_i^c$ be the new data distribution compatible with the clustering process
    $recovered \leftarrow invert\&recover(\mathbf{W}, \mathbf{X_i^c}, \mathbf{P_i^c}, \mathbf{G^c})$
    if $recovered == FAIL$ then
        return $FAIL$
    Store all variables in $recovered$ to $Vector_X$ except the one corresponding to $x_+^i$
return $Vector_X$
```

The following function checks to see if theorem 9 can be applied to recover $P(X)$. If so it returns the estimand computed using the theorem, otherwise it returns FAIL.

Function $invert\&recover(\mathbf{W}, \mathbf{X}, \mathbf{P}, \mathbf{G})$

```
if $|W| == |X|$ and ($P(W)$ and $P(W|X)$ are recoverable in $G$) then
    if $M_{WX}$ is invertible then
        Let $E$ be the estimand computed using theorem 9
        return $E$
return FAIL
```

Clustering creates dependencies among neighbors of the variables being clustered. The following function constructs and returns a graph that reflects the dependencies introduced by clustering. The function uses latent projection [VP91, Pea09, SMP15] to ensure that the model contains no conditional independence of the form $X \perp\!\!\!\perp Y|Z$, in which Z contains a clustered variable.

Function $getClusteredGraph(W, X, G)$

```
Let $G_l$ be the latent projection constructed with respect to $W$
Let $E_1$ be the set of edges in $G_l$ that are not in $G$.
Add the edges in $E_1$ to $G$
**return** $G$
```

Consider the m-graph $G: \ W \rightarrow X \rightarrow R_X$. We will detail how algorithm *recoverHardMNAR* recovers $P(X)$ in each of the following cases. When $|W|$ and $|X|$ are equal, recoverability is straightforward as exemplified in example 14. When $|W| > |X|$, we will cluster values of W. For instance let $W = \{w_1, w_2, w_3\}$ and $X = \{x_1, x_2\}$. We shall now cluster W to form $W^c = \{w_1, w_+\}$, where w_+ was created by clustering w_2 and w_3. G^c is the same as G. Using W_c, X and the distribution $P^c(W_c, X, R_X)$, we can now apply theorem 9 to recover $P(X)$. When $|W| < |X|$, X is the variable to be clustered. $G^c: W \rightarrow X \rightarrow R_X \ W \rightarrow R_X$. $P(W|X)$ is not recoverable in G^c and hence theorem 9 is not applicable.

5.1.3 Scope of Theorem 9

Applicability of Theorem 9 to problems in which joint distribution is recoverable Clearly, theorem 9 is not restricted to any specific graph structure and can be applied to all missing data problems that satisfy its conditions, including m-graphs that are not hard-MNAR as exemplified below.

Example 15. *Let G be $W \rightarrow X \rightarrow Y \rightarrow R_x$ and the query of interest be $P(X)$. Although the model is MAR and $P(X)$ can be recovered as $\sum_Y P(X^*|Y, R_x = 0)P(Y)$, it can also be recovered using theorem 9.*

The following example presents a scenario where theorem 9 will be particularly helpful, even when the joint distribution is recoverable in the model.

Example 16. *Let G be $W \rightarrow X \rightarrow Y \rightarrow R_x \ R_y$ and the query of interest be $P(X)$. $P(X)$ can be recovered using theorem 2 as $\sum_Y P(X^*|Y^*, R_x = 0, R_y = 0)P(Y^*|R_y = 0)$. Suppose Y is a variable that is significantly affected by missingness i.e. only in a small number of samples Y is observed. Then the estimand above is likely to not yield good 'quality' estimates in practice. However, the estimand yielded by theorem 9 is independent of Y and can be used to compute $P(X)$.*

Recall that a requirement for recoverability is that the manifest distribution $P(X^*, Y^*, R_x, R_y)$ be strictly positive. Interestingly, even when this requirement is not met, theorem 9 can be used for recovery as long as $P(W, X^*, R_x = 0)$ is strictly positive.

The following theorem states that theorem 9 presents a necessary condition for recoverability in hard-MNAR problems.

Theorem 10 (Necessity)**.** *Given an m-graph G encoding a hard-MNAR problem and ancillary variables W, $P(V)$ is recoverable* ***only if*** *M_{WV} is invertible.*

Proof: See appendix A.3.1

Remark 6. *Given ancillary variables W, theorem 9 presents a* ***necessary and sufficient*** *condition for recoverability of joint distributions in hard-MNAR problems.*

We note that theorem 9 presents only a sufficient criterion for problems that are 'not' hard-MNAR. For example, consider the disconnected m-graph G: W Y R_y, comprising binary variable. While $P(X, Y)$ is recoverable using theorem 3, it cannot be recovered using theorem 9 since M_{WY} will not be invertible in this case.

The results presented in this section are inspired by similar results in epidemiology ([RGL08]), regression analysis ([CRS06]) and causal inference ([Pea12, KP14]). In contrast to [Pea12] that relied on external studies to compute causal effect in the presence of an unmeasured confounder, [KP14] showed how the same could be effected without external studies. In missing data setting we have access to partial information that allows us to compute conditional distributions. This allows us to adapt the procedure in [Pea12] to compute consistent estimates given models in Figures 5.1 and 5.2.

5.2 Computing Informative Bounds

We now deal with the case where no ancillary variables are available and no parametric assumptions such as linearity hold. In situations where data are scarce, it is prudent to salvage the available data to compute an error bound rather than discarding the whole dataset for non-recoverability reasons. Hence we compute bounds for queries of interest. In order to compute tight bounds we partition a given query into factors that are recoverable and non-recoverable and then tighten the bounds by estimating the recoverable factors.

Missingness graphs as a guide for future data collection: Interestingly, this bounding strategy identifies specific variables such that recoverability is guaranteed once data is collected over these variables again. For example, suppose the query of interest is $P(X, Y)$ and it is non-recoverable given the m-graph G. Even if we have the means to re-conduct the survey, doing so and collecting data over X and Y may not always be helpful; quite possibly in any future attempt at collecting data over X and Y, the population would behave in an identical manner as it did before, yielding a similar dataset and identical m-graph G. However by partitioning $P(X, Y)$ into a recoverable part say, $P(Y|X)$ and a non-recoverable factor say, $P(X)$, we need to collect data *only* over X for recovering $P(X, Y)$. In this way, we can harness information embedded in the bounds themselves to identify the set of variables that would further tighten the bounds or even facilitate recovery. Furthermore, in many cases we might be able to use external data sources such as census bureau (for age) and public salary records (for income).

We will now illustrate the subtleties involved in the bounding process.

5.2.1 Bounding Self-masking Models

Example 17. *Let the relation of interest be $P(Y = y)$ and the missingness process be depicted in the m-graph in Figure 5.1 (a). Since $P(Y)$ is non-recoverable, the best we can say is that either all missing values correspond to $Y = 0$ or that they correspond to $Y = y$. Therefore, we can express $P(Y)$ as the sum, $P(Y, r_y) + P(Y, r'_y)$ and use $0 \leq P(Y, r_y) \leq P(r_y)$ to bound $P(Y = y)$ as:*

$$P(r'_y, Y = y) \leq P(Y = y) \leq P(r'_y, Y = y) + P(r_y)$$

When $P(r_y)$ is negligible, we observe that the estimation error is also negligible.

Definition 9 (Trivial bounds)**.** *The trivial bounds of a distribution $P(x_m, x_o|y)$, where $X_m \subseteq V_m$, $(X_o, Y) \subseteq V_o$, $X_m \cap Y = \emptyset$ are $P(x_o, x_m, r'_{x_m}|y) \leq P(x_m, x_o|y) \leq P(x_o, x_m, r'_{x_m}|y) + P(x_o, r_{x_m}|y)$.*

The bounds are computed by considering the extreme cases. The lower bound is computed under the assumption that no missing sample assumes the value $X_m = x_m$ and the upper bound is computed under the assumption that all missing samples assume the value $X_m = x_m$. Although a trivial bound is a tight bound in the preceding example, the following example demonstrates that given other m-graphs we can compute better bounds by decomposing a non-recoverable query Q into recoverable and non-recoverable parts.

Example 18. *Let the relation of interest be $P(X = x, Y = y)$ and the missingness process be depicted in the m-graph in Figure 6.1 (b). Since $P(XY)$ is non-recoverable, it is possible that either no missing values correspond to $X = x, Y = y$ or that all missing values correspond to $X = x, Y = y$ i.e. we can trivially bound $P(X = x, Y = y)$ as:*

$$\begin{aligned} P(r'_y, r'_x, x, y) \leq P(x, y) \leq &P(r'_y, r'_x, x, y) + P(r_y, r'_x, x) + \\ &P(r'_y, r_x, y) + P(r_y, r_x) \end{aligned}$$

However, in this case we know that $P(X|Y)$ may be recovered as $P(X^*|Y^*, R_y = 0, R_x = 0)$ since $X \perp\!\!\!\perp (R_x, R_y)|Y$ ([MPT13]). Using $P(X, Y) = P(X|Y)P(Y)$ and the the trivial bound for $P(Y)$, we can compute the following tighter bound for $P(X, Y)$:

$$\begin{aligned} P(X|Y)P(r'_y, Y) \leq P(X, Y) \leq P(X|Y)(&P(r'_y, Y) \\ &+P(r_y)) \end{aligned}$$

The bounds derived in example 18 are called **non-trivial bounds**. In contrast to the trivial bounds described above they guarantee to provide consistent estimates even when missingness remains non negligible on portion of the database. We can therefore characterize each non-trivial bound by that portion of the database in which missingness must approach zero for inconsistency to vanish. Moreover, non-trivial bounds help us quantify the rate of convergence and, hence, the relative importance of minimizing missingness in different variables, or combinations of variables. For instance, in the preceding example it would be a waste of resources to minimize missingness on X, whereas minimizing missingness of Y is crucial.

5.2.2 Bounding Collider-induced Models

In the preceding example we bounded a marginal distribution. In this case the procedure is similar except that we will bound a conditional distribution.

Example 19. *Let the relation of interest be $P(X = x, Y = y)$ and the missingness process be depicted in the m-graph in Figure 5.1 (c). We know that $P(Y)$ is recoverable since Y is fully observed. Using $P(X,Y) = P(X|Y)P(Y)$ and the the trivial bound for $P(X|Y)$, we can compute the following tighter bound for $P(X = x, Y = y)$:*

$$P(X, r'_x = 0|Y)P(Y) \leq P(X,Y) \leq P(Y)(P(r'_x, X|Y) \\ +P(r_y|Y))$$

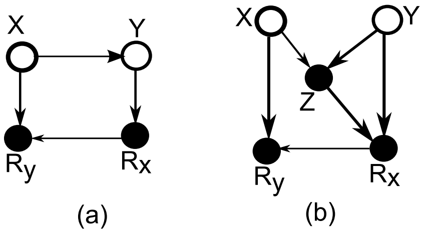


Figure 5.2: m-graphs depicting entangled models

5.2.3 Bounding Entangled Models

In this case, we capitalize on the recoverability of specific events to recover sub-queries in the decomposition.

Example 20. *Let the relation of interest be $P(X = x, Y = y)$ and the missingness process be depicted in the m-graph in Figure 5.2 (a). Although $X \not\perp\!\!\!\perp (R_y, R_x)|Y$, the conditional distribution*

$P(X|Y)$ may be recovered as shown below:

$$
\begin{aligned}
P(X,Y,r'_x,R_y) &= P(Y|X,r'_x,R_y)P(X,r'_x,R_y)\\
&= P(Y|X,r'_x,r'_y)P(X,r'_x,R_y)\\
&\textit{(since } Y \perp\!\!\!\perp R_y|(X,R_x)\textit{)}\\
&= P(Y^*|X^*,r'_x,r'_y)P(X^*,r'_x,R_y) \qquad (5.1)\\
P(X|Y) = P(X|Y,r'_x) &= \frac{P(X,Y,r'_x)}{P(Y,r'_x)}\\
&= \frac{\sum_{R_y} P(X,Y,r'_x,R_y)}{\sum_{R_y,X} P(X,Y,r'_x,R_y)}\\
&\textit{(By eq 5.1, } P(X,Y,r'_x,R_y) \textit{ is recoverable)}
\end{aligned}
$$

Since $P(X|Y)$ is recoverable, the bound here is the same as the bound in example 18. Given m-graph in Figure 5.2 (b) $P(X,Y,Z)$ may be bounded in a similar manner by decomposing it as $P(X|YZ)P(YZ)$. Here $P(X|YZ)$ is recoverable since the event $P(X,Y,Z,R_y,r'_x)$ may be recovered as $P(Y^*|X^*,Z,r'_y,r'_x)P(X^*,Z,R_y,r'_x)$. Finally we apply the trivial bound on $P(Y,Z)$.

5.3 Summary

We presented two strategies of overcoming impediments to estimation. The first, based on matrix inversion is applicable to variables with discrete, finite states, applicable to a broad set of problems and is independent of parametric assumptions. For problems that cannot be handled by the preceding strategy, we exemplified methods to compute bounds for the target queries. We demonstrated that by decomposing a query into sub-queries, some of which are recoverable, we can take advantage of the recoverable part to produce tighter bounds on the target query.

CHAPTER 6

Linear Models for Missing Data

Linear Structural Equation Modeling has been widely used for estimating parameters of interest under missing data conditions ([All03, Gra03, End06, SBC10, UB03]). Almost all existing SEM techniques for missing data employ maximum likelihood or multiple imputation based approaches for computing parameter estimates. Additionally, these techniques require that the missing data mechanism be ignorable i.e. the data be generated by a MCAR or MAR process. Exceptions include [Pea13] in which path-analytic techniques were used to recover covariance matrix from MNAR data. In this chapter we demonstrate that aided by proxy variables, causal effects and sometimes the entire covariance matrix can be recovered given models depicting hard-MNAR problems i.e. models in which neither the joint distribution nor identifiable causal effects are recoverable non-parametrically.

Chapter Outline: In section 6.1 we discuss SEMs, and present definitions of covariance, variance and regression coefficients. Quasi-linear missingness model is defined and an example of recoverability in MAR under linear assumption is presented in section 6.2. Section 6.3 presents methods for recovering variance and path coefficients in hard-MNAR problems.

6.1 Preliminaries: Structural Equation Models, Variance and Covariance

Before formally defining the linear missingness model, we shall briefly review Structural Equation Models. For a detailed discussion see [Pea09] (chapter-5) and [Bri04].

6.1.1 Structural Equation Models

A structural equation model (SEM) is a system of equations defined over a set of variables, such that each variable appears on the left hand side of at most one equation. Each equation describes the dependence of one variable in terms of the others and contains an error term to account for the influence of unobserved factors. Example: $X = \epsilon_x$ and $Y = \alpha X + \epsilon_y$. As in [Pea13], we interpret structural equations as an assignment process whose directionality is captured by a path diagram (see Figure 6.1). All substantive variables ($\{V_m \cup V_o \cup U\}$), and error terms are assumed to be drawn from a Gaussian distribution.

6.1.2 Covariance, Variance and Regression Coefficients

The following is a list of basic formulae used in this paper.

For two variables X and Y covariance and variance are defined as follows:

$$cov(X, Y) = E(XY) - E(X)E(Y) \tag{6.1}$$

$$var(X) = E(X^2) - E(X)^2 \tag{6.2}$$

The regression coefficient denoted by β_{yx}, representing the rate of change of Y as a function of X is given by,

$$\beta_{yx} = \frac{cov(X, Y)}{var(X)} = \frac{d}{dx}E(Y|X = x) \tag{6.3}$$

The partial regression coefficient denoted by $\beta_{yx.z}$, representing the rate of change of Y as a function of X, computed from cases in which $Z = z$ is given by,

$$\beta_{yx.z} = \frac{d}{dx}E(Y|X = x, Z = z) \tag{6.4}$$

6.2 Quasi-linear Missingness Model

The causal missingness mechanism is a binary variable and as such the function generating it cannot be linear. Therefore, we define the quasi-linear model below to capture the missingness process.

Definition 10. *A Quasi-linear Missingness Model is a Structural Equation Model such that:*
1. every substantive variable X is a linear function of its causes ,Y, and a random error term ϵ_x

$$c1.\, X_1 = \epsilon_{x_1}$$

$$c2.\, Y = \alpha X_1 + \epsilon_y$$

$$c3.\, X_2 = \gamma Y + \delta X_1 + \epsilon_{x_2}$$

$$c4.\, R_y = f(Y, \epsilon_{r_y})$$

$$c5.\, Y^* = (1 - R_y)Y + mR_y$$

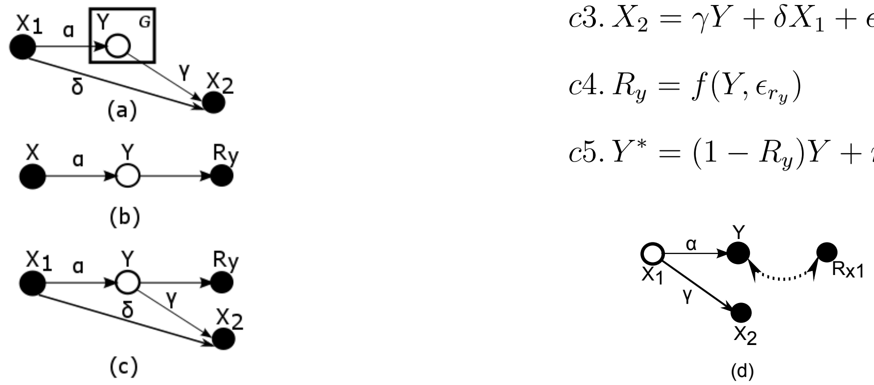


Figure 6.1: (a), (b), (c) and (d) are quasi-linear missingness models and equations c1, c2, c3, c4 and c5 constitute the SEM corresponding to m-graph (c)

$$X = \alpha_1 Y_1 + \alpha_2 Y_2 + ... + \alpha_n Y_n + \epsilon_x$$

The coefficient α's are called path coefficients or structural parameters.
2. For every $R_x \in R$, $R_x = f(Z, \epsilon_{R_x})$ where Z is the set of causes and f is a non-linear function. No R variable is a parent of any substantive variable.
3. Every proxy variable X^* is generated by the non-linear function: $X^* = (1 - R_x)X + mR_x$

We will now exemplify recoverability of covariance matrix in MAR and MCAR cases.

Example 21. *Consider the problem of estimating the covariance matrix given the MAR model of Figure 6.2 (b). Since Y is fully observed, $var(Y)$ is trivially recoverable. In order to recover $cov(X, Y)$, we will first recover β_{XY}, the regression coefficient of Y on X.*

$$\beta_{XY} = \frac{d}{dy} E(X|y)$$

Since $X \perp\!\!\!\perp R_x|Y$ we have the license to compute β_{XY} (using OLS) from samples in which X is observed. $cov(X, Y)$ can now be recovered as:

$$cov(X, Y) = var(Y)\beta_{XY}$$

Finally, to recover $var(X)$ we require a procedure that consistently estimates $E(X)$. However, the recoverability of $E(X)$ is far from obvious because the model does not encode the conditional independence: $X \perp\!\!\!\perp R_x$, thereby leading one to presume that $E(X)$ cannot be recovered until one conditions on Y. However, this turns out not to be the case as shown below:

$$E(X) = E(E(X|Y))$$

Since $X \perp\!\!\!\perp R_X|Y$, $E(X|Y) = \beta_{XY}Y + c$ can be computed by linear regression from samples in which X is observed. Therefore,

$$\begin{aligned} E(X) &= E(\beta_{XY}Y + c) \\ &= \beta_{XY}E(Y) + E(c) \end{aligned}$$

Let μ_Y denote the mean of Y computed from all samples. Then,

$$E(X) = \beta_{XY}\mu_Y + c$$

To estimate $var(X)$, we proceed in a similar manner using the formula: $var(X) = E(Var(X|Y)) + Var(E(X|Y))$.

An upshot of the estimation procedure in example 21 is the estimability of the causal effect of X on Y, β_{YX} as shown below:

$$\beta_{YX} = \frac{cov(X, Y)}{var(X)}$$

We observe that although a consistent estimate of β_{YX} cannot be computed directly from fully observed data (i.e. $P(X, Y, R_x = 0)$) , it can be recovered by a procedure in which each factor in the estimand is independently estimated from a subset of the available dataset.

Estimation methods applicable to MAR are applicable to MCAR as well because by the weak union axiom of graphoids, Missing Completely at Random (MCAR: $(V_m, V_o) \perp\!\!\!\perp R$) implies Missing At Random (MAR: $V_m \perp\!\!\!\perp R|V_o$). Therefore, it implicitly follows that queries (such as covariance and variance) that are recoverable given MAR datasets are recoverable given MCAR datasets as well. In a manner similar to the estimation procedure in example 21, the covariance matrix can be recovered given the MNAR problem in Figure 6.2(c) as well.

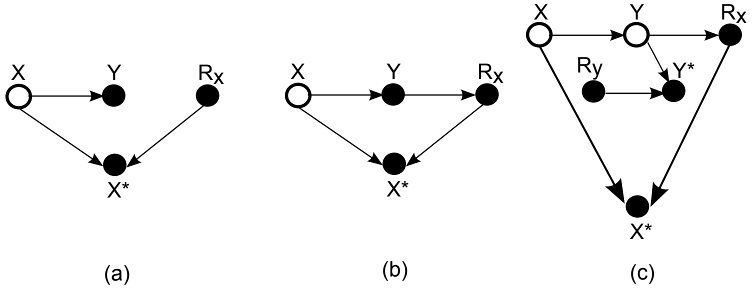


Figure 6.2: Examples of (a) MCAR, (b) MAR and (c) MNAR missing data generation processes

6.2.1 Estimating Mean of Partially Observed Variables

Theorem 11. *Let $X \in V_m$. $E(X)$ is recoverable if there exists $Z = \{Z_1, Z_2, ...Z_n\}$ such that $X \perp\!\!\!\perp R_x R_z | Z$ and $E(Z_i)$ is recoverable for all $Z_i \in Z$.*

Proof.

$$E(X) = E(E(X|Z)) = \sum_z P(z)E(X|Z)$$

Performing linear regression of Z on X will return the intercept c and coefficients α_i's. More importantly, regression is performed using only those samples in which all variables in $Z \cup \{X\}$ are observed (i.e. not missing).

$$\begin{aligned} E(X) &= \sum_z P(z)(c + \sum_{i=1}^{n}(\alpha_i Z_i)) = c\sum_z P(z) + \sum_z P(z)\sum_{i=1}^{n}(\alpha_i Z_i) \\ &= c + \sum_{i=1}^{n}\alpha_i E(Z_i) \qquad (6.5) \end{aligned}$$

□

6.3 Recovering Path Coefficients, Variance and Covariance in hard-MNAR problems

Variance and covariance are recoverable in an uncomplicated manner in many instances such as when the concerned variables are fully observed or missing completely at random. In this subsection we develop methods to recover variance and covariance for cases where traditional methods fail.

In the following lemmata we re-phrase and state known results for estimating covariance ([Pea09, Bri04, Pea13, Wri21]):

Lemma 4. ***Single path:*** *Let G be an m-graph and p be an unblocked path between X_1 and X_n with intermediate nodes: $X_2, X_3 ... X_{n-1}$. Let X_j be the ancestor of all nodes on p.*

$$\boxed{cov(X_1, X_n) = var(X_j) * \prod_{i=1, i \neq j}^{n} \alpha_i}$$

where $\prod\limits_{i=1, i \neq j}^{n} \alpha_i$ is the product of all causal parameters on path p.

Lemma 5. ***Multiple paths:*** *Let G be an m-graph with k unblocked paths $p_1, ..p_k$ between X_1 and X_n. Let X_j^m be the ancestor of all nodes on path p_m.*

$$\boxed{cov(X_1, X_n) = \sum_{m=1}^{k} var(X_j^m) * \prod_{i=1, i \neq j}^{n} \alpha_i^m}$$

where $\prod\limits_{i=1, i \neq j}^{n} \alpha_i^m$ is the product of all causal parameters on path p_m.

For example, in figure 6.1 (c), there exist two paths, $X_1 \rightarrow Y \rightarrow X_2$ and $X_1 \rightarrow X_2$, between X_1 and X_2. Therefore using lemma 5, $cov(X_1, X_2) = \alpha\gamma + \delta$

Lemma 6. ***Computing path coefficients:*** *Let G be an m-graph with k unblocked paths $p_1, ..p_k$ between X_1 and X_n. If $cov(X_1, X_n)$ is known and all path coefficients are known, except one, say α_a^b, then α_a^b is recoverable as:*

$$\boxed{\frac{cov(X_1, X_n) - \sum\limits_{m=1, m \neq b}^{k} var(X_j^m) * \prod_{i=1, i \neq j}^{n} \alpha_i^m}{var(X_j^b)} - \prod_{i=1, i \neq j, i \neq a}^{n} \alpha_i^m}$$

where $\prod\limits_{i=1, i \neq j}^{n} \alpha_i^m$ is the product of all causal parameters on path p_m.

Proof follows from lemma 5

In figure 6.1 (a) the rectangular box is a shorthand for indicating that Y is part of an m-graph G such that no other variable in G, except Y is a neighbor of X_1 and X_2.

Theorem 12. *Let $Y \in V_m$. $Var(Y)$ is recoverable if there exists X_1, X_2 as shown in Figure 6.1 (a) such that*

1. *$var(X_1)$ and $cov(X_1, X_2)$ are recoverable*
2. *$\gamma \neq 0$, $\alpha \neq 0$*

Proof. 1. Recovering δ

$$\delta = \beta_{X_2 X_1 . Y} \text{ (by single door criteria [BP02])}$$

Since $X_2 \perp\!\!\!\perp R_y | (Y, X_1)$, $\beta_{X_2 X_1 . Y}$ can be recovered by linear regression from samples in which Y is observed.

2. Recovering γ

$$\gamma = \beta_{X_2 Y . X_1} \text{(using back door criteria ([Pea09]))}$$

Since $X_2 \perp\!\!\!\perp R_y | Y, X_1$, $\beta_{X_2 Y . X_1}$ can be recovered by linear regression from samples in which Y is observed.

3. Recovering α

$$\begin{aligned} cov(X_2, X_1) &= \gamma cov(X_1, Y) + \delta var(X_1) \\ &= \gamma \alpha var(X_1) + \delta var(X_1) \\ \text{Therefore, } \alpha &= \frac{1}{\gamma}\Big(\frac{cov(X_2, X_1)}{var(X_1)} - \delta\Big) \end{aligned} \tag{6.6}$$

The above result follows immediately from lemma 6.

4. Recovering $var(Y)$

$$var(Y) = \frac{cov(X_1, Y)}{\beta_{X_1 Y}}$$

Since $X_1 \perp\!\!\!\perp R_y | Y$, regression coefficient $\beta_{X_1 Y}$ is recoverable by linear regression using samples in which Y is observed.

$$\begin{aligned} &= \frac{cov(X_1, Y)}{\beta_{X_1 Y}} \\ &= \frac{\alpha * var(X_1)}{\beta_{X_1 Y}} \end{aligned} \tag{6.7}$$

5. Recovering $E(Y)$

   Perform linear regression with Y as regressor and X_1 as regressand, using samples in which Y is observed. Let c be the intercept and β_{X_1Y} be the coefficient produced as output. It follows from equation 6.5 that,

$$E(X_1) = c + \beta_{XY}E(Y)$$

$$\text{Therefore, } E(Y) = \frac{E(X_1) - c}{\beta_{X_1,Y}}$$

□

Scope of the Results: Theorem 12 does not impose constraints or restrictions on the structure of the m-graph G which contains the node Y. So it can be used to compute the variance of Y in all m-graphs so long as its conditions are met.

Using the results derived thus far, we exemplify below recoverability of path coefficients in collider-induced and entangled models.

6.3.1 Recoverability in Collider-Induced Models

Example 22. *Given the model in Figure 6.1 (d), covariance matrix is recoverable.*

1. *Recovering $var(X_1)$*

   Since $X_1 \perp\!\!\!\perp R_{x_1}$, $var(X_1)$ is recoverable and may be estimated from samples in which X_1 is fully observed.

2. *Recovering γ*

   Using $X_2 \perp\!\!\!\perp R_{x_1}|X_1$, we can recover γ as $\beta_{X_2X_1}$ by linear regression using samples in which X_1 is fully observed.

3. *Recovering α*

$$cov(X_2, Y) = \alpha cov(X_1, X_2) = \alpha\gamma var(X_1)$$

$$\textit{Therefore, } \alpha = \frac{cov(X_2, Y)}{\gamma\, var(X_1)}$$

   The above result follows immediately from lemma 6.

Since the path coefficients and variances are recoverable, the covariance matrix is recoverable by lemma 5.

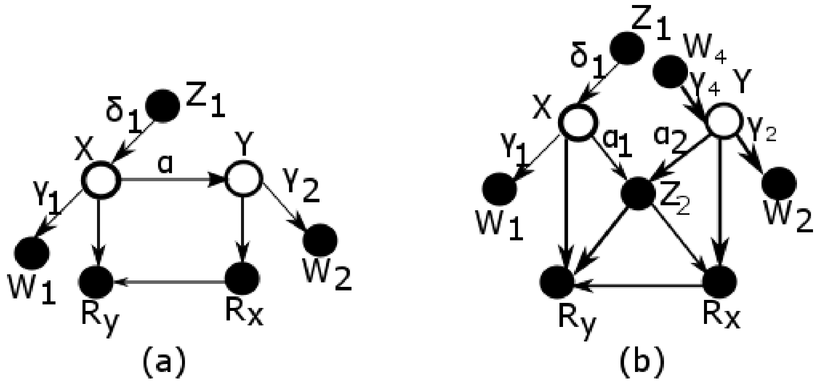


Figure 6.3: Quasi-linear models in which causal effect of (a) X on Y is recoverable (b) X on Z and Y on Z are recoverable

6.3.2 Recoverability in Entangled Models

We will now exemplify that in the models in Figure 6.3 (a), direct effect of X on Y is recoverable and in Figure 6.3 (b), direct effect of X on Z and Y on Z are recoverable.

Example 23. *Given the entangled models in Figure 6.3 (a) and (b), causal effects are recoverable. In both entangled models γ_1, γ_2 and δ_1 can be recovered as shown below:*

1. $\gamma_1 = \beta_{W_1 X}$

Since $W_1 \perp\!\!\!\perp R_X | X$, regression coefficient $\beta_{W_1 X}$ can be recovered by linear regression from samples in which X is observed.

2. $\gamma_2 = \beta_{W_2 Y}$

Since $W_2 \perp\!\!\!\perp R_Y | Y$, regression coefficient $\beta_{W_2 Y}$ can be recovered by linear regression from samples in which Y is observed.

3. Since $cov(W_1, Z_1)$, γ_1 and $var(Z_1)$ are known, δ_1 can be recovered using lemma 6 as $\delta_1 = \frac{cov(W_1,Z_1)}{\gamma_1 var(Z_1)}$

Recovering α in Figure 6.3 (a): *α may be recovered using lemma 6 as,*

$$\alpha = \frac{cov(Z_1,W_2)}{\delta_1\gamma_2 var(Z_1)}$$

Recovering α_1 ***and*** α_2 ***in Figure 6.3 (b)****: Using lemma 6,* α_2 *may be recovered as:*

$$\alpha_2 = \frac{cov(Z_2, W_4)}{var(W_4)\gamma_4}$$

γ_4 *is recoverable using lemma 6 as:* $\gamma_4 = \frac{cov(W_4,W_2)}{\gamma_2 var(W_4)}$*. Therefore,*

$$\alpha_2 = \frac{cov(Z_2, W_4)}{var(W_4)} \frac{\gamma_2 var(W_4)}{cov(W_4, W_2)}$$

Using lemma 6, α_1 *may be recovered as:*

$$\alpha_1 = \frac{cov(Z_1,Z_2)}{\delta_1 var(Z_1)}$$

Furthermore, $var(X)$ *and* $var(Y)$ *are recoverable in Figure 6.3 (b) by theorem 12. Since the path coefficients and variances are recoverable, it follows from lemma 5 that the covariance matrix is recoverable.*

6.4 Summary

In this chapter we showed that linearity assumptions can aid recoverability in missingness problems in which joint distribution is not recoverable and variables are continuous. We defined quasi-linear models for missingness and derived conditions under which parameters can be recovered in hard-MNAR problems. Specifically, we showed how the full covariance matrix can be recovered using a sequential procedure dictated by the graph structure.

CHAPTER 7

Testability under Missingness

Researchers are typically uncertain about the model that accounts for loss of data while at the same time procedures for recovering information from missing data rely on such models. These two facts motivate us to address the question of whether one can submit a given model to a test of compatibility with the data available, which of course is corrupted by missingness. Specifically, we ask whether it is possible to detect misspecifications of the missingness model, we demonstrate this possibility, and identify conditions that permit such detection.

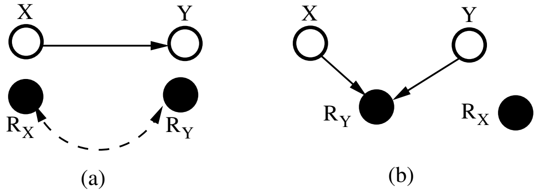


Figure 7.1: m-graphs that yield different estimands for the query $P(X|Y)$

Motivating Example: The following example will demonstrate the sensitivity of recoverability to the structure of the graph. Let G_1 (Figure 7.1(a)) and G_2 (Figure 7.1(b)) be the graphs hypothesized by the researcher for a given manifest distribution $P(X^*, Y^*, R_x, R_y)$. Let $P(X|Y)$ be the query to be recovered. G_1 embeds, $X \perp\!\!\!\perp R_x, R_y|Y)$. Hence, $P(X|Y) = P(X|Y, R_x = 0, R_y = 0)$. On applying Equation-(1) we get,

$$P(X|Y) = P(X^*|Y^*, R_x = 0, R_y = 0).$$

On the other hand, G_2 embeds the CI: $X \perp\!\!\!\perp Y$. Therefore, $P(X|Y) = P(X)$. Furthermore, G_2 also embeds the CI: $X \perp\!\!\!\perp R_x$. Therefore, $P(X) = P(X|R_x = 0)$. On applying Equation-(1) we get,

$$P(X|Y) = P(X^*|R_x = 0)$$

We observe that G_1 and G_2 dictate different estimands which yield different results depending on the missingness process that each portrays. Therefore it is imperative to test whether the manifest distribution and hypothesized model are compatible.

In other words, some properties of P (called "queries") that are recoverable in one graph are not recoverable in another. Moreover, this sensitivity persists even when the two graphs are statistically indistinguishable and the natural question to ask is whether the structure of the m-graph lends itself to statistical tests, given that we are not in possession of the underlying distribution but a distortion thereof in the form of a dataset with missing values. We will show that such tests are indeed available albeit weaker than misspecification tests under complete data.

Chapter outline: Section 7.1 defines testability of CIs portrayed by the m-graph and develops sufficient conditions under which a specific CI is testable given missing data. In Section 7.2 we call attention to an impediment which prevents testability of certain conditional independencies even when the distribution that carries these CIs is fully recoverable. We then present sufficient conditions for non-testability of CIs. Section 7.3 deals with testability of CIs comprising of substantive variables and presents sufficient conditions for such dependence to exist. In Section 7.4 we apply these theoretical results to classes of models which have been analysed in traditional missing data literature and show that (extending the results of [PTP06]) a large class of models traditionally thought of as non-testable are in fact testable. Finally, we use the results developed so far to show that model sensitivity persists in many models typically categorized as MNAR.

7.1 Testability of CI (d-separations) in m-graphs

In this paper we will limit our discussion to testable implications in the form of conditional independence claims entailed by the model. In Figure 7.2 for example, the model claims $X \perp\!\!\!\perp Y|Z$, $Z \perp\!\!\!\perp R_z|(X,Y)$ and $(X,Y,Z,R_z) \perp\!\!\!\perp R_x$. Such claims constitute the totality of testable implications if the underlying model is Markovian i.e. recursive and with independent error terms ([Pea09]). For constraints induced by latent variables, see [TP02b] and [SP08].

Definition 11 (Testable d-separation). *Let $X \cup Y \cup Z \subseteq V_o \cup V_m \cup R$ and $X \cap Y \cap Z = \emptyset$. $X \perp\!\!\!\perp Y|Z$ is testable if there exists a dataset D governed by a distribution $P(V_o, V^*, R)$ such that*

$X \perp\!\!\!\perp Y|Z$ is refuted in all underlying distributions $P(V_o, V_m, R)$ compatible with the distribution $P(V_o, V^*, R)$.

If X and Y are singletons, $X \perp\!\!\!\perp Y|Z$ is termed as **singleton d-separation** and if not, $X \perp\!\!\!\perp Y|Z$ is termed as **compound d-separation**. Let us look at examples of singleton and compound d-separations that are testable.

Example 24. *Let $X \in V_o$ and $Y \in V_m$. $X \perp\!\!\!\perp R_y$ is testable since X and R_y are fully observed variables and we can always find a dataset that refutes $P(X|R_y = 0) = P(X|R_y = 1)$. Similarly when $\{X, Y\} \subseteq V_m$, $R_x \perp\!\!\!\perp R_y$ and $X \perp\!\!\!\perp Y|(R_x = 0, R_y = 0)$ are testable. $R_x \perp\!\!\!\perp R_y$ is testable since R_x and R_y are fully observed variables. $X \perp\!\!\!\perp Y|(R_x = 0, R_y = 0)$ is testable since given $R_x = 0$ and $R_y = 0$ we can apply Equation 2.1 and equivalently write the CI as $X^* \perp\!\!\!\perp Y^*|(R_x = 0, R_y = 0)$ i.e. CI can be expressed equivalently in terms of observed variables and hence it can be refuted.*

Example 25. *Following are two examples of compound d-separations that are testable.*

a. CI: $(X, R_x) \perp\!\!\!\perp (Y, R_y)|(Z, R_z)$ *implies* $P(X^*, R_x^{'}|Z^*, R_z^{'}) = P(X^*, R_x^{'}|Y^*, R_y^{'}, Z^*, R_z^{'})$

b. CI: $(X, R_x) \perp\!\!\!\perp (R_w, R_y)|Y$ *implies* $P(X^*, R_x^{'}|Y^*, R_y^{'}, R_w) = P(X^*, R_x^{'}|Y^*, R_y^{'}, R_w^{'})$

Since both CIs imply CIs that can be expressed in terms of observed variables, the CIs can be refuted. Hence they are testable.

We would like to remark that there exist non-testable CI claims and they are discussed in Section 7.2, Example 27. From definition-11, we conclude that a d-separation is termed testable when it has at least one implication that is testable. Example:26 demonstrates that, in some cases, it might be necessary to examine all implications of a compound d-separation before labeling it as testable.

Example 26. *Consider the d-separation S : $(X, R_y, R_{z1}) \perp\!\!\!\perp (Y, R_x, R_{z2})|(Z_1, Z_2)$. This d-separation translates to $\frac{P(X,R_y^{'},R_{z1}^{'},Y=0,R_x^{'},R_{z2}^{'})}{P(Y=0,R_x^{'},R_{z2}^{'},Z_1,Z_2)} = \frac{P(X,R_y^{'},R_{z1}^{'},Y=1,R_x^{'},R_{z2}^{'})}{P(Y=1,R_x^{'},R_{z2}^{'},Z_1,Z_2)}$. Observe that the denominators cannot be directly expressed in terms of observed variables. To affirm testability of S, we have to examine its implications until we find an implication that is testable.*
For example, S' : $X \perp\!\!\!\perp Y|(Z_1, Z_2, R_y, R_{z1}, R_x, R_{z2})$, obtained by applying weak union graphoid

axiom to S is testable since it translates into $\frac{P(P(X,R_y^{'},R_{z1}^{'},Y=0,R_x^{'},R_{z2}^{'}))}{P(R_y^{'},R_{z1}^{'},Y=0,R_x^{'},R_{z2}^{'})} = \frac{P(P(X,R_y^{'},R_{z1}^{'},Y=1,R_x^{'},R_{z2}^{'}))}{P(R_y^{'},R_{z1}^{'},Y=1,R_x^{'},R_{z2}^{'})}$. *Since S' is testable we can conclude that S is testable.*

Clearly enumerating and testing the set of all implied d-separations is hard since the number of implications is exponential in the sizes of sets X and Y. The next subsection provides a rule to circumvent this enumeration for certain types of d-separations.

7.1.1 Directly testable d-separations

Testability of certain d-separations (such as the compound d-separations in Example:25) can be affirmed in one shot i.e. without explicitly examining all their implications. In other words, testability can be certified by looking at the placement of a mechanism R_X relative to its partially observed variable X in the d-separation statement. We call such d-separations **directly testable**. The following is a syntactic criterion for determining direct testability of d-separations.

Theorem 13. *Let $X, Y, Z \subset V_o \cup V_m \cup \mathbb{R}$ and $X \cap Y \cap Z = \emptyset$. The conditional independence statement S: $X \perp\!\!\!\perp Y|Z$ is directly testable if all the following conditions hold:*

1. $Y \nsubseteq (R_{X_m} \cup R_{Z_m})$

   In words, Y should contain at least one element that is not in $R_{X_m} \cup R_{Z_m}$.

2. $R_{X_m} \subseteq X \cup Y \cup Z$

   In words, the missingness mechanisms of all partially observed variables in X are contained in $X \cup Y \cup Z$.

3. $R_{Z_m} \cup R_{Y_m} \subseteq Z \cup Y$

   In words, the missingness mechanisms of all partially observed variables in Y and Z are contained in $Y \cup Z$.

Proof. Let $Y_1 \in R \cup V_o \cup V_m$ be an element in Y such that condition (1) is satisfied. $X \perp\!\!\!\perp Y|Z$ implies:

$\frac{P(X,Y_1=0,Y-Y_1,Z)}{P(Y_1=0,Y-Y_1,Z)} = \frac{P(X,Y_1=1,Y-Y_1,Z)}{P(Y_1=1,Y-Y_1,Z)} \qquad (a)$

From conditions (2) and (3), we know that the terms in the numerator of both fractions contain R_{X_m}, R_{Y_m} and R_{Z_m}. Similarly, from condition (3), we know that the terms in the denominator of both fractions contain R_{Y_m} and R_{Z_m}. Consider the event where all R variables in $R_{X_m} \cup R_{Y_m} \cup R_{Z_m}$ are equal to zero. We can apply Equation 2.1 and express the numerators and denominators of equation-(a) in terms of observed variables, thereby making the claim testable. □

7.1.2 Graphical Criteria for Testability

The criterion for detecting testable implications reads as follows: *A d-separation condition displayed in the graph is testable if the R variables associated with all the partially observed variables in it are either present in the separator set or can be added to the separator without spoiling the separation.* Formally, we can state this criterion using three syntactic rules. General criterion for identifying untestable conditional independence in the graph are the following ([MP14]):

$$X \perp\!\!\!\perp Y | Z, R_x, R_y, R_z \tag{7.1}$$

$$X \perp\!\!\!\perp R_y | Z, R_x, R_z \tag{7.2}$$

$$R_x \perp\!\!\!\perp R_y | Z, R_z \tag{7.3}$$

In words, any d-separation that can be expressed in the format stated above is testable. It is understood that, if X or Y or Z are fully observed, the corresponding R variables may be removed from the conditioning set. Clearly, any conditional independence comprised exclusively of fully observed variables is testable. To search for such refutable claims, one needs to only examine the missing edges in the graph and check whether any of its associated set of separators satisfy the syntactic format above.

7.1.2.1 Tests Corresponding to the Conditions 7.1, 7.2 and 7.3

When a test satisfies a criterion above, it imposes a condition on the observed data. For example, a specific instance of the claim $X \perp\!\!\!\perp Y | Z, R_x, R_y, R_z$, when $R_x = 0, R_y = 0, R_z = 0$ gives $X \perp\!\!\!\perp Y | Z, R_x = 0, R_y = 0, R_z = 0$. This translates to the equation,

$$P(X|Z, R_x = 0, R_y = 0, R_z = 0) = P(X|Y, Z, R_x = 0, R_y = 0, R_z = 0)$$

The above can be rewritten using equation 2.1 as:

$$P(X^*|Z^*, R_x = 0, R_y = 0, R_z = 0) = P(X^*|Y^*, Z^*, R_x = 0, R_y = 0, R_z = 0)$$

This equation exclusively comprises of observed quantities and can be directly tested given the input distribution: $P(X^*, Y^*, Z^*, R_x, R_y, R_z)$. In a similar manner we can devise tests for the second and third criteria (statements 7.2 and 7.3).

The tests corresponding to the three criteria are:

- $P(X^*|Z^*, R_x = 0, R_y = 0, R_z = 0) = P(X^*|Y^*, Z^*, R_x = 0, R_y = 0, R_z = 0)$,
- $P(X^*|Z^*, R_x = 0, R_z = 0) = P(X^*|R_y, Z^*, R_x = 0, R_z = 0)$
- $P(R_x|Z^*, R_z = 0) = P(R_x|R_y, Z^*, R_z = 0)$

So far we have discussed testable CI statements. In the following section we shall discuss an impediment to testability when data are afflicted by missingness.

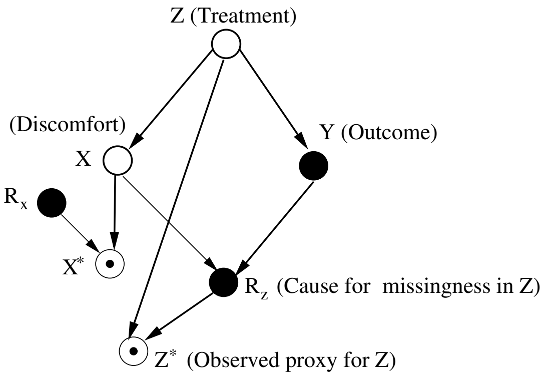


Figure 7.2: m-graph with no conditional independence statement that is testable using the criteria in section 7.1.2

7.2 Impediments to Testability in Missing Data

Unlike testability under complete data, testability in missing data has an impediment to overcome. When data are complete we simply select a conditional independence statement in the model and test it against the data. Under missing data however, some conditional independencies in the model may not be testable even when the joint distribution is recoverable. An example demonstrating this impediment is discussed below.

Example 27. *Consider the missingness process described by the graph G in Figure 7.5 (a) that states the CI: $X \perp\!\!\!\perp R_x|Y$. Let $Q : P(X, Y, R_x)$ be the query to be recovered. We will show that although Q is recoverable, the CI statement $X \perp\!\!\!\perp R_x|Y$ is not testable.*
First we will prove that Q is recoverable.

$$P(X, Y, R_x = 1) = P(X|Y, R_x = 1)P(Y, R_x = 1)$$

Since G embeds $X \perp\!\!\!\perp R_x|Y$ we have,$P(X|Y, R_x = 1) = P(X|Y, R_x = 0)$. Therefore,

$$P(X, Y, R_x = 1) = P(X|Y, R_x = 0)P(Y, R_x = 1)$$

Using Equation 2.1,$P(X|Y, R_x = 0) = P(X^*|Y, R_x = 0)$. Therefore,

$$P(X, Y, R_x = 1) = P(X^*|Y, R_x = 0)P(Y, R_x = 1)$$

Hence $P(X, Y, R_x = 1)$ is recoverable.
Using Equation 2.1,
$P(X, Y, R_x = 0) = P(X^*, Y, R_x = 0)$. Thus, $P(X, Y, R_x = 0)$ is also recoverable.
We will now show that $X \perp\!\!\!\perp R_x|Y$ is not testable. $X \perp\!\!\!\perp R_x|Y$ translates into,
$P(X|Y, R_x = 1) = P(X|Y, R_x = 0)$
Hence, $P(X, Y, R_x = 1) = \frac{P(X,Y,R_x=0)}{P(Y,R_x=0)}P(Y, R_x = 1)$
In other words, for any manifest distribution $P^*(X^*, Y, R_X)$ in which $P^*(Y, R_X = 0) > 0$, we can always construct (as shown below) a compatible distribution $P(X, Y, R_X)$ in which the CI statement $X \perp\!\!\!\perp R_x|Y$ holds.
$\forall x, y$

$$\begin{aligned} P(X = x, Y = y, R_x = 0) &= P^*(X^* = x, Y = y, R_x = 0) \\ P(X = x, Y = y, R_x = 1) &= \frac{P^*(X^* = x, Y = y, R_x = 0)}{P^*(Y = y, R_x = 0)} \\ &\quad * P^*(Y = y, R_x = 1) \end{aligned}$$

Thus, $X \perp\!\!\!\perp R_x|Y$ is not refutable and hence we conclude that it is not testable.

This example showed that a probability distribution $P(v)$ can be perfectly recoverable from missingness, (i.e., it can be estimated consistently, *as if* no missingness occurred) and yet, $P(v)$ may have testable implications (eg, conditional independence (CI) statements) that are not testable for any data with the same manifest structure (i.e. the same sets of partially and fully observed variables).

The explanation of this impediment is as follows. When we say $P(V)$ has testable implications we refer to refutation by some distribution taken from the space of all distributions on V. In contrast, when we say 'testable under missingness' we demand refutation by a set of distributions with the same manifest structure. The refutation power of the latter set is weaker than the former.

The next theorem characterizes a set of CI that are not testable from missing data.

Theorem 14. *Given that $Y \subseteq V_o \cup R$, the singleton d-separation $X \perp\!\!\!\perp R_x|Y$ is not testable.*

Proof. We can always compute $P(X, R_x = 1, Y)$ as $P(X, R_x = 1, Y) = P(R_x = 1, Y)P(X^*|R_x = 0, Y)$ such that $X \perp\!\!\!\perp R_x|Y$ is always true. Hence $X \perp\!\!\!\perp R_x|Y$ is not refutable given any manifest distribution that is strictly positive over complete cases. Hence $X \perp\!\!\!\perp R_x|Y$ is not testable. □

Corollary 3. *Given that Y contains at least one partially observed variable and $R_{y_m} \subset Y$, singleton conditional independence $X \perp\!\!\!\perp R_x|Y_r = 0, Y - Y_r$ is not testable.*

Corollary 4. *Direct Testability of a conditional independence statement does not imply testability of all its implications.*

Proof. Consider the CI statement $X \perp\!\!\!\perp (Y, R_y, R_x)$. On applying decomposition graphoid axiom, we get the non-testable CI: $X \perp\!\!\!\perp R_x$. □

However, there exist directly testable d-separations whose implications obtained by weak union and decomposition graphoid axioms are always testable.

Example 28. *Let $X \perp\!\!\!\perp Y|Z, R_z, R_x, R_y$ be a compound d-separation such that $X \cup Y \cup Z \subseteq V_o \cup V_m$. In this case it can be easily seen that all implications obtained by applying decomposition and weak union graphoid axioms comply with conditions for direct testability given in Theorem-13. Hence they are all testable.*

7.3 Testability of CIs comprising of only substantive variables

Let us examine the testability of singleton CI: $X \perp\!\!\!\perp Y$. Clearly, when $X, Y \in V_o$, $X \perp\!\!\!\perp Y$ is testable. However, testability of $X \perp\!\!\!\perp Y$ when $X \in V_o$ and $Y \in V_m$ is not obvious. In the following theorem we prove that $X \perp\!\!\!\perp Y$ is testable when $X \in V_o$ and $Y \in V_m$ and, X and Y are *binary*. We further specify necessary conditions that the manifest distribution must satisfy for $X \perp\!\!\!\perp Y$ to hold true in the underlying distribution.

Theorem 15. *Given that $X \in V_o$ and $Y \in V_m$, the conditional independence statement $X \perp\!\!\!\perp Y$ is testable. Moreover, a graph depicting $X \perp\!\!\!\perp Y$ should be summarily rejected if none of the following conditions hold:*

$$0 \leq \frac{-k}{P(x)} \leq P(x', r_y) \tag{7.4}$$

$$0 \leq \frac{k}{P(x')} \leq P(x, r_y) \tag{7.5}$$

$$0 \leq \frac{k + P(x)P(x', r_y)}{P(x')} \leq P(x, r_y) \tag{7.6}$$

$$0 \leq \frac{P(x')P(x, r_y) - k}{P(x)} \leq P(x', r_y) \tag{7.7}$$

where $k = P(x)(P(x', y, r'_y) + P(x, y, r'_y)) - P(x, y, r'_y)$.

Proof. We first show that violation of all conditions from 7.4 to 7.7 is sufficient to rule out $X \perp\!\!\!\perp Y$. Then by constructing an example that violates conditions 7.4 to 7.7, we confirm the testability of $X \perp\!\!\!\perp Y$.

$X \perp\!\!\!\perp Y$ may be equivalently written as,

$P(x, y) = P(x)P(y)$

The equation above is equivalent to,

$P(x, y, r_y) - P(x)(P(x', y, r_y) + P(x, y, r_y)) = P(x)(P(x', y, r'_y) + P(x, y, r'_y)) - P(x, y, r'_y)$

Let the constant terms in RHS evaluate to k. Then we can rewrite the equation as:

$$P(x', y, r_y) = \frac{P(x')}{P(x)} P(x, y, r_y) + \frac{-k}{P(x)} \tag{7.8}$$

Equation 7.8 is linear, the variables are $P(x', y, r_y)$ and $P(x, y, r_y)$ and it resembles the general equation of a line:*y*=*mx*+*c*. Equation 7.8 should also satisfy:

(a) $0 \leq P(x, y, r_y) \leq P(x, r_y)$

(b) $0 \leq P(x', y, r_y) \leq P(x', r_y)$

The constraints (a) and (b) above delineate a rectangular region $\Re$ in the first quadrant of the Cartesian plane. Equation 7.8 can be solved subject to constraints (a) and (b) only if the line described in Equation 7.8 intersects the boundary lines enclosing $\Re$ (i.e. at least one intersection point should satisfy (a) and (b)).

Intersection of Eq 7.8 and left boundary of $\Re$ yields:

$0 \leq \frac{-k}{P(x)} \leq P(x', r_y)$

Intersection of Eq 7.8 and bottom boundary of $\Re$ yields:

$0 \leq \frac{k}{P(x')} \leq P(x, r_y)$

Intersection of Eq 7.8 and top boundary of $\Re$ yields:

$0 \leq \frac{k+P(x)P(x',r_y)}{P(x')} \leq P(x, r_y)$

Intersection of Eq 7.8 and right boundary of $\Re$ yields:

$0 \leq \frac{P(x')P(x,r_y)-k}{P(x)} \leq P(x', r_y)$

We prove testability of $X \perp\!\!\!\perp Y$ by presenting manifest distribution P_3 in Table 7.1 that violates conditions 7.4 to 7.7 and thus refutes the claim: $X \perp\!\!\!\perp Y$. □

R_y	X	Y^*	P_1	P_2	P_3
0	0	0	$\frac{8}{81}$	$\frac{13}{41}$	$\frac{100}{125}$
0	0	1	$\frac{6}{81}$	$\frac{11}{41}$	$\frac{5}{125}$
0	1	0	$\frac{4}{81}$	$\frac{7}{41}$	$\frac{7}{125}$
0	1	1	$\frac{3}{81}$	$\frac{5}{41}$	$\frac{3}{125}$
1	0	m	$\frac{20}{81}$	$\frac{3}{41}$	$\frac{5}{125}$
1	1	m	$\frac{40}{81}$	$\frac{2}{41}$	$\frac{5}{125}$

Table 7.1: $X \perp\!\!\!\perp Y$ can hold in manifest distributions P_1 and P_2 but cannot hold in manifest distribution P_3

Example 29. *Table 7.1 describes three distributions; P_1 and P_2 in which $X \perp\!\!\!\perp Y$ could possibly hold and P_3 in which $X \perp\!\!\!\perp Y$ cannot hold. $X \perp\!\!\!\perp Y$ can possibly hold in P_1 and P_2 because both the distributions satisfy condition 7.5. P_3 does not satisfy any of the conditions from 7.4 to 7.7; hence $X \perp\!\!\!\perp Y$ cannot hold in P_3.*

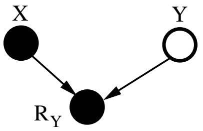


Figure 7.3: m-graph in which recoverability of $P(X|Y)$ depends only on $X \perp\!\!\!\perp Y$.

The following example demonstrates an application of Theorem 15. It describes an instance where recoverability of a given query hinges exclusively on the independence between X and Y.

Example 30. *Let G_1 in Figure 7.3 be the hypothesized graph and $Q = P(X|Y)$ be the query to be recovered. $P(X, Y)$ is not recoverable from G_1 since Y itself is the cause of its missingness(R_y). G_1 embeds the CI statement: $X \perp\!\!\!\perp Y$ and if we assume G_1 is the true graph then $P(X|Y)$ can be recovered as follows:*

$$P(X|Y) = P(X)$$

Recoverability however depends critically on the independence $X \perp\!\!\!\perp Y$embedded in G_1. Our question is whether or not the CI statement $X \perp\!\!\!\perp Y$ holds in any underlying distribution compatible with the data available. Theorem 15 answers this question immediately by providing us with four conditions, one of which ought to be satisfied by the manifest distribution for $X \perp\!\!\!\perp Y$ to hold. For example, given P_3 in Table 7.1 and G_1, we can immediately conclude that G_1 and P_3 are not compatible.

It is interesting to note that though recoverability is generally facilitated by (usually non-testable) CI between a variable and its missingness mechanism such as $Y \perp\!\!\!\perp R_y$ or $Y \perp\!\!\!\perp R_y|X$, in Example 30 recoverability of Q facilitated by the independence between substantive variables X and Y.

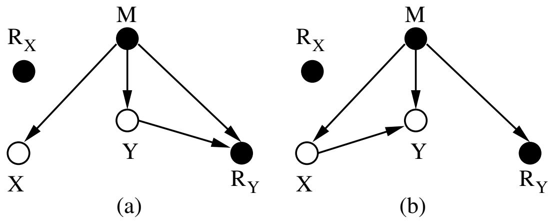


Figure 7.4: (a) m-graph depicting MNAR (b) m-graph depicting MAR^+

7.4 Testability of MCAR and MAR

Missingness mechanisms are traditionally classified into three categories ([Rub76]): Missing Completely At Random(MCAR), Missing At Random(MAR) and Missing Not At Random (MNAR). A chi square based test for MCAR was proposed by [Lit88] in which a high value falsified MCAR([Rub76]). MAR ([Rub76]) is not testable([All02], page 4).

MAR as defined in Chapter 2 was shown to be testable in ([PTP06]). Theorem 16, given below presents stronger conditions under which a given MAR^+ model is testable. Furthermore, it provides diagnostic insight in case the test is violated.

Theorem 16. *Given that $|V_m| > 0$, MAR^+ ($V_m \perp\!\!\!\perp R|V_o$) is testable if and only if $|V_m| > 1$ i.e. $|V_m|$ is not a singleton set.*

Proof. Let $|V_m| = k > 1$ and $X \subseteq V_m$ such that $|X| = k - 1$. By applying decomposition graphoid axiom to $V_m \perp\!\!\!\perp R|V_o$, we get $(V_m - X) \perp\!\!\!\perp R|V_o$ that is directly testable by Theorem 13. Therefore, $V_m \perp\!\!\!\perp R|V_o$ is testable if V_m is not a singleton. On the other hand if $|V_m| = 1$ then by Theorem-14, $V_m \perp\!\!\!\perp R|V_o$ is not testable. □

Example 31. *In the graph in Figure 7.4(b), MAR^+ holds because $(Y, X) \perp\!\!\!\perp (R_x, R_y)|M$. Therefore, the tests are:*

(i) $X^* \perp\!\!\!\perp R_y|M, R_x = 0$

(ii) $Y^* \perp\!\!\!\perp R_x|M, R_y = 0$

(iii) $X^* \perp\!\!\!\perp R_y = 0|M, R_x = 0, Y$

(iv) $Y^* \perp\!\!\!\perp R_x = 0|M, R_y = 0, X$

(i) and (ii) are tests obtained by applying weak union and decomposition graphoid axioms to $(Y, X) \perp\!\!\!\perp (R_x, R_y)|M$ and (iii) and (iv) are tests obtained by applying weak-union graphoid ax-

iom to $(Y, X) \perp\!\!\!\perp (R_x, R_y)|M$. Note that the graph has more testable implications than those listed above. For example, the graph advertises the CI statement $R_x \perp\!\!\!\perp R_y$. However, the latter test is model specific, whereas (i)-(iv) are model-independent, applicable to any MAR^+ model with the same manifest structure.

The following corollary shows that MCAR is testable.

Corollary 5. *Given that $|V_m| > 0$, MCAR ($(V_m \cup V_o) \perp\!\!\!\perp R$) is testable if and only if $|V_o \cup V_m| \geq 2$.*

If the dataset contains only one variable(X) and $X \in V_m$, then $X \perp\!\!\!\perp R_x$ is not testable (by Theorem 14), even though the corresponding missingness mechanism is MCAR. If the dataset additionally contained at least another fully observed variable (Y) then $(X, Y) \perp\!\!\!\perp R_x$ is testable since its implication $Y \perp\!\!\!\perp R_x$ is testable. On the other hand, if the dataset additionally contained at least another partially observed variable (Z) then $(X, Z) \perp\!\!\!\perp (R_x, R_z)$ is testable since its implications such as $Z \perp\!\!\!\perp R_x|R_z = 0$ and $X \perp\!\!\!\perp R_z|R_x = 0$ are testable.

7.4.1 Detecting MNAR missingness mechanism

Consider the graph in Figure 7.4(a). The model is clearly MNAR since there is an edge between Y and R_y. However, Theorem 16 will not be able to falsify MAR^+. The following subsection will show that such falsification is nevertheless possible.

7.4.1.1 Graph based tests for detecting the edge between a variable an its missingness mechanism (eg: $X \rightarrow R_x$)

Ordinarily an edge E between a variable and its missingness mechanism is not testable. However, if the contentious edge is embedded in a structure that meets certain conditions we will show that a test exists to ascertain the existence of E. The following lemma gives the condition under which an edge $X \rightarrow R_x$ may be detected in a Markovian Model.

Lemma 7. *Given a Markovian model in which (1) there exists Z which is a parent of X and not a parent of R_x and (2) no R variable is a parent of another R variable, an edge $X \rightarrow R_x$ exists whenever $Z \not\perp\!\!\!\perp R_x|R_z = 0, (R \cup V) - \{X, Z, R_x, R_z\}$.*

Proof. Condition (2) prevents R_x from being a parent of any node in R and by definition of m-graph R_x cannot be a parent of variables in $V_o \cup V_m$. Hence no variable in $V_o \cup V_m \cup R$ is a child of any R variable. Moreover, the model is Markovian. Therefore the m-graph can only contain uni-directed edges that enter R_x and thus no parent of R_x can be a collider on any path that enters R_x. In the test, $Z \perp\!\!\!\perp R_x | R_z = 0, (R \cup V) - \{X, Z, R_x, R_z\}$ we condition on all variables except X. Therefore, if the test does not hold true then it is because there is an unblocked path from Z to R_x via X (by condition-1, $Z \to R_x$ does not exist). This is possible only if X is a parent of R_x i.e. there exists an edge between X and R_x.

□

Example 32. *Consider the m-graph G_1 in Figure 7.2 that implies $X \perp\!\!\!\perp R_x$. Let it be the case that Z does not cause the missingness in X. Then, we can confirm dependence i.e. the existence of $X \to R_x$, if $Z \perp\!\!\!\perp R_x | Y, R_z = 0$ does not hold.*

7.5 Model Sensitivity of Estimation Procedures

An important consequence of identifying the testable implications of a given model is the ability to demonstrate the limits of model-blind algorithms, i.e. algorithms that attempt to handle missing-data problems on the basis of the data alone, without making any assumptions about the structure of the missingness process. A fundamental limitation of model-blind algorithms is unveiled in Example 33, which presents two statistically indistinguishable models such that a given query is recoverable in one and non-recoverable in the other.

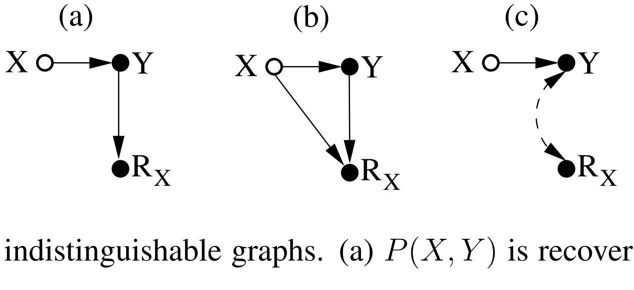


Figure 7.5: Statistically indistinguishable graphs. (a) $P(X, Y)$ is recoverable (b) $P(X, Y)$ is not recoverable (c) $P(X)$ is recoverable

Example 33. *The two graphs in Fig. 7.5 (a) and (b) cannot be distinguished by any statistical means, since Fig. 7.5(a) has no testable implications and Fig. 7.5(b) is a complete graph. However in Fig. 7.5 (a) $P(X, Y)$ is recoverable (refer Example 27)while in Fig. 7.5 (b) $P(X, Y)$ is not recoverable (by Theorem-2 in [MPT13]).*

An even stronger limitation is demonstrated in Example 34; it shows that no model-blind algorithm exists[1] even in those cases where recoverability is feasible. We construct two statistically indistinguishable models, G_1 and G_2, dictating different estimation procedure S_1 and S_2 respectively; yet Q is not recoverable in G_1 by S_2 or in G_2 by S_1.

Example 34. *The graphs in Fig. 7.5 (a) and (c) are statistically indistinguishable; neither has testable implications. Let the target relation of interest be $Q = P(X)$. In Fig. 7.5 (a), Q may be estimated as $P(X) = \sum_y P(X|Y, R_x = 0)P(Y)$ since $X \perp\!\!\!\perp R_x|Y$ and in Fig. 7.5 (b), Q can be derived as $P(X) = P(X|R_x = 0)$ since $X \perp\!\!\!\perp R_x$.*

7.6 Summary

In this chapter we illuminated the boundary between testable and non-testable models with emphasis on models which as considered MNAR in the literature. We have provided syntactic rules for ensuring testability of given conditional independence claims (CI) based on the type of variables that appear in the CI. We further presented conditions for non-testability of CI and discussed a peculiar property of testability in missing data. We refined the results of [PTP06] and showed that the class of models denominated as MAR^+ are testable as long as $V_m \geq 2$ and the class of models denominated as MCAR are testable as long as $V_o \cup V_m \geq 2$. Additionally we presented graphical conditions that would allow a test of confirmation of dependence between a variable and its missingness mechanism. Finally we demonstrated sensitivity of missing data recovery procedures to hypothesized models and further proved that this sensitivity is inevitable in datasets classified as MNAR.

[1] We leave open the unlikely possibility that there exists an estimation scheme, different from ours that could recover $Q = P(X)$ in both models. We propose this example as a litmus test for any such estimator.

CHAPTER 8

Robust Algorithms for Closed-form Estimation

In this chapter we demonstrate how our recoverability results derived in chapter 3 using graphical models can be applied to the problem of estimation. To this end, we develop a suite of algorithms for learning the parameters of Bayesian Network. When learning the parameters of a Bayesian network from data with missing values, the conventional wisdom among machine learning practitioners is that there are two options: use *expectation maximization* (EM) or *gradient methods* (to optimize the likelihood); see, e.g., [Dar09, KF09, Mur12, Bar12]. Both of these approaches, however, suffer from the following disadvantages, which prevent them from scaling to large networks and datasets; see also [TMH01]. First, they are *iterative*, and hence may need many passes over a potentially large dataset. Next, these algorithms may get stuck in *local optima*, which means that, in practice, one must run these algorithms multiple times with different initial seeds, and hope that one of them leads to a good optimum. Last, but not least, these methods *require inference* in the network, which places a hard limit on the networks where EM and gradient methods can even be applied, namely for networks where exact inference is tractable, i.e., they have small enough treewidth, or sufficient local structure [CD06, CD07].

In this chapter, we propose a family of practical and efficient algorithms for estimating the parameters of a Bayesian network from incomplete data. For the cases of both MCAR and MAR data, where the missingness graph need not be explicit, we start by deriving the closed-form parameter estimates, as implied by [MPT13]. We next show how to obtain better estimates, by exploiting a factorized representation that allows us to aggregate distinct, yet asymptotically equivalent estimates, hence utilizing more of the data. We also show how to obtain improved estimates, when the missingness graph is only partially explicated (based on domain or expert knowledge). As in [MPT13], all of our estimation algorithms are asymptotically consistent, i.e., they converge to the true parameters of a network, in the limit of infinite data.

As we show empirically, our parameter estimation algorithms make learning from incomplete data viable for larger Bayesian networks and larger datasets, that would otherwise be beyond the scope of algorithms such as EM and gradient methods. In particular, our algorithms (1) are non-iterative, requiring only a single pass over the data, (2) provide estimates in closed-form, and hence do not suffer from local optima, and (3) require no inference, which is the primary limiting factor for the scalability of algorithms such as EM. We note that these advantages are also available when learning Bayesian networks from *complete* data.

Chapter Outline: We present closed form estimation algorithms for MCAR and MAR problems in section 8.1. Empirical results are discussed in section 8.2. Section 8.4 summarizes the related works.

Notations We use upper case letters (X) to denote variables and lower case letters (x) to denote their values. Variable sets are denoted by bold-face upper case letters ($\mathbf{X}$) and their instantiations by bold-face lower case letters ($\mathbf{x}$). Generally, we will use X to denote a variable in a Bayesian network and $\mathbf{U}$ to denote its parents. A network parameter will therefore have the general form $\theta_{x|\mathbf{u}}$, representing the probability $\Pr(X{=}x|\mathbf{U}{=}\mathbf{u})$.

Given an incomplete dataset $\mathfrak{D}$, we want to learn the parameters of the Bayesian network $\mathcal{N}$ that the dataset originated from. This network induces a distribution $\Pr(\mathbf{X})$, which is in general unknown; instead, we only have access to the dataset $\mathfrak{D}$.

8.1 Closed-form Learning

We now present algorithms to learn the parameters of a Bayesian network $\mathcal{N}$ from data $\mathfrak{D}$. We first consider the classical missing data assumptions, with no further knowledge about the missingness graph that generated the data.

To estimate the conditional probabilities $\theta_{x|\mathbf{u}}$ that parameterize a Bayesian network, we estimate the joint distributions $\Pr(X, \mathbf{U})$, which are subsequently normalized, as a conditional probability table. Hence, it suffices, for our discussion, to estimate marginal distributions $\Pr(\mathbf{Y})$ for families $\mathbf{Y} = \{X\} \cup \mathbf{U}$. We let $\mathbf{Y}_o = \mathbf{Y} \cap \mathbf{X}_o$ denote the observed variables in $\mathbf{Y}$, and $\mathbf{Y}_m = \mathbf{Y} \cap \mathbf{X}_m$ denote the partially-observed variables. Further, we let $\mathbf{R}_{\mathbf{Z}} \subseteq \mathbf{R}$ denote the missingness mechanisms for

Table 8.1: Summary of Estimation Algorithms

Algorithm	Description (Section Number)
D-MCAR	Direct Deletion for MCAR data (8.1.1)
D-MAR	Direct Deletion for MAR data (8.1.2)
F-MCAR	Factored Deletion for MCAR data (8.1.3)
F-MAR	Factored Deletion for MAR data (8.1.3)
I-MAR	Informed Deletion for MAR data (8.3.1)
IF-MAR	Informed Factored Deletion for MAR data (8.3.1)

the partially-observed variables $\mathbf{Z}$. Through $\mathfrak{D}$, we have access to the data distribution $\Pr_{\mathfrak{D}}$ over the variables in the missingness dataset. Appendix A.4.4 illustrates our learning algorithms on a concrete dataset and Table 8.1 gives an overview of the different estimation algorithms in this paper.

8.1.1 Direct Deletion for MCAR

The statistical technique of *listwise deletion* is perhaps the simplest technique for performing estimation with MCAR data: we simply delete all instances in the dataset that contain missing values, and estimate our parameters from the remaining dataset, which is now complete. Of course, with this technique, we potentially ignore large parts of the dataset. The next simplest technique is perhaps pairwise deletion, or available-case analysis: when estimating a quantity over a pair of variables X and Y, we delete just those instances where variable X or variable Y is missing.

Consider now the following, more general, deletion technique, which is expressed in the terms of causal missingness mechanisms. In particular, to estimate the marginals $\Pr(\mathbf{Y})$ of a set of (family) variables $\mathbf{Y}$, from the data distribution $\Pr_{\mathfrak{D}}$, we can use the estimate:

$$\begin{aligned}
\Pr(\mathbf{Y}) &= \Pr(\mathbf{Y}_o, \mathbf{Y}_m | \mathbf{R}_{\mathbf{Y}_m}{=}0) && \text{by } \mathbf{X}_o, \mathbf{X}_m \perp\!\!\!\perp \mathbf{R} \\
&= \Pr(\mathbf{Y}_o, \mathbf{Y}^\star_m | \mathbf{R}_{\mathbf{Y}_m}{=}0) && \text{by } \mathbf{X}_m{=}\mathbf{X}^\star_m \text{ when } \mathbf{R}{=}0 \\
&\approx \Pr\nolimits_{\mathfrak{D}}(\mathbf{Y}_o, \mathbf{Y}^\star_m | \mathbf{R}_{\mathbf{Y}_m}{=}0)
\end{aligned}$$

That is, we can estimate $\Pr(\mathbf{Y})$ by using the subset of the data where every variable in $\mathbf{Y}$ is observed (which follows from the assumptions implied by MCAR data). Since the data distribution $\Pr_{\mathcal{D}}$ tends to the true distribution $\Pr$, this implies a consistent estimate for the marginals $\Pr(\mathbf{Y})$. In contrast, the technique of listwise deletion corresponds to the estimate $\Pr(\mathbf{Y}) \approx \Pr_{\mathcal{D}}(\mathbf{Y}_o, \mathbf{Y}^{\star}_m | \mathbf{R}_{\mathbf{X}_m}{=}0)$, and the technique of pairwise deletion corresponds to the above, when $\mathbf{Y}$ contains two variables. To facilitate comparisons with more interesting estimation algorithms that we shall subsequently consider, we refer to the more general estimation approach above as *direct deletion*.

8.1.2 Direct Deletion for MAR

In the case of MAR data, we cannot use the simple deletion techniques that we just described for MCAR data—the resulting estimates would not be consistent. However, we show next that it is possible to obtain consistent estimates from MAR data, using a technique that is as simple and efficient as direct deletion. Roughly, we can view this technique as deleting certain instances from the dataset, but then re-weighting the remaining ones, so that a consistent estimate is obtained. We shall subsequently show how to obtain even better estimates by factorization.

Again, to estimate network parameters $\theta_{x|\mathbf{u}}$, it suffices to show how to estimate family marginals $\Pr(\mathbf{Y})$, now under the MAR assumption. Let $\mathbf{X}'_o = \mathbf{X}_o \setminus \mathbf{Y}_o$ denote the fully-observed variables outside of the family variables $\mathbf{Y}$ (i.e., $\mathbf{X}_o = \mathbf{Y}_o \cup \mathbf{X}'_o$). We have

$$\begin{aligned}\Pr(\mathbf{Y}) &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_o, \mathbf{Y}_m, \mathbf{X}'_o) \\ &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_m | \mathbf{Y}_o, \mathbf{X}'_o) \Pr(\mathbf{Y}_o, \mathbf{X}'_o)\end{aligned}$$

Hence, we reduced the problem to estimating two sets of probabilities. Estimating the probabilities $\Pr(\mathbf{Y}_o, \mathbf{X}'_o)$ is straightforward, as variables $\mathbf{Y}_o$ and $\mathbf{X}'_o$ are fully observed in the data. The conditional probabilities $\Pr(\mathbf{Y}_m | \mathbf{Y}_o, \mathbf{X}'_o)$ contain partially observed variables $\mathbf{Y}_m$, but they are conditioned on all fully observed variables $\mathbf{X}_o = \mathbf{Y}_o \cup \mathbf{X}'_o$. The MAR definition implies that each subset of the data that fixes a value for $\mathbf{X}_o$ is locally MCAR. Like the MCAR case, we can estimate each conditional probability as

$$\Pr(\mathbf{Y}_m | \mathbf{Y}_o, \mathbf{X}'_o) = \Pr(\mathbf{Y}^{\star}_m | \mathbf{Y}_o, \mathbf{X}'_o, \mathbf{R}_{\mathbf{Y}_m}{=}0).$$

Algorithm 3 F-MCAR$(\mathbf{y}, \mathfrak{D})$

```
Input:
 $\mathbf{y}$:  A state of query variables $\mathbf{Y}$
 $\mathfrak{D}$:  An incomplete dataset with data distribution $\Pr_{\mathfrak{D}}$
Auxiliary:
 CACHE:  A global cache of estimated probabilities
Function:
 if $\mathbf{y} = \emptyset$ then return $1$
 if CACHE$[\mathbf{y}] \neq nil$ then return CACHE$[\mathbf{y}]$
 $\mathcal{E} \leftarrow \emptyset$                    // Initialize set of estimates for all $y \in \mathbf{y}$ do
 $\mathbf{u} \leftarrow \mathbf{y} \setminus \{y\}$                          // Factorize with parents $\mathbf{u}$
 add $\Pr_{\mathfrak{D}}(y|\mathbf{u}, \mathbf{R}_{\mathbf{y}}{=}0) \cdot$ F-MCAR$(\mathbf{u}, \mathfrak{D})$ to $\mathcal{E}$ end for
 CACHE$[\mathbf{y}]$ ←Aggregate estimates in $\mathcal{E}$          // E.g., mean return CACHE$[\mathbf{y}]$
```

This leads to the following estimation algorithm,

$$\Pr(\mathbf{Y}) \approx \sum_{\mathbf{X}'_o} \Pr_{\mathfrak{D}}(\mathbf{Y}^{\star}_m | \mathbf{Y}_o, \mathbf{X}'_o, \mathbf{R}_{\mathbf{Y}_m}{=}0) \Pr_{\mathfrak{D}}(\mathbf{Y}_o, \mathbf{X}'_o)$$

which uses only the fully-observed variables of the data distribution $\Pr_{\mathfrak{D}}$. Note that the summation requires only a single pass through the data, i.e., for only those instantiations of $\mathbf{X}'_o$ that appear in it. Again, $\Pr_{\mathfrak{D}}$ tends to the true distribution $\Pr$, as the dataset size tends to infinity, implying a consistent estimate of $\Pr(\mathbf{Y})$.

8.1.3 Factored Deletion

We now propose a class of deletion algorithms that exploit more data than direct deletion. In the first step, we generate multiple but consistent estimates for the query so that each estimates utilizes different parts of a dataset to estimate the query. In the second step, we aggregate these estimates to compute the final estimate and thus put to use almost all tuples in the dataset. Since this method exploits more data than direct deletion, it obtains a better estimate of the query.

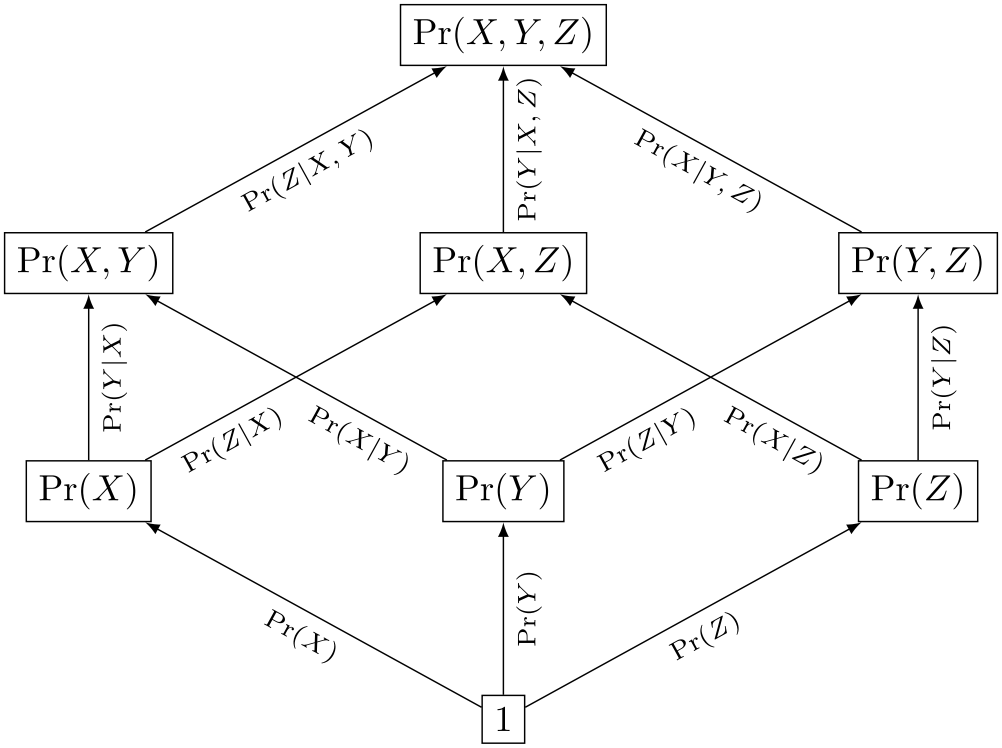


Figure 8.1: Factorization Lattice of $\Pr(X, Y, Z)$

Factored Deletion for MCAR Algorithm 3 implements factored deletion for MCAR. Let the query of interest be $\Pr(\mathbf{Y})$, and let $Y^1, Y^2, \ldots, Y^n$ be any ordering of the n variables in $\mathbf{Y}$. Each ordering yields a unique factorization:

$$\Pr(\mathbf{Y}) = \prod_{i=1}^{n} \Pr\left(Y^i \mid Y^{i+1}, \ldots, Y^n\right)$$

We can estimate each of these factors independently, on the subset of the data in which all of its variables are fully observed (as in direct deletion), i.e.,

$$\Pr(Y^i|Y^{i+1}, \ldots, Y^n_m) = \Pr(Y^i|Y^{i+1}, \ldots, Y^n_m, \mathbf{R}_{\mathbf{Z}^i}{=}\mathsf{0})$$

where $\mathbf{Z}^i$ is the set of partially-observed variables in the factor. When $|\mathbf{Y}_m| > 1$, we can utilize much more data than direct deletion. See Appendix A.4.4, for an example.

So far, we have discussed how a consistent estimate of $\Pr(\mathbf{Y})$ may be computed given a factorization. Now we shall detail how estimates from each factorization can be aggregated to compute more accurate estimates of $\Pr(\mathbf{Y})$. Let k be the number of variables in a family $\mathbf{Y}$. The number of possible factorizations is $k!$. However, different factorizations share the same sub-factors, which

Algorithm 4 F-MAR$(\mathbf{y}, \mathfrak{D})$

Input:

$\mathbf{y}$: A state of query variables $\mathbf{Y}$, consisting of $\mathbf{y}_o$ and $\mathbf{y}_m$

$\mathfrak{D}$: An incomplete dataset with data distribution $\Pr_{\mathfrak{D}}$

Function:

```
$e \leftarrow 0$        // Estimated probability for all $\mathbf{x}_o$ appearing in $\mathfrak{D}$ that agrees with $\mathbf{y}_o$ do
$\mathfrak{D}_{\mathbf{x}_o} \leftarrow$ subset of $\mathfrak{D}$ where $\mathbf{x}_o$ holds
$e \leftarrow e + \Pr_{\mathfrak{D}}(\mathbf{x}_o) \cdot$ F-MCAR$(\mathbf{y}_m, \mathfrak{D}_{\mathbf{x}_o})$ end for return $e$
```

we can estimate once, and reuse across factorizations. We can organize these computations using a lattice, as in Figure 8.1, which has only 2^k nodes and $k2^{k-1}$ edges. Our algorithm will compute as many estimates as there are edges in this lattice, which is only on the order of $O(n \log n)$, where n is the number of parameters being estimated for a family Y (which is also exponential in the number of variables k). To emphasize the distinction with direct deletion, which uses only those instances in the data where *all* variables in $\mathbf{Y}$ are observed, factored deletion uses any instance in the data where *at least one* variable in $\mathbf{Y}$ is observed.

More specifically, our factored deletion algorithm first estimates the conditional probabilities on the edges of the lattice, each estimate using the subset of the data where its variables are observed. Second, it propagate the estimates, bottom-up. For each node, there are several alternative estimates available, on its incoming edges. There are various ways of aggregating these estimates, such as mean, median, and propagating the lowest-variance estimate.[1]

Factored Deletion for MAR Algorithm 4 implements factored deletion for MAR. Let $Y_m^1, Y_m^2, \ldots, Y_m^n$ be any ordering of the n partially observed variables $\mathbf{Y}_m \subseteq \mathbf{Y}$ and let $\mathbf{X}'_o = \mathbf{X}_o \setminus \mathbf{Y}_o$ denote the fully-observed variables outside of $\mathbf{Y}$. Given an ordering, we have the factorization:

$$\Pr(\mathbf{Y}) = \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_o, \mathbf{X}'_o) \prod_{i=1}^{n} \Pr\left(Y_m^i \mid \mathbf{Z}_m^{i+1}, \mathbf{X}_o\right)$$

[1] In initial experiments, all aggregations performed similarly. Reported results use an inverse-variance weighting heuristic.

where $\mathbf{Z}_m^i = \{Y_m^j | i \leq j \leq n\}$. We then proceed in a manner similar to factored deletion for MCAR to estimate individual factors and aggregate estimates to compute $\Pr(\mathbf{Y})$. For equations and derivations, please see Appendix A.4.1.

8.2 Empirical Evaluation

To evaluate the learning algorithms we proposed, we simulate partially observed datasets from Bayesian networks, and re-learn their parameters from the data.[2]

In our first sets of experiments, we compare our parameter estimation algorithms with EM, on relatively small networks for MCAR and MAR data. These experiments are intended to observe general trends in our algorithms, in terms of their computational efficiency, but also in terms of the quality of the parameter estimates obtained. Our main empirical contributions are presented in Section 8.2.3, where we demonstrate the scalability of our proposed estimation algorithms, to larger networks and datasets, compared to EM (even when using approximate inference algorithms).

We consider the following algorithms:

D-MCAR & F-MCAR: direct deletion and factored deletion for MCAR data.

D-MAR & F-MAR: direct deletion and factored deletion for MAR data.

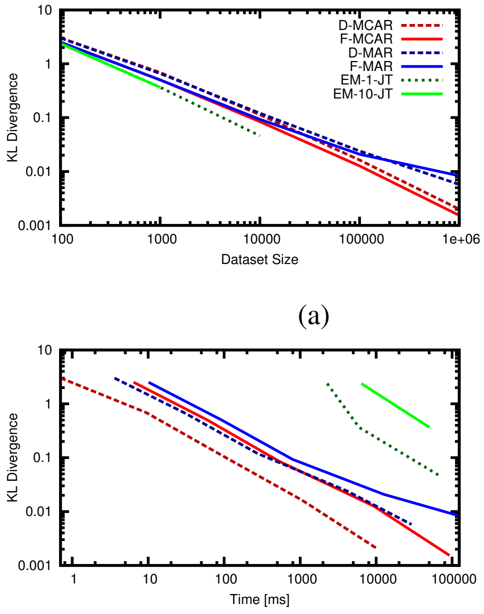
EM-k-JT: EM with k random restarts, jointree inference.

F-MAR + EM-JT: EM seeded with F-MAR estimates, jointree inference.

Remember that D-MCAR and F-MCAR are consistent for MCAR data only, while D-MAR and F-MAR are consistent for general MAR data. EM is consistent for MAR data, but only if it converges to maximum-likelihood estimates.

We evaluate the learned parameters in terms of their *likelihood* on independently generated, fully-observed test data, and the *Kullback–Leibler divergence* (KLD) between the original and learned Bayesian networks. We report per-instance log-likelihoods (which are divided by dataset size). We evaluate the learned models on unseen data, so all learning algorithms assume a symmetric

[2] An implementation of our system is available at `http://reasoning.cs.ucla.edu/deletion`.



(a)

(b)

Figure 8.2: Learning the `alarm` network from MCAR data.

Dirichlet prior on the network parameters with a concentration parameter of 2 (which corresponds to Laplace smoothing).

8.2.1 MCAR Data

First, we consider learning from *MCAR data*, evaluating the quality of the parameters learned by each algorithm. We simulate training sets of increasing size, from a given Bayesian network, selecting 30% of the variables to be partially observed, and removing 70% of their values completely at random. All reported numbers are averaged over 32 repetitions with different learning problems. When no number is reported, a 5 minute time limit was exceeded.

To illustrate the trade-off between data and computational resources, Figure 8.2 plots the KLDs as a function of dataset size and time; further results are provided in Table A.2 of Appendix A.4.2. First, we note that in terms of the final estimates obtained, there is no advantage in running EM with restarts: EM-1-JT and EM-10-JT learn almost identical models. This indicates that the likelihood landscape for MCAR data has few local optima, and is easy to optimize. Hence, EM may be obtaining maximum-likelihood estimates in these cases. In general, maximum-likelihood estimators are more statistically efficient (asymptotically) than other estimators, i.e., they require fewer samples. However, other estimators (such as method-of-moments) can be more computationally efficient; see, e.g., [Was11]. We also observe this trend here. EM obtains better estimates with smaller datasets, with smaller KLDs. However, direct and factored deletion (D-MCAR and F-MCAR) are both orders-of-magnitude faster, and can scale to much larger datasets, than EM (which requires inference). Further, F-MCAR needs only a modest amount of additional data to obtain comparable estimates.

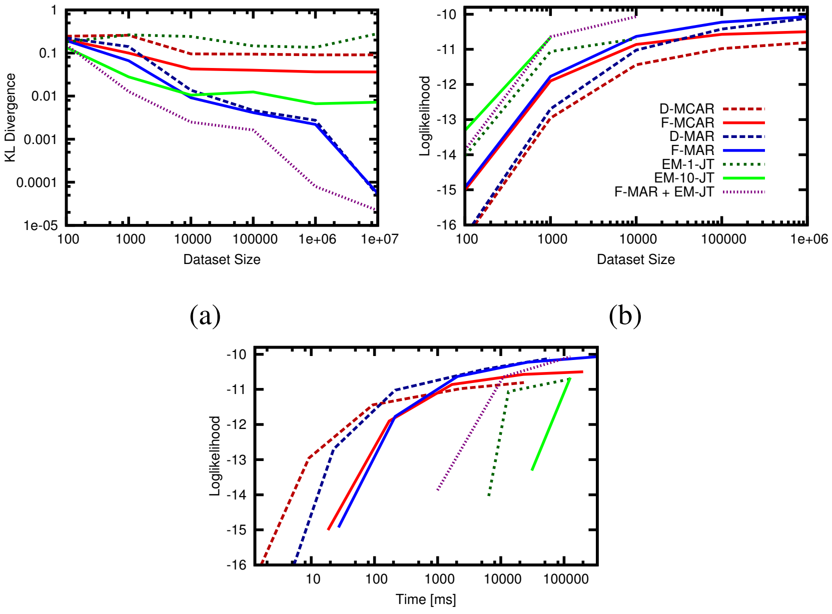
To compare our direct and factored methods, we see that F-MCAR is slower than D-MCAR, as it estimates more quantities (one for each lattice edge). F-MCAR learns better models, however, as it uses a larger part of the available data. Finally, D-MAR performs worse than F-MCAR and D-MCAR, as it assumes the weaker MAR assumption. All learners are consistent, as all KLDs converge to zero.

8.2.2 MAR Data

Next, we consider the more challenging problem of learning from *MAR data*, which we generate as follows: (1) select an m-fraction of the variables to be partially observed, (2) add a missingness mechanism variable R_X for each partially-observed variable X, (3) assign p parents to each R_X, randomly selected from the set of observed variables, giving preference to neighbors of X in the network, (4) sample parameters for the missingness mechanism CPTs from a Beta distribution, (5) sample a complete dataset with R_X values, and (6) hide values of X accordingly.

For our first MAR experiment, we use a small network that is tractable enough for EM to scale to large dataset sizes, so that we can observe trends in this regime. Figure 8.3a shows KLD for the `fire alarm` network, which has only 6 variables (and hence, the complexity of inference is negligible). The missing data mechanisms were generated with $m = 0.3$, $p = 2$, and a Beta distribution with shape parameters 1.0 and 0.5. All numbers are averaged over 64 repetitions with different random learning problems.[3]

[3] On our chosen parameters: (1) the number of repetitions was chosen to produce smooth learning curves; (2) a Beta distribution with shape parameter 1 is uniform, and with parameter 0.5, it is slightly biased (so that it acts more like an MAR, and less like an MCAR, mechanism); (3) $m = 0.3$ corresponds to a low amount of missing data, and later $m = 0.9$ corresponds to high amount; and (4) $p = 2$ encourages sparsity and keeps the CPTs small, although



(a)

(b)

(c)

Figure 8.3: Learning small, tractable Bayesian networks from MAR data. The legend is given in sub-figure (b).

There is a significant difference between EM, with and without restarts, indicating that the likelihood landscape is challenging to optimize (compared to MCAR, which we just evaluated). EM-10-JT performs well for small dataset sizes, but stops converging after around 1,000 instances. This could be due to all restarts getting stuck in local optima. The KLD of F-MAR starts off between EM-1-JT and EM-10-JT for small sizes, but quickly outperforms EM. For the largest dataset sizes, it learns networks whose KLD is two orders of magnitude smaller than EM-10-JT. The KLD improves further when we use F-MAR estimates to seed EM. This approach is on par with EM-10 for small datasets, while still converging for large dataset sizes. However, note that using F-MAR to seed EM will not be practical for larger networks, where inference becomes a bottleneck. D-MCAR and F-MCAR are not consistent for MAR data, and indeed converge to a biased estimate with a KLD around 0.1. Finally, we observe that the factorized algorithms generally outperform their direct counterparts.

For our second MAR experiment, we work with the classical `alarm` network, which has 37 vari-

setting p to 1 or 3 does not change the results.

ables. The missing data mechanisms were generated with $m = 0.9$, $p = 2$, and a Beta distribution with shape parameters 0.5. All reported numbers are averaged over 32 repetitions, and when no number is reported, a 10 minute time limit was exceeded.

Figures 8.3b and 8.3c show test set likelihood as a function of dataset *size* and learning *time*. EM-10-JT performs well for very small dataset sizes, and again outperforms EM-1-JT. However, inference time is non-negligible and EM-10-JT fails to scale beyond 1,000 instances, whereas EM-1-JT scales to 10,000 (as one would expect). The closed-form learners dominate all versions of EM as a function of time, and scale to dataset sizes that are two orders of magnitude larger. EM seeded by F-MAR achieves similar quality to EM-10-JT, while being significantly faster than EM learners with random seeds. D-MAR and F-MAR are more computationally efficient, and can scale to much larger dataset sizes. Further, as seen in Figure 8.3c, they can obtain good likelihoods even before the EM methods report their first likelihoods.

8.2.3 Scaling to Larger Networks

Table 8.2: Log-likelihoods of large networks, with higher treewidths, learned from MAR data (5 min. time limit).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	-57.14	-80.92	-57.01	-80.80	**-56.53**	Water	-19.10	**-18.76**	-25.31	-21.76	-25.29	-21.81
10^3		-	-65.41	-38.54	-30.07	-38.27	**-29.86**		-	**-14.73**	-19.13	-16.45	-18.93	-16.36
10^4		-	-	-25.95	-23.30	-25.36	**-22.88**		-	-20.70	-16.66	-14.90	-16.33	**-14.67**
10^5		-	-	-22.74	-22.01	**-21.60**	-		-	-	-15.49	-	**-14.90**	-
10^2	Munin 1	-	**-103.72**	-115.50	-105.81	-115.41	-104.87	Barley	-	-89.22	-89.54	-89.26	-89.60	**-89.14**
10^3		-	-69.03	-71.01	-65.91	-70.61	**-65.51**		-	-74.26	-71.67	-70.46	-71.68	**-70.18**
10^4		-	-157.23	-56.07	**-54.24**	-55.46	-		-	-	-56.44	**-55.12**	-56.40	-
10^5		-	-	**-52.00**	-	-	-		-	-	-	-	-	-

In our last set of experiments of this section, we evaluate our algorithms on their ability to scale

Table 8.3: Log-likelihoods of large networks, with higher treewidths, learned from MAR data (25 min. time limit).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	**-49.15**	-80.00	-56.45	-79.81	-55.94	Water	-18.88	**-18.73**	-25.84	-22.11	-25.87	-22.25
10^3		-	-53.64	-38.14	-29.32	-37.75	**-29.09**		-17.63	**-14.41**	-18.39	-15.95	-18.27	-15.79
10^4		-	-85.65	-26.21	-23.05	-25.45	**-22.62**		-	-14.52	-15.57	-14.07	-15.24	**-13.92**
10^5		-	-	-22.78	-21.54	-21.60	**-20.79**		-	-24.99	-14.17	-13.46	-13.71	**-13.19**
10^6		-	-	-	-	-	-		-	-	**-13.73**	-	-	-
10^2	Munin 1	-	**-99.15**	-114.76	-106.07	-114.66	-105.12	Barley	-89.05	-89.15	-89.57	-89.17	-89.62	**-89.03**
10^3		-	-67.85	-74.18	-67.81	-73.82	**-67.39**		-	-70.38	-71.86	-70.54	-71.87	**-70.27**
10^4		-	-66.62	-57.50	-54.94	-56.96	**-54.64**		-	-76.48	-56.37	**-55.13**	-56.33	-
10^5		-	-	-53.07	**-51.66**	-52.27	-		-	-	-51.31	-	**-51.19**	-

to larger networks, with higher treewidths, where exact inference is more challenging.[4] Again, inference is the main factor that limits the scalability of algorithms such as EM, to larger networks and datasets (EM invokes inference as a sub-routine, once per data instance, per iteration). Tables 8.2 & 8.3 report results on four networks, where we simulated MAR datasets, as in the previous set of experiments. Each method is given a time limit of 5 or 25 minutes. Appendix A.4.3 provides results on additional settings. We consider the following methods:

EM-JT The EM-10-JT algorithm used in anytime fashion, which returns, given a time limit, the best parameters found in any restart, even if EM did not converge.

EM-BP A variant of EM-JT that uses (loopy) belief propagation for (approximate) inference (in the E-step).

We see that EM-JT, which performs exact inference, does not scale well to these networks. This problem is mitigated by EM-BP, which performs *approximate* inference, yet we find that it also has difficulties scaling (dashed entries indicate that EM-JT and EM-BP did not finish 1 iteration

[4] The `grid` network has 400 variables, `munin1` has 189 variables, `water` has 32 variables, and `barley` has 48 variables.

of EM). In contrast, F-MAR, and particularly D-MAR, can scale to much larger datasets. This efficiency is due to the relative simplicity of the D-MAR and F-MAR estimation algorithms: they are not iterative and require only a single pass over the data. In contrast, with EM-BP, the EM algorithm is not only iterative, but the BP algorithm that EM-BP invokes as a sub-routine, is itself an iterative algorithm. As for accuracy, F-MAR typically obtains the best likelihoods (in bold) for larger datasets, while EM-BP can perform better on smaller datasets. We also evaluated D-MCAR and F-MCAR, although they are not in general consistent for MAR data. We find that they scale even further, and can also produce good estimates in terms of likelihood.

8.3 Exploiting Missingness Graphs

We have so far made very general assumptions about the structure of the missingness graph, capturing the MCAR and MAR assumptions. In this section, we show how to exploit additional knowledge about the missingness graph to further improve the quality of our estimates. Having deeper knowledge of the nature of the missingness mechanisms will even enable us to obtain consistent estimators for datasets that are not MAR (in some cases).

8.3.1 Informed Deletion for MAR

Consider any MAR dataset, and a missingness graph where each $R \in \mathbf{R}$ depends every observed variable in $\mathbf{X}_o$. This would be an MAR missingness graph that assumes the least, in terms of conditional independencies, about the causal mechanisms $\mathbf{R}$. If we know more about the nature of the missingness (i.e., the variables that the $\mathbf{R}$ depend on), we can exploit this to obtain more accurate estimates. Note that knowing the parents of an R is effectively equivalent to knowing the Markov blanket of R [Pea87], which can be learned from data [TAS03, YM05]. With sufficient domain knowledge, an expert may be able to specify the parents of the $\mathbf{R}$. It suffices even to identify a set of variables that just *contains* the Markov blanket.

Suppose that we have such knowledge of the missing data mechanisms of an MAR problem, namely that we know the subset $\mathbf{W}_o$ of the observed variables $\mathbf{X}_o$ that suffice to separate the missing values from their causal mechanisms, i.e., where $\mathbf{X}_m \perp\!\!\!\perp \mathbf{R} \mid \mathbf{W}_o$. We can exploit this

Table 8.4: `alarm` network with Informed MAR data

Size	F-MCAR	D-MAR	F-MAR	ID-MAR	IF-MAR
Kullback-Leibler Divergence					
10^2	**1.921**	2.365	2.364	2.021	2.011
10^3	0.380	0.454	0.452	0.399	**0.375**
10^4	0.073	0.071	0.072	0.059	**0.053**
10^5	0.041	0.021	0.022	0.011	**0.010**
10^6	0.040	0.006	0.008	**0.001**	**0.001**
Test Set Log-Likelihood (Fully Observed)					
10^2	**-11.67**	-12.13	-12.13	-11.77	-11.76
10^3	**-10.40**	-10.47	-10.47	-10.42	**-10.40**
10^4	-10.04	-10.04	-10.04	**-10.02**	**-10.02**
10^5	-10.00	-9.98	-9.98	**-9.97**	**-9.97**
10^6	-10.00	-9.97	-9.97	**-9.96**	**-9.96**

knowledge in our direct deletion algorithm, to obtain improved parameter estimates. In particular, we can reduce the scope of the summation in our direct deletion algorithm from the variables $\mathbf{X}'_o$ (the set of variables in $\mathbf{X}_o$ that lie outside the family $\mathbf{Y}$), to the variables $\mathbf{W}'_o$ (the set of variables in $\mathbf{W}_o$ that lie outside the family $\mathbf{Y}$), yielding the algorithm:

$$\begin{aligned}&Pr(\mathbf{Y})\\ &\approx \sum_{\mathbf{W}'_o} Pr_{\mathfrak{D}}(\mathbf{Y}^\star_m | \mathbf{Y}_o, \mathbf{W}'_o, \mathbf{R}_{\mathbf{Y}_m}{=}0)\, Pr_{\mathfrak{D}}(\mathbf{Y}_o, \mathbf{W}'_o)\end{aligned}$$

Again, we need only consider, in the summation, the instantiations of $\mathbf{W}'_o$ that appear in the dataset. We refer to this algorithm as *informed direct deletion*. By reducing the scope of the summation, we need to estimate fewer sub-terms $Pr_{\mathfrak{D}}(\mathbf{Y}^\star_m | \mathbf{Y}_o, \mathbf{W}'_o, \mathbf{R}_{\mathbf{Y}_m}{=}0)$. This results in a more efficient computation, but further, each individual sub-expression can be estimated on more data. Moreover, our estimates remain consistent. We can similarly replace $\mathbf{X}_o$ by $\mathbf{W}_o$ in the factored deletion algorithm, to obtain an *informed factored deletion* algorithm.

Empirical Evaluation Here, we evaluate the benefits of informed deletion. In addition to the MAR assumption, with this setting, we assume that we know the set of parents $\mathbf{W}_o$ of the missingness mechanism variables. To generate data for such a mechanism, we select a random set of s variables to form $\mathbf{W}_o$. We further employ the sampling algorithm previously used for MAR data, but now insist that the parents of $\mathbf{R}$ variables come from $\mathbf{W}_o$. Table 8.4 shows likelihoods and KLDs on the `alarm` network, for $s = 3$, and other settings as in the MAR experiments. Informed D-MAR (ID-MAR) and F-MAR (IF-MAR) consistently outperform their non-informed counterparts.

8.3.2 Learning From MNAR Data

A missing data problem that is not MAR is classified as MNAR. Here, the parameters of a Bayesian network may not even be identifiable. Further, maximum-likelihood estimation is in general not consistent, so EM and gradient methods can yield biased estimates. However, if one knows the mechanisms that dictate missingness (in the form of a missingness graph), it becomes possible again to obtain consistent estimates, in some cases [MPT13].

For example, consider the missingness graph of Figure 3.2(a), which is an MNAR problem, where both variables X and Y are partially observed, and the missingness of each variable depends on the value of the other. Here, it is still possible to recover $\Pr(X, Y)$ by applying theorem 5. Clearly, procedures for recovering queries under MNAR are extremely sensitive to the structure of the missingness graph.

8.4 Related Work

When estimating the parameters of a Bayesian network, maximum-likelihood (ML) estimation is the typical approach, where for incomplete data, the common wisdom among machine learning practitioners is that one needs to use Expectation-Maximization (EM) or gradient methods [DLR77, Lau95]; see also, e.g., [Dar09, KF09, Mur12, Bar12]. Again, such methods do not scale to large datasets or large networks as (1) they are iterative, (2) they suffer from local optima, and most notably, (3) they require inference in a Bayesian network. Considerable effort has been expended in improving on EM across these dimensions, in order to, for example, (1) accelerate the convergence of EM, and to intelligently sample subsets of a dataset, e.g., [TMH01], (2) escape local optima, e.g., [ENF02], and (3) use approximate inference algorithms in lieu of exact ones when inference is intractable, e.g., [GJ97, CJJ05]. Further, while EM is suitable for data that is MAR (the typical assumption in practice), there are some exceptions, such as work on recommender systems that explicitly incorporate missing data mechanisms [MZ09, MZR07, MZR11].

In the case of complete data, the parameter estimation task simplifies considerably, in the case of Bayesian networks: maximum-likelihood estimates can be obtained inference-free and in closed-form, using just a single pass over the data: $\theta_{x|\mathbf{u}} = \Pr_{\mathcal{D}}(x|\mathbf{u})$. In fact, the estimation algorithms that we proposed in this paper also obtain the same parameter estimates in the case of complete data, although we are not concerned with maximum-likelihood estimation here—we simply want to obtain estimates that are consistent (as in estimation by the method of moments).

Other inference-free estimators have been proposed for other classes of graphical models. [AKN06] identified a method for closed-form, inference-free parameter estimation in factor graphs of bounded degree from complete data. More recently, [HS13] proposed an efficient, inference-free method for consistently estimating the parameters of noisy-or networks with latent variables, under certain structural assumptions. From the perspective of maximum-likelihood learning, where evaluating the likelihood (requiring inference) seems to be unavoidable, the ability to consistently estimate parameters—without the need for inference—greatly extends the accessibility and utility of such models. For example, it opens the door to practical structure learning algorithms, under incomplete data, which is a notoriously difficult problem in practice [AKN06, JHS13].

8.5 Summary

In summary, we developed a family of efficient, and scalable algorithms for computing consistent estimates of the parameters of Bayesian networks, from MCAR and MAR datasets. We further introduced and discussed some improved approaches for parameter estimation, when given information about the m-graph. Empirically, we demonstrated that our algorithms can scale to much larger datasets, and much larger Bayesian networks, than EM.

CHAPTER 9

Concluding Remarks

All methods of missing data analysis rely on assumptions regarding the causes of missingness. Casting these assumptions in a graphical model permits researchers to benefit from the inherent transparency of such models as well as their ability to explicate the statistical implication of the underlying assumptions in terms of conditional independence relations among observed and partially observed variables. We have shown that these features of graphical models can be harnessed to study unchartered territories of missing data research. In particular, we charted the estimability of statistical and causal parameters in broad classes of MNAR problems, and the testability of a model's assumptions under missingness conditions. The testability criteria derived in this paper can be used not only to rule out misspecified models but also to locate specific mis-specifications for the purpose of model updating and re-specification. Testability results are applicable to all problems including MNAR and MAR.

Furthermore, we have identified graphical structures that forbid recovery of parameters given MNAR data. Knowing which sub-structures in the graph prevent recoverability can guide future data collection procedures by pinpointing auxiliary variables that need to be measured to ensure recovery. To overcome non-recoverability, we developed procedures that can in many cases exploit properties of the data to facilitate recovery. Specifically, we have presented necessary and sufficient conditions, based on the technique of matrix inversion, to estimate joint distributions in non-recoverable models. We further developed procedures to recover queries of interest such as causal effects and covariances, in such non-recoverable models under the linearity assumption. Finally, we presented techniques for computing informative bounds on queries of interest.

On the practical side, we have developed suites of algorithms for closed form estimation of Bayesian network parameters from MAR and MCAR data. The empirical results showed that our model-guided procedures yield faster and better quality results compared to state of the art algorithms such as EM.

In the following subsection, we briefly outline future research directions.

9.0.1 Future Directions of Research

Learning m-graphs: In our research we assume that m-graphs are always available and it is true that graphs can be hand-crafted for datasets with few variables. However, in order to facilitate recovery in datasets with large number of variables, it is necessary to design algorithms for learning m-graphs. This is likely to be a challenging problem because conditional independencies involving partially observed variables are almost never verifiable and critical conditional independencies of the form $X \perp\!\!\!\perp R_X|Z$ are not refutable, even given infinitely many samples.

Recoverability given sparse manifest distributions In practice, we rarely find datasets that are strictly positive. Therefore it is necessary to tailor existing algorithms to recover parameters given manifest distributions that are not strictly positive. We demonstrated the possibility in example 16 in chapter 5 and also in appendix A.2.1.

Developing complete algorithms for Recoverability In chapter 3 we presented complete solutions for a broad class of missing data problems. Developing algorithms that are *complete* for recovering causal, probabilistic and counterfactual queries for all missing data problems is still an open problem.

Extending testability results to Verma Constraints The testability results presented in chapter 7 dealt with conditional independence statements. However, causal models also embed functional constraints called Verma Constraints [VP91, TP02b], which are critical for recoverability. Determining whether or not these constraints are testable and devising tests for them whenever they are testable, are open problems in the field.

Developing software packages for handling finite samples Software packages for handling MNAR data are not common. Given the ubiquity of MNAR data, it would be useful to develop theoretically sound procedures for recoverability given *finite* samples. One such possibility is developing model guided imputation procedures.

APPENDIX A

Appendix

A.1 Chapter 2

A.1.1 Testing Compatibility between Underlying and Manifest Distributions

Example 35. *Let the incomplete dataset contain two partially observed variables, Z and W.The tests for compatibility between manifest distribution: $P_m(Z^*, W^*, R_z, R_w)$ and the underlying distribution: $P_u(Z, W, R_z, R_w)$ are:*

Case-1: *Let $X = \{Z, W\}$, then $Y = V_m \setminus X = \{\}$*

$P_m(Z^* = z, W^* = w, R_z = 0, R_w = 0) = P_u(Z = z, W = w, R_z = 0, R_w = 0) \forall z, w$

Case-2: *Let $X = \{Z\}$, then $Y = \{W\}$*

$P_m(Z^* = z, W^* = m, R_z = 0, R_w = 1) = \sum_w P_u(Z = z, w, R_z = 0, R_w = 1) \forall z$

Case-3: *Let $X = \{W\}$, then $Y = \{Z\}$*

$P_m(Z^* = m, W^* = w, R_z = 1, R_w = 0) = \sum_z P_u(z, W = w, R_z = 1, R_w = 0) \forall w$

Case-4: *Let $X = \{\}$, then $Y = \{Z, W\}$*

$P_m(Z^* = m, W^* = m, R_z = 1, R_w = 1) = \sum_{z,w} P_u(z, w, R_z = 1, R_w = 1)$

A.2 Chapter 3

A.2.1 Recoverability when Manifest Distribution is not Strictly Positive

In the following example we describe an instance where joint distribution is recoverable when $P(X^*, Y^*, R_x = 0, R_y = 0) = 0$ for all $X^* = x, Y^* = y$. However,$P(X^*, R_x = 0) > 0$ and $P(Y^* = y, R_y = 0) > 0$.

Example 36. *Consider the m-graph G: $X \rightarrow R_y$ $Y \rightarrow R_x$. Let the query of interest be the joint* *distribution $P(X,Y)$.*

$$\begin{aligned} P(X,Y) &= P(X)P(Y) \textit{ (Using } X \perp\!\!\!\perp Y\textit{)} \\ &= P(X|R_x=0)P(Y|R_y=0) \textit{ (Using } X \perp\!\!\!\perp R_x \textit{ and } Y \perp\!\!\!\perp R_y\textit{)} \\ &= P(X^*|R_x=0)P(Y^*|R_y=0) \textit{ (Using equation 2.1)} \end{aligned}$$

Of note is that even though the manifest distribution is not strictly positive, joint distribution is still estimable, as detailed above.

A.2.2 Heuristics for Finding Admissible Factorization

Consider the task of estimating $Q = P(X)$, where X is a set, by searching for an admissible factorization of $P(X)$ (one that satisfies Theorem 2), possibly by resorting to additional variables, Z, residing outside of X that serve as separating sets. Since there are exponentially large number of ordered factorizations, it would be helpful to rule out classes of non-admissible ordering prior to their enumeration whenever non-admissibility can be detected in the graph. In this section, we provide lemmata that would aid in pruning process by harnessing information from the graph.

Lemma 8. *An ordered set O will not yield an admissible decomposition if there exists a partially observed variable V_i in the order O which is not marginally independent of R_{V_i} such that all minimal separators (refer definition-1) of V_i that d-separate it from R_{v_i} appear before V_i.*

Proof. Let the order be $O = V_1, V_2, V_3, ...V_n$. The factorization corresponding to O is :

$$P(V_1,..,V_n) = \prod_j P(V_j|V_{j+1},...,V_n) = P(V_i|V_{i+1},...V_n)\prod_{j\neq i} P(V_j|V_{j+1},...,V_n)$$

If there is no (minimal) separator S such that $S \subseteq \{V_{i+1},...V_n\}$ then we must have $V_i \not\!\perp\!\!\!\perp R_{V_i}|V_{i+1},...V_n$. Thus we have shown that there exists a term $P(V_i|V_{i+1},\ldots V_n)$ in the factorization that does not satisfy the condition in Theorem-2, thereby making O non-admissible. □

Applying lemma-8 requires a solution to a set of disjunctive constraints which can be represented by directed constraint graphs ([DMP91]).

Example 37. *Let $Q = P(X)$ be the relation to be recovered from the graph in Fig. A.1 (a). Let $X = \{A,B,C,D,E\}$ and $Z = F$. The total number of ordered factorizations is $6! = 720$.*

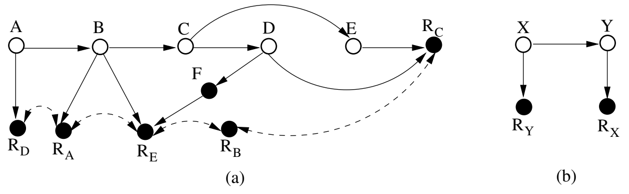


Figure A.1: (a) Depiction of pruning in Example-37 (b) m-graph in which $P(X, Y)$ is recoverable

The independencies implied by minimal separators (as required by Lemma-8) are: $A \perp\!\!\!\perp R_A|B$, $B \perp\!\!\!\perp R_B|\phi$, $C \perp\!\!\!\perp R_C|\{D, E\}$, ($D \perp\!\!\!\perp R_D|A$ or $D \perp\!\!\!\perp R_D|C$ or $D \perp\!\!\!\perp R_D|B$) and ($E \perp\!\!\!\perp R_E|\{B, F\}$ or $E \perp\!\!\!\perp R_E|\{B, D\}$ or $E \perp\!\!\!\perp R_E|C$). To test whether (B,A,D,E,C,F) is potentially admissible we need not explicate all 6 variables; this order can be ruled out as soon as we note that A appears after B. Since B is the only minimal separator that d-separates A from R_A and B precedes A, Lemma-8 is violated. Orders such as (C, D, E, A, B, F), (C, D, A, E, B, F) and (C, E, D, A, F, B) satisfy the condition stated in Lemma 8 and are potential candidates for admissibility.

The following lemma presents a simple test to determine non-admissibility by specifying the condition under which a given order can be summarily removed from the set of candidate orders that are likely to yield admissible factorizations.

Lemma 9. *An ordered set O will not yield an admissible decomposition if it contains a partially observed variable V_i for which there exists no set $S \subseteq V$ that d-separates V_i from R_{V_i}.*

Proof: The factor $P(V_i|V_{i+1}, \ldots, V_n)$ corresponding to V_i can never satisfy the condition required by Theorem 2.

An interesting consequence of Lemma 9 is the following corollary that gives a sufficient condition under which no ordered factorization can be labeled admissible.

Corollary 6. *For any disjoint sets X and Y, there exists no admissible factorization for recovering the relation $P(Y|X)$ by Theorem 2 if Y contains a partially observed variable V_i for which there exists no set $S \subseteq V$ that d-separates V_i from R_{V_i}.*

A.2.3 Proof of theorem 2

Proof. Follows from Theorem 1 noting that ordered factorization is one specific form of decomposition. □

A.2.4 Proof of Theorem 3

Proof. Since the model is Markovian, $P(v)$ may be decomposed as

$$P(v) = \prod_{i,V_i \in V_o} P(v_i|pa_i^o, pa_i^m) \prod_{j,V_j \in V_m} P(v_j|pa_j^o, pa_j^m). \tag{A.1}$$

$R_{Pa_i^m}$ must be non-descendants of V_i, otherwise they will be descendants of Pa_i^m. Therefore $V_i \perp\!\!\!\perp R_{Pa_i^m}|(Pa_i^o \cup Pa_i^m)$. Similarly, R_{V_j} and $R_{Pa_j^m}$ must be non-descendants of V_j and we have $V_j \perp\!\!\!\perp (R_{V_j} \cup R_{Pa_j^m})|(Pa_j^o \cup Pa_j^m)$. Using these conditional independences we obtain Eq. (3.4) from (A.1). □

A.2.5 Proof of Theorem 4

Lemma 10. *$P(X)$ is not recoverable in a m-graph G over (V, U, R) containing a single edge $X \to R_X$.*

Proof. To prove non-recoverability of $P(X)$ we present two models compatible with G:

$$P^{M_1}(v, u, r) = P_1(x, r_X) \prod_{i,V_i \neq X} P(v_i) \prod_j P(u_j) \prod_{k,R_k \neq R_X} P(r_k), \tag{A.2}$$

$$P^{M_2}(v, u, r) = P_2(x, r_X) \prod_{i,V_i \neq X} P(v_i) \prod_j P(u_j) \prod_{k,R_k \neq R_X} P(r_k). \tag{A.3}$$

We construct $P_1(x, r_X)$ and $P_2(x, r_X)$ as given in Table A.1 such that they agree on the observed distributions: $P_1(X, R_X = 0) = P_2(X, R_X = 0) > 0$ and $P_1(R_X = 1) = P_2(R_X = 1) > 0$, but disagree on the query $P_1(X) \neq P_2(X)$.

Then we have that the two models agree on all the observed distributions:

$$P^{M_i}(R_S = 0, R_X = 0, R_{V_m' \setminus S} = 1, x, s, v_o) = P_i(R_X = 0, x)P(R_S = 0, R_{V_m' \setminus S} = 1, s, v_o), \quad i = 1, 2, \tag{A.4}$$

X	R_X	$P_1(X, R_X)$	$P_2(X, R_X)$
0	0	1/3	1/3
1	0	1/3	1/3
0	1	0	1/3
1	1	1/3	0

Table A.1: Two distributions for $X \to R_X$.

and

$$P^{M_i}(R_S = 0, R_X = 1, R_{V'_m \setminus S} = 1, s, v_o) = P_i(R_X = 1)P(R_S = 0, R_{V'_m \setminus S} = 1, s, v_o), \quad i = 1, 2, \tag{A.5}$$

where $V'_m = V_m \setminus \{X\}$ and $S \subseteq V'_m$. But $P^{M_1}(x) = P_1(x)$ disagrees with $P^{M_2}(x) = P_2(x)$. □

Lemma 11. *If a target relation Q is not recoverable in m-graph G, then Q is not recoverable in the graph G' resulting from adding a single edge to G.*

Proof. If Q is not recoverable in G, then there exist two models $P^{M_1}(V, U, R)$ and $P^{M_2}(V, U, R)$ compatible with G decomposed as

$$P^{M_k}(v, u, r) = \prod_i P^{M_k}(v_i|pa_i^v) \prod_j P^{M_k}(u_j|pa_j^u) \prod_l P^{M_k}(r_l|pa_l^r), \quad k = 1, 2, \tag{A.6}$$

such that, for all $S \subseteq V_m$

$$P^{M_1}(R_S = 0, R_{V_m \setminus S} = 1, S, V_o) = P^{M_2}(R_S = 0, R_{V_m \setminus S} = 1, S, V_o) > 0, \tag{A.7}$$

and

$$Q^{M_1} \neq Q^{M_2}. \tag{A.8}$$

For the graph G', we can specify model parameters in such a way that the extra edge added to G is ineffective and hence construct the same distributions as M_1 and M_2. Without loss of generality, assuming G' is obtained from G by adding edge $X \rightarrow V_q$ where X could be a V or U variable. We construct two models M_1' and M_2' compatible with G' with parameters given by

$$P^{M_k'}(v_q|pa_q^v, x) = P^{M_k}(v_q|pa_q^v), \quad k = 1, 2, \tag{A.9}$$

$$P^{M_k'}(v_i|pa_i^v) = P^{M_k}(v_i|pa_i^v), \quad i \neq q, \quad k = 1, 2, \tag{A.10}$$

$$P^{M_k'}(u_j|pa_j^u) = P^{M_k}(u_j|pa_j^u), \quad \forall j, \quad k = 1, 2, \tag{A.11}$$

$$P^{M_k'}(r_l|pa_l^r) = P^{M_k}(r_l|pa_l^r), \quad \forall l, \quad k = 1, 2. \tag{A.12}$$

Clearly $P^{M_k'}(v, u, r) = P^{M_k}(v, u, r), k = 1, 2$. Therefore the two models M_1' and M_2' also satisfy Eqs. (A.7) and (A.8). And we conclude Q is not recoverable in G'. The same arguments apply if G' is obtained from G by adding a parent to U or R variable. □

Non-recoverability of $P(V)$ when X is a parent of R_x has been proved. We will now prove non-recoverability of $P(X)$ and hence $P(V)$ when X and R_x have a latent parent.

Lemma 12. *$P(X)$ is non-recoverable when X and R_x have a latent parent.*

Proof. M_1 and M_2 are two models in which variables U, X and R_x are binary and U is a fair coin. In M_1, $X = 0$ and $R_x = u$ and in M_2, $X = u$ and $R_x = u$. Notice that although the two models agree on the manifest distribution, they disagree on the query $P(X)$. Hence $P(X)$ is non-recoverable in $X < - - U - - > R_x$. Using Lemma-11, we can conclude that $P(V)$ is non-recoverable in any m-graph in which X and R_x are connected by a bi-directed edge. □

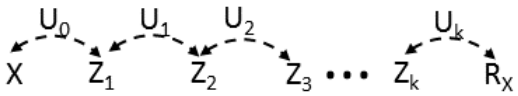


Figure A.2: An m-graph in which $P(X, Z)$ is not-recoverable where $Z = \{Z_1, Z_2, ..., Z_k\}$. X is partially observed, all Z variables are fully observed, parents of Z_i are U_{i-1} and U_i, parent of X is U_o and parent of R_x is U_k.

Lemma 13. *In the m-graph in Figure A.2, $P(X, Z_1, Z_2 ... Z_k)$ is non-recoverable.*

Proof. Let M_3 and M_4 be two models such that all the variables are binary, all the U variables are fair coins, $X = U_0$, $R_x = U_k$ and $Z_i = U_{i-1} \oplus U_i$, $1 \leq i < k$. In M_3, $Z_k = U_{k-1}$ and in M_4,

$Z_k = U_{k-1} \oplus U_k$. Both models yield the same manifest distribution. However, they disagree on the query $P(X, Z_1, Z_2 ... Z_k)$. For instance, in M_3, $P(X = 0, Z = 0, R_x = 1) > 0$ where as in M_4, $P(X = 0, Z = 0, R_x = 1) = 0$. Therefore in M_4, $P(X = 0, Z = 0) = P(X = 0, Z = 0, R_x = 0)$ and in M_3, $P(X = 0, Z = 0) = P(X = 0, Z = 0, R_x = 0) + P(X = 0, Z = 0, R_x = 1)$. Hence in the m-graph in figure A.2, the joint distribution $P(X, Z)$ is non-recoverable. Using lemma 11, we can conclude that joint distribution is non-recoverable in any m-graph which has a bi-directed path from any partially observed variable X to its missingness mechanism R_x. □

A.2.6 Proof of non-recoverability of models in Figure 3.3

Joint Distribution is non-recoverable in Figure 3.3 (a). Let all the substantive variables be binary. Manifest Distribution corresponding to Figure 5.1 (e): $\forall x, y\, P(x^*, y^*, r'_x, r'_y) = \frac{1}{8}$, $P(x^*, y^*, r_x, r_y) = \frac{1}{8}$, $P(x^*, y^*, r_x, r'_y) = 0$, $P(x^*, y^*, r'_x, r_y) = 0$.

Underlying Distributions: In model-1, $\forall x, y\, P(x, y, r_x, r_y) = \frac{1}{8}$ and in model-2 $\forall x, y\, P(x = 0, y = 0, r_x, r_y) = P(x = 0, y = 1, r_x, r_y) = \frac{1}{16}$, $P(x = 1, y = 0, r_x, r_y) = P(x = 1, y = 1, r_x, r_y) = \frac{3}{16}$.

Therefore in model-1 $P(x = 0, y = 0) = \frac{1}{4}$ and in model-2 $P(x = 0, y = 0) = \frac{3}{16}$ □

Joint Distribution is non-recoverable in Figure 3.3 (b). Let all the substantive variables be binary. Manifest Distribution corresponding to Figure 5.1 (f): $\forall y, x, z\, P(y^*, x^*, z, r'_y, r'_x) = \frac{1}{16}$, $P(y^*, x^*, z, r_y, r'_x) = 0$, $P(y^*, x^*, z, r'_y, r_x) = 0$, $P(x^*, y^*, z, r_y, r_x) = \frac{1}{4}$.

Underlying Distributions: In model-1, $\forall y, x, z\, P(y, x, z, r_y, r_x) = \frac{1}{16}$ and in model-2 $\forall x, z\, P(y = 0, x, z, r_y, r_x) = \frac{1}{32}$, $P(y = 1, x, z, r_y, r_x) = \frac{3}{32}$.

Therefore in model-1 $P(x = 0, y = 0, z = 0) = \frac{1}{8}$ and in model-2 $P(x = 0, y = 0, z = 0) = \frac{3}{32}$. □

A.2.7 Proof of Theorem 5

Proof.

$$\begin{aligned} P(V) &= \frac{P(R=0,V)}{P(R=0|V)} \\ &= \frac{P(R=0,V)}{P(R^{(1)}=0,R^{(2)}=0,...R^N=0|V)} \end{aligned}$$

$Mb(R^{(i)})$ d-separates $R^{(i)}$ from all variables that are not in $R^{(i)} \cup Mb(R^{(i)})$ i.e. $R^{(i)} \perp\!\!\!\perp (\{R,V\} - \{R^{(i)}, Mb(R^{(i)})\})|Mb(R^{(i)})$. Hence,

$$P(V) = \frac{P(R=0,V)}{\prod_i P(R^{(i)}=0|Mb(R^{(i)}))}$$

Using $R^{(i)} \cap R_{Mb(R^{(i)})} = \emptyset$ and $R^{(i)} \perp\!\!\!\perp (\{R,V\} - \{R^{(i)}, Mb(R^{(i)})\})|Mb(R^{(i)})$ we get,

$$P(V) = \frac{P(R=0,V)}{\prod_i P(R^{(i)}=0|Mb(R^{(i)}), R_{Mb(R^{(i)})}=0)}$$

Now we can directly apply equation 2.1 and express $P(V)$ in terms of quantities estimable from the available dataset. Therefore, $P(V)$ is recoverable.

Proof of necessity follows from theorem 4. □

A.2.8 Proof of Corollary 2

Proof. Let $|V_m| = 1$ and $Y_1 \in Y$ be the only partially observed variable. Let G' be the subgraph containing all variables in $X \cup Y \cup \{R_{y_1}, Y_1^*\}$. We know that if (1) or (2) are true, then, (i) $P(X,Y)$ is not recoverable in G' and (ii) $P(X)$ is recoverable in G'. Therefore, $P(Y|X) = \frac{P(Y,X)}{P(X)}$ is not recoverable in G' and hence by lemma 11, not recoverable in G. □

A.2.9 Proof of Theorem 6

Proof. $P(Y|do(X)) = \sum_{z,w'} P(Y|Z,W',do(X))P(Z,W'|do(X))$

If condition 1 holds, then by Rule-2 of do-calculus ([Pea09]) we have:

$P(Y|Z,W',do(X)) = P(Y|Z,do(X),do(W'))$

Since $Y \perp_w R_y|Z$,

$$\begin{aligned} P(Y|Z, do(X), do(W')) &= P(Y|Z, do(X), do(W'), R'_y) \\ &= P(Y^*|Z, do(X), do(W'), R'_y) \end{aligned}$$

Therefore, $P(y|do(x))$ is recoverable. □

A.2.10 Proof of Theorem 7

Proof. (sufficiency) Whenever (1) and (2) are satisfied, $Y \perp\!\!\!\perp R_y|V_o$ holds. Hence, $P(V)$ which may be written as $P(Y|V_O)P(V_O)$ can be recovered as $P(Y^*|V_O, R_y = 0)P(V_O)$.

(necessity) follows from theorem 4. □

A.2.11 Proof of Theorem 8

Proof. (sufficiency) Under simple attrition, all paths to R_y from Y containing X are blocked by X. Therefore, when both conditions specified in the theorem are satisfied, it implies that Y and R_y are separable. Given that Z is any separator between Y and R_y, $P(Y|X)$ may be recovered as $\sum_z P(Y^*|X, Z, R'_y)P(Z|X)$.

(necessity) follows from theorem 4 □

A.3 Chapter 5

A.3.1 Proof theorem 10

Proof: Let $G : X \to R_x$ and the augmented model with the ancillary variable W be: $W \to X \to R_x$. Let W and X be binary variables. The manifest distribution $P(X^*, Rx, W)$ is given below.

W	X^*	R_x	$P(W, X^*, R_x)$
0	0	0	0.1
0	1	0	0.2
1	0	0	0.1
1	1	0	0.2
0	m	1	0.2
1	m	1	0.2

$P(W)$ is trivially recoverable since W is fully observed. We will now show that $P(W|X)$ is recoverable.

$P(W|X) = P(W|X, R_x = 0)$, since $W \perp\!\!\!\perp R_x|X$.

$P(W|X, R_x = 0) = P(W^*|X, R_x = 0)$, using equation 2.1. Therefore, $P(W|X) = P(W|X, R_x = 0)$ and

$$M_{WX} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

Obviously, M_{WX} is not invertible.

We prove non-recoverability by presenting $P_1(W, X, R_x)$ and $P_2(W, X, R_x)$ that agree on the manifest distribution $P(W, X^*, R_x)$ but disagree on $P(X)$.

W	X	R_x	$P_1(W, X, R_x)$	$P_2(W, X, R_x)$
0	0	0	0.1	0.1
0	1	0	0.2	0.2
1	0	0	0.1	0.1
1	1	0	0.2	0.2
0	0	1	0.2	0
1	0	1	0.2	0
0	1	1	0	0.2
1	1	1	0	0.2

A.4 Chapter 8

A.4.1 Factored Deletion for MAR

We now give a more detailed derivation of the factored deletion algorithm for MAR data. Let the query of interest be $\Pr(\mathbf{Y})$, and let $\mathbf{X}'_o = \mathbf{X}_m \setminus \mathbf{Y}_m$ and $\mathbf{Z}^i_m = \{Y^j_m | i \leq j \leq n\}$. We can then factorize the estimation of $\Pr(\mathbf{Y})$ as follows.

$$\begin{aligned}
\Pr(\mathbf{Y}) &= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_m, \mathbf{Y}_o, \mathbf{X}'_o) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{Y}_o, \mathbf{X}'_o) \Pr(\mathbf{Y}_m | \mathbf{Y}_o, \mathbf{X}'_o) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}_o) \Pr(\mathbf{Y}_m | \mathbf{X}_o) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}_o) \prod_{i=1}^{n} \Pr\left(Y^i_m | \mathbf{Z}^{i+1}_m, \mathbf{X}_o\right) \\
&= \sum_{\mathbf{X}'_o} \Pr(\mathbf{X}_o) \prod_{i=1}^{n} \Pr\left(Y^i_m | \mathbf{Z}^{i+1}_m, \mathbf{X}_o, \mathbf{R}_{\mathbf{Z}^i_m} = \mathsf{0}\right)
\end{aligned}$$

Table A.2: `alarm` network with MCAR data

Size	EM-1-JT	EM-10-JT	D-MCAR	F-MCAR	D-MAR	F-MAR
Runtime [s]						
10^2	2	6	0	0	0	0
10^3	6	50	0	0	0	0
10^4	69	-	0	1	0	1
10^5	-	-	1	9	4	13
10^6	-	-	11	92	29	124
Test Set Log-Likelihood						
10^2	-12.18	-12.18	-12.85	-12.33	-12.82	-12.32
10^3	-10.41	-10.41	-10.73	-10.55	-10.69	-10.55
10^4	-10.00	-	-10.07	-10.04	-10.07	-10.05
10^5	-	-	-9.98	-9.98	-9.99	-9.98
10^6	-	-	-9.96	-9.96	-9.97	-9.97
Kullback-Leibler Divergence						
10^2	2.381	2.381	3.037	2.525	3.010	2.515
10^3	0.365	0.365	0.688	0.502	0.659	0.502
10^4	0.046	-	0.113	0.084	0.121	0.093
10^5	-	-	0.016	0.013	0.024	0.021
10^6	-	-	0.002	0.002	0.006	0.008

The last step makes use of the MAR assumption. This leads us to the following algorithm, based on the data distribution $\Pr_{\mathfrak{D}}$, and the fully-observed proxy variables $Y_m^{i,\star}$ and $\mathbf{Z}_m^{i+1,\star}$.

$$\begin{aligned} &\Pr(\mathbf{Y}) \\ &\approx \sum_{\mathbf{X}'_o} \Pr_{\mathfrak{D}}(\mathbf{X}_o) \prod_{i=1}^{n} \Pr_{\mathfrak{D}}\big(Y_m^{i,\star} \big| \mathbf{Z}_m^{i+1,\star}, \mathbf{X}_o, \mathbf{R}_{\mathbf{Z}_m^i}{=}0\big) \end{aligned}$$

A.4.2 Extended Empirical Evaluation: MCAR

Table A.2 shows additional results for the classical `alarm` Bayesian network, from Section 8.2.1.

A.4.3 Extended Empirical Evaluation: MAR

In this Appendix, we expand on the empirical results of Section 8.2 w.r.t. learning from MAR data. Here, we provide additional empirical results on standard real-world networks where inference is challenging, as originally highlighted in Table 8.3.

Table A.3: Log-likelihoods of large networks learned from MAR data (1 min. time limit, 1st setting).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	-62.38	-64.15	-50.78	-63.51	**-50.24**	Water	-	-19.50	-20.51	-19.37	-20.41	**-19.35**
10^3		-	-79.75	-38.96	-32.77	-38.26	**-32.44**		-	-16.11	-16.26	-15.27	-16.09	**-15.23**
10^4		-	-	-30.65	-28.61	-30.05	**-28.34**		-	-	-15.03	-14.22	-14.86	**-14.14**
10^5		-	-	-	-	-	-		-	-	**-14.30**	-	-	-
10^2	Munin 1	-	-98.95	-103.59	-98.68	-103.54	**-98.49**	Barley	-	**-85.33**	-85.84	-85.68	-86.13	-85.75
10^3		-	-79.83	-70.49	-67.27	-69.78	**-66.97**		-	-	-67.70	-67.18	-67.67	**-67.13**
10^4		-	-	-59.25	**-57.11**	-	-		-	-	**-54.93**	-	-	-

We consider two settings of generating MAR data, as in Section 8.2. In the *first setting*, the missing data mechanisms were generated with $m = 0.3$, $p = 2$, and a Beta distribution with shape parameters 1.0 and 0.5. In the second setting, we have $m = 0.9$, $p = 2$, and a Beta distribution with shape parameters 0.5 (as in Section 8.2.3). We consider three time limits, of 1 minute, 5 minutes, and 25 minutes. For all combinations of these setting, test set log-likelihoods are shown in Table 8.3, and in Tables A.3 to A.6.

We repeat the observations from the main paper (cf. Section 8.2). The EM-JT learner, which performs exact inference, does not scale well to these networks. This problem is mitigated by EM-BP, which performs *approximate* inference, yet we find that it also has difficulties scaling (dashed entries indicate that EM-JT and EM-BP did not finish 1 iteration of EM). In contrast, F-MAR, and particularly D-MAR, can scale to much larger datasets. As for accuracy, the F-MAR method typically obtains the best likelihoods (in bold) for larger datasets, although EM-BP can perform better on small datasets. We further evaluated D-MCAR and F-MCAR, although they are

Table A.4: Log-likelihoods of large networks learned from MAR data (5 min. time limit, 1st setting).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	-56.23	-63.34	-50.55	-62.38	**-50.06**	Water	-18.84	**-18.06**	-21.23	-19.61	-21.07	-19.57
10^3		-	-55.04	-39.89	-33.34	-39.09	**-33.01**		-	**-14.99**	-16.47	-15.33	-16.24	-15.26
10^4		-	-98.20	-30.46	-27.26	-29.73	**-26.98**		-	-17.39	-15.59	-14.52	-15.26	**-14.43**
10^5		-	-	-28.63	**-26.06**	-27.89	-		-	-	**-15.22**	-	-	-
10^6		-	-	-	-	-	-		-	-	**-15.09**	-	-	-
10^2	Munin 1	-	**-96.51**	-102.51	-98.21	-102.40	-97.95	Barley	-	**-85.59**	-85.70	-85.60	-85.99	-85.66
10^3		-	-68.04	-67.82	-65.49	-67.21	**-65.22**		-	-67.07	-67.58	-66.97	-67.53	**-66.91**
10^4		-	-95.01	-57.68	-56.00	-57.05	**-55.79**		-	-	-55.04	**-54.33**	-54.78	-
10^5		-	-	**-54.30**	-	-	-		-	-	-	-	-	-

Table A.5: Log-likelihoods of large networks learned from MAR data (25 min. time limit, 1st setting).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	**-47.66**	-59.84	-48.34	-59.39	-47.88	Water	-21.30	**-18.66**	-21.58	-19.87	-21.36	-19.83
10^3		-	-46.53	-37.29	-31.60	-36.76	**-31.28**		-17.67	-17.10	-18.64	-15.95	-18.27	**-15.86**
10^4		-	-62.98	-28.74	-26.71	-28.26	**-26.45**		-	-14.83	-16.71	-14.58	-16.30	**-14.44**
10^5		-	-	-25.88	-24.97	-25.43	**-24.75**		-	-18.78	-16.31	-14.38	-15.62	**-14.08**
10^6		-	-	-25.27	-	**-24.78**	-		-	-	**-15.25**	-	-	-
10^7		-	-	-	-	-	-		-	-	**-15.13**	-	-	-
10^2	Munin 1	-	**-90.79**	-98.57	-94.50	-98.48	-94.28	Barley	**-85.11**	-85.53	-86.00	-85.74	-86.24	-85.80
10^3		-	**-60.71**	-66.06	-63.95	-65.45	-63.67		-	**-65.96**	-67.88	-67.23	-67.79	-67.15
10^4		-	-60.35	-56.57	-55.38	-55.95	**-55.16**		-	-57.21	-55.34	-54.56	-55.05	**-54.43**
10^5		-	-	-54.29	**-53.38**	-53.67	-		-	-	**-51.09**	-	-	-

Table A.6: Log-likelihoods of large networks learned from MAR data (1 min. time limit, 2nd setting).

Size		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR		EM-JT	EM-BP	D-MCAR	F-MCAR	D-MAR	F-MAR
10^2	Grid 90-20-1	-	-62.25	-80.10	-56.59	-79.93	**-56.07**	Water	-	**-20.15**	-26.40	-22.85	-26.24	-22.88
10^3		-	-129.38	-38.74	-29.88	-38.51	**-29.70**		-	-17.76	-20.45	-17.80	-20.32	**-17.64**
10^4		-	-	-27.83	-24.30	-27.25	**-23.97**		-	-	-17.59	-15.40	-17.28	**-15.29**
10^5		-	-	-	-	-	-		-	-	**-15.38**	-	-	-
10^2	Munin 1	-	**-99.49**	-111.95	-104.07	-111.72	-103.10	Barley	-	-89.16	-89.63	-89.13	-89.66	**-88.99**
10^3		-	-99.56	-70.32	-66.08	-69.76	**-65.57**		-	-	-71.76	**-70.50**	-71.74	-
10^4		-	-	-56.25	**-54.36**	-	-		-	-	**-56.59**	-	-	-

not in general consistent for MAR data, and find that they scale even further, and can also produce relatively good estimates (in terms of likelihood).

A.4.4 Example: Data Exploitation by Closed-form Estimators

This appendix demonstrates with an example how each learning algorithm exploits varied subsets of data to estimate marginal probability distributions, given the manifest (or data) distribution in Table A.7 which consists of four variables, $\{X, Y, Z, W\}$ such that $\{X, Y\} \in \mathbf{X}_m$ and $\{Z, W\} \in \mathbf{X}_o$.

We will begin by examining the data usage by deletion algorithms while estimating $\Pr(x, w)$ under the MCAR assumption. All three deletion algorithms, namely listwise deletion, direct deletion and factored deletion guarantee consistent estimates when data are MCAR. Among these algorithms, listwise deletion utilizes the least amount of data (4 distinct tuples out of 36 available tuples, as shown in table A.8) to compute $\Pr(xw)$ whereas factored deletion employs two thirds of the tuples (24 distinct tuples out of 36 available tuples as shown in table A.8) for estimating $\Pr(xw)$.

Under MAR, no guarantees are available for listwise deletion. However the three algorithms, namely direct deletion, factored deletion and informed deletion, guarantee consistent estimates. While estimating $\Pr(x, y)$, all the three algorithms utilize every tuple in the manifest distribution

Table A.7: Manifest (Data) Distribution with $\{X, Y\} \in \mathbf{X}_m$ and $\{Z, W\} \in \mathbf{X}_o$.

#	X	Y	W	Z	R_X	R_Y
1	0	0	0	0	0	0
2	0	0	0	1	0	0
3	0	0	1	0	0	0
4	0	0	1	1	0	0
5	0	1	0	0	0	0
6	0	1	0	1	0	0
7	0	1	1	0	0	0
8	0	1	1	1	0	0
9	1	0	0	0	0	0
10	1	0	0	1	0	0
11	1	0	1	0	0	0
12	1	0	1	1	0	0
13	1	1	0	0	0	0
14	1	1	0	1	0	0
15	1	1	1	0	0	0
16	1	1	1	1	0	0
17	0	?	0	0	0	1
18	0	?	0	1	0	1
19	0	?	1	0	0	1
20	0	?	1	1	0	1
21	1	?	0	0	0	1
22	1	?	0	1	0	1
23	1	?	1	0	0	1
24	1	?	1	1	0	1
25	?	0	0	0	1	0
26	?	0	0	1	1	0
27	?	0	1	0	1	0
28	?	0	1	1	1	0
29	?	1	0	0	1	0
30	?	1	0	1	1	0
31	?	1	1	0	1	0
32	?	1	1	1	1	0
33	?	?	0	0	1	1
34	?	?	0	1	1	1
35	?	?	1	0	1	1
36	?	?	1	1	1	1

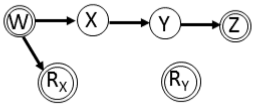
Table A.8: Enumeration of sample # used for computing $\Pr(x, w)$ by listwise deletion, direct deletion and factored deletion algorithms under MCAR assumptions.

Algorithm	Estimator and Sample #
Listwise	$\Pr(xw) = \Pr(xw \mid R_X = 0, R_Y = 0)$ 11,12,15,16
Direct	$\Pr(xw) = \Pr(xw \mid R_X = 0)$ 11,12,15,16,23,24
Factored	$\Pr(xw) = \Pr(x \mid w, R_X = 0)\Pr(w)$ 3,4,7,8,11,12,15,16,19,20,23,24,27,28,31,32,35,36 $\Pr(xw) = \Pr(w \mid x, R_X = 0)\Pr(x \mid R_X = 0)$ 9,10,11,12,13,14,15,16,21,22,23,24

at least once (see Table A.9). Compared to the direct deletion algorithm, the factored deletion algorithm utilizes more data while computing $\Pr(x, y)$ since it has multiple factorizations with more than two factors in each of them; this allows more data to be used while computing each factor (see Table A.8). In contrast to both direct and factored deletion, the informed deletion algorithm yields an estimator that involves factors with fewer elements in them ($\Pr(w)$ vs. $\Pr(zw)$) and hence can be computed using more data ($\Pr(w = 0)$ uses 18 tuples compared to $\Pr(z = 0, w = 0)$ that uses 9 tuples).

Precise information regarding the missingness process is required for estimation when dataset falls under the MNAR category. In particular, only algorithms that consult the missingness graph can answer questions about the estimability of queries.

Table A.9: Enumeration of sample # used for computing $\Pr(x, y)$ by direct deletion, factored deletion and informed deletion algorithms under MAR assumption.

Algorithm	Estimator and Sample #
Direct	$\Pr(xy) = \sum_{z,w} \Pr(xy\|w, z, R_X = 0, R_Y = 0) \Pr(zw)$ 13, 14, 15, 16 for $\Pr(xy\|w, z, R_X = 0, R_Y = 0)$ all tuples: [1,36] for $\Pr(z, w)$
Factored	$\Pr(xy) = \sum_{z,w} \Pr(x\|w, z, y, R_X = 0, R_Y = 0)$ $\Pr(y\|z, w, R_Y = 0) \Pr(zw)$ 13, 14, 15, 16 for $\Pr(x\|y, w, z, R_X = 0, R_Y = 0)$ 5, 6, 7, 8, 13, 14, 15, 16, 29, 30, 31, 32 for $\Pr(y\|w, z, R_Y = 0)$ all tuples: [1,36] for $\Pr(z, w)$ $\Pr(xy) = \sum_{z,w} \Pr(y\|x, w, z, R_X = 0, R_Y = 0)$ $\Pr(x\|z, w, R_X = 0) \Pr(zw)$ 13, 14, 15, 16 for $\Pr(y\|x, w, z, R_X = 0, R_Y = 0)$ 9, 10, 11, 12, 13, 14, 15, 16, 21, 22, 23, 24 for $\Pr(x\|w, z, R_X = 0)$ all tuples: [1,36] for $\Pr(z, w)$
Informed (direct) 	$\Pr(xy) = \sum_{w} \Pr(xy\|w, R_X = 0, R_Y = 0) \Pr(w)$ 13, 14, 15, 16 for $\Pr(xy\|w, R_X = 0, R_Y = 0)$ all tuples: [1,36] for $\Pr(w)$

REFERENCES

[AD96] Silvia Acid and Luis M. De Campos. "An algorithm for finding minimum d-separating sets in belief networks." In *UAI*, pp. 3–10. Morgan Kaufmann Publishers Inc., 1996.

[Ada07] Jon Adams. *Researching complementary and alternative medicine*. Routledge, 2007.

[AKN06] Pieter Abbeel, Daphne Koller, and Andrew Y. Ng. "Learning Factor Graphs in Polynomial Time and Sample Complexity." *Journal of Machine Learning Research*, **7**:1743–1788, 2006.

[All02] Paul D. Allison. "Missing data series: Quantitative applications in the social sciences.", 2002.

[All03] Paul D. Allison. "Missing data techniques for structural equation modeling." *Journal of abnormal psychology*, **112**(4):545, 2003.

[Bar12] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[BGN08] Eric R. Buhi, Patricia Goodson, and Torsten B. Neilands. "Out of sight, not out of mind: strategies for handling missing data." *American journal of health behavior*, **32**:83–92, 2008.

[BMC15] Guy Van den Broeck, Karthika Mohan, Arthur Choi, Adnan Darwiche, and Judea Pearl. "Efficient Algorithms for Bayesian Network Parameter Learning from Incomplete Data." In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 161–170. 2015.

[BP02] Carlos Brito and Judea Pearl. "A new identification condition for recursive models with correlated errors." *Structural Equation Modeling*, **9**(4):459–474, 2002.

[Bri04] Carlos Brito. *Graphical Models for Identification in Structural Equation Models*. PhD thesis, University of California Los Angeles, 2004.

[CD06] Mark Chavira and Adnan Darwiche. "Encoding CNFs to Empower Component Analysis." In *SAT*, pp. 61–74, 2006.

[CD07] Mark Chavira and Adnan Darwiche. "Compiling Bayesian Networks Using Variable Elimination." In *Proceedings of IJCAI*, pp. 2443–2449, 2007.

[CF13] Kuo-Chu Chang and Robert Fung. "Refinement and coarsening of Bayesian networks." *arXiv preprint arXiv:1304.1138*, 2013.

[CJJ05] Brian S. Caffo, Wolfgang Jank, and Galin L. Jones. "Ascent-Based Monte Carlo Expectation-Maximization." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **67**(2):pp. 235–251, 2005.

[CRS06] Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement error in nonlinear models: a modern perspective*. CRC press, 2006.

[CW96] David Roxbee Cox and Nanny Wermuth. *Multivariate dependencies: Models, analysis and interpretation*, volume 67. CRC Press, 1996.

[CW15] Ding-Geng Chen and Jeffrey Wilson. *Innovative Statistical Methods for Public Health Data*. Springer, 2015.

[Dar09] Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge University Press, 2009.

[Daw79] A.P. Dawid. "Conditional independence in statistical theory." *Journal of the Royal Statistical Society, Series B*, **41**(1):1–31, 1979.

[DKC12] Rhian M. Daniel, Michael G. Kenward, Simon N. Cousens, and Bianca L. De Stavola. "Using causal diagrams to guide analysis in missing data problems." *Statistical methods in medical research*, **21**(3):243–256, 2012.

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[DMG08] Jan De Leeuw, Erik Meijer, and Harvey Goldstein. "Handbook of multilevel analysis." 2008.

[DMP91] Rina Dechter, Itay Meiri, and Judea Pearl. "Temporal constraint networks." *Artificial intelligence*, 1991.

[End06] Craig K. Enders. "Analyzing structural equation models with missing data." *Structural equation modeling: A second course*, pp. 313–342, 2006.

[End10] Craig K. Enders. *Applied Missing Data Analysis*. Guilford Press, 2010.

[End11] Craig K. Enders. "Missing not at random models for latent growth curve analyses." *Psychological methods*, **16**(1):1, 2011.

[ENF02] Gal Elidan, Matan Ninio, Nir Friedman, and Dale Shuurmans. "Data Perturbation for Escaping Local Maxima in Learning." In *Proceedings of AAAI*, pp. 132–139, 2002.

[Gar13] Fernando M. Garcia. "Definition and Diagnosis of Problematic Attrition in Randomized Controlled Experiments." Working paper, April 2013. Available at SSRN: http://ssrn.com/abstract=2267120.

[GJ97] Zoubin Ghahramani and Michael I. Jordan. "Factorial Hidden Markov Models." *Machine Learning*, **29**(2-3):245–273, 1997.

[GR97] Richard D. Gill and James M. Robins. "Sequential models for coarsening and missingness." In *Proceedings of the First Seattle Symposium in Biostatistics*, pp. 295–305. Springer, 1997.

[Gra03] John W. Graham. “Adding missing-data-relevant variables to FIML-based structural equation models.” *Structural Equation Modeling*, **10**(1):80–100, 2003.

[Gra09] John W. Graham. “Missing data analysis: Making it work in the real world.” *Annual review of psychology*, **60**:549–576, 2009.

[Gra12] John W. Graham. *Missing Data: Analysis and Design (Statistics for Social and Behavioral Sciences)*. Springer, 2012.

[GS75] Terry C. Gleason and Richard Staelin. “A proposal for handling missing data.” *Psychometrika*, **40**(2):229–252, 1975.

[GVR97] Richard D. Gill, Mark J. Van Der Laan, and James M. Robins. “Coarsening at random: Characterizations, conjectures, counter-examples.” In *Proceedings of the First Seattle Symposium in Biostatistics*, pp. 255–294. Springer, 1997.

[Hai68] Yoel Haitovsky. “Missing data in regression analysis.” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 67–82, 1968.

[Hec77] James J. Heckman. “Sample selection bias as a specification error (with an application to the estimation of labor supply functions).”, 1977.

[HR91] Daniel F. Heitjan and Donald B. Rubin. “Ignorability and coarse data.” *The Annals of Statistics*, pp. 2244–2253, 1991.

[HS13] Yoni Halpern and David Sontag. “Unsupervised Learning of Noisy-Or Bayesian Networks.” In *Proceedings of UAI*, 2013.

[HT52] Daniel G. Horvitz and Donovan J. Thompson. “A generalization of sampling without replacement from a finite universe.” *Journal of the American statistical Association*, **47**(260):663–685, 1952.

[HV12] Yimin Huang and Marco Valtorta. “Pearl’s calculus of intervention is complete.” *arXiv preprint arXiv:1206.6831*, 2012.

[JHS13] Yacine Jernite, Yonatan Halpern, and David Sontag. “Discovering Hidden Variables in Noisy-Or Networks using Quartet Tests.” In *NIPS*, pp. 2355–2363, 2013.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. 2009.

[KP14] Manabu Kuroki and Judea Pearl. “Measurement bias and effect restoration in causal inference.” *Biometrika*, **101**(2):423–437, 2014.

[Lau95] Steffen L. Lauritzen. “The EM algorithm for graphical association models with missing data.” *Computational Statistics and Data Analysis*, **19**:191–201, 1995.

[Lau96] Steffen L. Lauritzen. *Graphical models*, volume 17. Oxford University Press, 1996.

[Lit88] Roderick J. A. Little. “A test of missing completely at random for multivariate data with missing values.” *Journal of the American Statistical Association*, **83**(404):1198–1202, 1988.

[Lit95] Roderick J. A. Little. “Modeling the drop-out mechanism in repeated-measures studies.” *Journal of the American Statistical Association*, **90**(431):1112–1121, 1995.

[LR02] Roderick J. A. Little and Donald B. Rubin. *Statistical analysis with missing data*. Wiley, 2002.

[LR03] Mark J. Van der Laan and James M. Robins. *Unified methods for censored longitudinal data and causality*. Springer Verlag, 2003.

[LSL13] Lingling Li, Changyu Shen, Xiaochun Li, and James M. Robins. “On weighting approaches for missing data.” *Statistical methods in medical research*, **22**(1):14–30, 2013.

[MGG06] Lawrence S Meyers, Glenn Gamst, and Anthony J Guarino. *Applied multivariate research: Design and interpretation*. Sage, 2006.

[MKG01] Geert Molenberghs, Michael G. Kenward, and Els Goetghebeur. “Sensitivity analysis for incomplete contingency tables: the Slovenian plebiscite case.” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **50**(1):15–29, 2001.

[MMS07] Patrick E. McKnight, Katherine M. McKnight, Souraya Sidani, and Aurelio J. Figueredo. *Missing data: A gentle introduction*. Guilford Press, 2007.

[MP14] Karthika Mohan and Judea Pearl. “On the testability of models with missing data.” *Proceedings of AISTAT*, 2014.

[MPT13] Karthika Mohan, Judea Pearl, and Jin Tian. “Graphical Models for Inference with Missing Data.” In *Advances in Neural Information Processing Systems 26*, pp. 1277–1285. 2013.

[Mur12] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[MZ09] Benjamin M. Marlin and Richard S. Zemel. “Collaborative prediction and ranking with non-random missing data.” In *Proceedings of the third ACM conference on Recommender systems*, pp. 5–12. ACM, 2009.

[MZR07] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. “Collaborative filtering and the missing at random assumption.” In *UAI*, 2007.

[MZR11] Benjamin M. Marlin, Richard S. Zemel, Sam T. Roweis, and Malcolm Slaney. “Recommender systems: missing data and statistical model estimation.” In *IJCAI*, 2011.

[New14] Daniel A. Newman. “Missing data: Five practical guidelines.” *Organizational Research Methods*, **17**(4):372–411, 2014.

[Osb12] Jason W. Osborne. *Best practices in data cleaning: A complete guide to everything you need to do before and after collecting your data*. Sage Publications, 2012.

[Osb14] Jason W. Osborne. *Best practices in logistic regression*. SAGE Publications, 2014.

[PE02] Cara Lee Okleshen Peters and Craig K. Enders. "A primer for the estimation of structural equation models in the presence of missing data: Maximum likelihood algorithms." *Journal of Targeting, Measurement and Analysis for Marketing*, **11**(1):81–95, 2002.

[PE04] James L. Peugh and Craig K. Enders. "Missing data in educational research: A review of reporting practices and suggestions for improvement." *Review of educational research*, **74**(4):525–556, 2004.

[Pea87] Judea Pearl. "Evidential reasoning using stochastic simulation of causal models." *AIJ*, **32**(2):245–257, 1987.

[Pea09] Judea Pearl. *Causality: models, reasoning and inference*. Cambridge Univ Press, New York, 2009.

[Pea12] Judea Pearl. "On measurement bias in causal inference." *arXiv preprint arXiv:1203.3504*, 2012.

[Pea13] Judea Pearl. "Linear models: A useful microscope for causal analysis." *Journal of Causal Inference*, **1**(1):155–170, 2013.

[PTP06] Richard F. Potthoff, Gail E. Tudor, Karen S. Pieper, and Vic Hasselblad. "Can one assess whether missing data are missing at random in medical studies?" *Statistical methods in medical research*, **15**(3):213–234, 2006.

[RGL08] Kenneth J. Rothman, Sander Greenland, and Timothy L. Lash. *Modern epidemiology*. Lippincott Williams & Wilkins, 2008.

[RGP11] Noémie Resseguier, Roch Giorgi, and Xavier Paoletti. "Sensitivity analysis when data are missing not-at-random." *Epidemiology*, **22**(2):282, 2011.

[RHB00] James M. Robins, Miguel Angel Hernan, and Babette Brumback. "Marginal structural models and causal inference in epidemiology.", 2000.

[RRS98] Andrea Rotnitzky, James M. Robins, and Daniel O. Scharfstein. "Semiparametric regression for repeated outcomes with nonignorable nonresponse." *Journal of the american statistical association*, **93**(444):1321–1339, 1998.

[RRZ94] James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. "Estimation of regression coefficients when some regressors are not always observed." *Journal of the American statistical Association*, **89**(427):846–866, 1994.

[RRZ95] James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. "Analysis of semiparametric regression models for repeated outcomes in the presence of missing data." *Journal of the American Statistical Association*, **90**(429):106–121, 1995.

[Rub76] Donald B. Rubin. “Inference and missing data.” *Biometrika*, **63**:581–592, 1976.

[Rub87] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley Online Library, New York, NY, 1987.

[Rub96] Donald B. Rubin. “Multiple imputation after 18+ years.” *Journal of the American statistical Association*, **91**(434):473–489, 1996.

[SBC10] Gabriel L. Schlomer, Sheri Bauman, and Noel A. Card. “Best practices for missing data management in counseling psychology.” *Journal of Counseling psychology*, **57**(1):1, 2010.

[Sch02] Judi Scheffer. “Dealing with missing data.” 2002.

[SG02] Joseph L. Schafer and John W. Graham. “Missing data: our view of the state of the art.” *Psychological Methods*, **7**(2):147–177, 2002.

[SK16] Bas van Stein and Wojtek Kowalczyk. “An Incremental Algorithm for Repairing Training Sets with Missing Values.” In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 175–186. Springer, 2016.

[SMP15] Ilya Shpitser, Karthika Mohan, and Judea Pearl. “Missing Data as a Causal and Probabilistic Problem.” In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. 2015.

[SP06] Ilya Shpitser and Judea Pearl. “Identification of Conditional Interventional Distributions.” In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 437–444. 2006.

[SP08] Ilya Shpitser and Judea Pearl. “Dormant Independence.” In *In Proceedings of the Twenty-Fourth Conference on Artificial Intelligence 24*, pp. 1081–1087, 2008.

[Sve15] Oleksandr Sverdlov. *Modern adaptive randomized clinical trials: statistical and practical aspects*. Chapman and Hall/CRC, 2015.

[TAS03] Ioannis Tsamardinos, Constantin F. Aliferis, Alexander R. Statnikov, and Er Statnikov. “Algorithms for Large Scale Markov Blanket Discovery.” In *Proceedings of FLAIRS*, volume 2003, pp. 376–381, 2003.

[TM15] Felix Thoemmes and Karthika Mohan. “Graphical Representation of Missing Data Problems.” *Structural Equation Modeling: A Multidisciplinary Journal*, 2015.

[TMH01] Bo Thiesson, Christopher Meek, and David Heckerman. “Accelerating EM for Large Databases.” *Machine Learning*, **45**(3):279–299, 2001.

[TMM02] Herbert Thijs, Geert Molenberghs, Bart Michiels, Geert Verbeke, and Desmond Curran. “Strategies to fit pattern-mixture models.” *Biostatistics*, **3**(2):245–265, 2002.

[TP02a] Jin Tian and Judea Pearl. “A general identification condition for causal effects.” In *AAAI/IAAI*, pp. 567–573, 2002.

[TP02b] Jin Tian and Judea Pearl. “On the testable implications of causal models with hidden variables.” In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pp. 519–527. Morgan Kaufmann Publishers Inc., 2002.

[TPP98] Jin Tian, Azaria Paz, and Judea Pearl. *Finding minimal d-separators*. Computer Science Department, University of California, 1998.

[TR13] Felix Thoemmes and Norman Rose. “Selection of auxiliary variables in missing data problems: Not all auxiliary variables are created equal.” Technical Report R-002, Cornell University, 2013.

[UB03] Jodie B. Ullman and Peter M. Bentler. *Structural equation modeling*. Wiley Online Library, 2003.

[VP91] Thomas S. Verma and Judea Pearl. “Equivalence and Synthesis of Causal Models.” In *Proceedings of the Sixth Conference in Artificial Intelligence*, pp. 220–227. Association for Uncertainty in AI, 1991.

[Was11] Larry Wasserman. *All of Statistics*. Springer Science & Business Media, 2011.

[WL94] Michae P Wellman and Chao-Lin Liu. “State-space abstraction for anytime evaluation of probabilistic networks.” In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pp. 567–574. Morgan Kaufmann Publishers Inc., 1994.

[Wot00] Werner Wothke. *Longitudinal and multigroup modeling with missing data*. Lawrence Erlbaum Associates Publishers, 2000.

[Wri21] Sewall Wright. “Correlation and causation.” *Journal of agricultural research*, **20**(7):557–585, 1921.

[YM05] Sandeep Yaramakala and Dimitris Margaritis. “Speculative Markov blanket discovery for optimal feature selection.” In *Proceedings of ICDM*, 2005.