Proceedings of Machine Learning Research 77:574–589, 2017 ACML 2017

Recovering Probability Distributions from Missing Data

Jin Tian JTIAN@IASTATE.EDU
Iowa State University

Editors: Yung-Kyun Noh and Min-Ling Zhang

Abstract

A probabilistic query may not be estimable from observed data corrupted by missing values if the data are not missing at random (MAR). It is therefore of theoretical interest and practical importance to determine in principle whether a probabilistic query is estimable from missing data or not when the data are not MAR. We present algorithms that systematically determine whether the joint probability distribution or a target marginal distribution is estimable from observed data with missing values, assuming that the data-generation model is represented as a Bayesian network, known as m-graphs, that not only encodes the dependencies among the variables but also explicitly portrays the mechanisms responsible for the missingness process. The results significantly advance the existing work.

Keywords: Missing Data, Bayesian network

1. Introduction

Missing data occur when some variable values are missing from recorded observations. It is a common problem across many disciplines including artificial intelligence, machine learning, statistics, economics, and the health and social sciences. Missing data pose a major obstacle to valid statistical and causal inferences in a broad range of applications. There is a vast literature on dealing with missing data in diverse fields. We refer to Mohan et al. (2013); Mohan and Pearl (2014a) for a review of related work. Most work in machine learning assumes data are *missing at random (MAR)* Rubin (1976); Little and Rubin (2002), under which likelihood-based inference (as well as Bayesian inference) can be carried out while ignoring the mechanism that leads to missing data.

In principle, however, to analyze data with missing values, we need to understand the mechanisms that lead to missing data, in particular whether the fact that variables are missing is related to the underlying values of the variables in the data set. Indeed some work in machine learning explicitly incorporates missing data mechanism into the model Jaeger (2006); Marlin et al. (2007, 2011). Recently Mohan et al. (2013) have used directed acyclic graphs (DAGs) or Bayesian networks to encode the missing data model, called *m-graphs*, by representing both conditional independence relations among variables and the mechanims responsible for the missingness process. M-graphs provide a general framework for inference with missing data when the MAR assumption does not hold and the data are categorized as *missing not at random (MNAR)*. Whether a DAG model or MAR assumption is testable with missing data is studied in Mohan and Pearl (2014b); Tian (2015). A graphical version of MAR defined in terms of graphical structures is discussed in Tian (2015).

One important research question under this graphical model framework is: Is a target probabilistic query estimable from observed data corrupted by missing values given a missing data model represented as a m-graph? It is known that when the data are MAR, the joint distribution is estimable. On the other hand, when the data are MNAR, a probabilistic query may or may not be estimable depending on the query and the exact missing data mechanisms. For example, consider a single random variable X and assume that whether the values of X are missing is related to the values of X (e.g., in a salary survey, people with low income are less likely to reveal their income). The model is MNAR and we can not estimate $P(X)$ consistently even if infinite amount of data are collected. In practice it is important to determine in principle whether a target query is estimable from missing data or not. Several sufficient conditions have been derived under which probability queries of the form $P(x, y)$ or $P(y|x)$ are estimable Mohan et al. (2013). Mohan and Pearl (2014a) extended those results and further developed conditions for recovering causal queries of the form $P(y|do(x))$. Shpitser et al. (2015) formulated the problem as a causal inference problem and developed a systematic algorithm called MID for estimating the joint distribution.

In this paper we develop algorithms for systematically determining the recoverability of the joint distribution and any target marginal distributions from missing data in m-graphs. The results are significantly more general than the sufficient conditions in Mohan et al. (2013); Mohan and Pearl (2014a). Compared to the MID algorithm in Shpitser et al. (2015) we treat the problem in a purely probabilistic framework without appealing to causality theory, and our algorithm can cover models where MID may fail to work.

The paper is organized as follows. Section 2 defines the notion of m-graphs as introduced in Mohan et al. (2013). Section 3 formally defines the notion of recoverability from missing data and briefly reviews previous work. Section 4 presents some basic results that will facilitate developing algorithms for recovering probability distributions. Section 5 presents our algorithm for recovering the joint distribution and Section 6 presents our algorithm for recovering marginal distributions. Section 7 concludes the paper.

2. Missing Data Model as a Bayesian Network

Bayesian networks are widely used for representing data generation models Pearl (2000); Koller and Friedman (2009). Mohan et al. (2013) used DAGs called *m-graphs*, to represent both the data generation model and the mechanisms responsible for the missingness process. In this section we define m-graphs, mostly following the notation in Mohan et al. (2013).

Let G be a DAG over a set of variables $V \cup R$ where V is the set of observable variables and R is the set of missingness indicator variables introduced in order to represent the mechanisms that are responsible for missingness. In this paper we consider models with no unobserved latent variables, known as Markovian models. We assume that V is partitioned into V_o and V_m such that V_o is the set of variables that are observed in all data cases and V_m is the set of variables that are missing in some data cases and observed in other cases.[1] Every variable $V_i \in V_m$ is associated with a variable $R_{V_i} \in R$ such that, in any observed data case, $R_{V_i} = 1$ if the value of corresponding V_i is missing and $R_{V_i} = 0$ if V_i is observed.

1. We assume we could partition the V variables into V_o and V_m based on domain knowledge (or modeling assumption). In many applications, we have the knowledge that some variables are always observed in all data cases.

We require that R variables may not be parents of variables in V, since R variables are missingness indicator variables and we assume that the data generation process over V variables does not depend on the missingness mechanism. For any set $S \subseteq V_m$, let R_S represent the set of R variables corresponding to variables in S.

The DAG G provides a compact representation of the missing data model $P(V,R) = P(V)P(R|V)$, and will be called a m-graph of the model. The m-graph depicts both the dependency relationships among variables in V and the missingness mechanisms. See Figures 1 and 2 for examples of m-graphs. We use solid circles to represent always observed variables in V_o and R, and hollow circles to represent partially observed variables in V_m.

3. Recoverability from Missing Data

Given a m-graph and observed data with missing values, it is important to know whether we can in principle compute a consistent estimate of a given probabilistic query q (e.g. $P(x)$). If q is deemed to be not estimable (or recoverable) then it is not estimable even if we have collected infinite amount of data. Next we formally define the notion of recoverability.

In any observation, let $S \subseteq V_m$ be the set of observed variables (i.e., values of variables in $V_m \setminus S$ are missing). Then the observed data is governed by the distribution $P(V_o, S, R_{V_m \setminus S} = 1, R_S = 0)$. Formally

Definition 1 (Recoverability) *Given m-graph G, a target probabilistic query q is said to be* recoverable *if q can be expressed in terms of the set of observed positive probabilities $\{P(V_o, S, R_{V_m \setminus S} = 1, R_S = 0) : S \subseteq V_m\}$ - that is, if $q^{M_1} = q^{M_2}$, for every pair of models $P^{M_1}(V,R)$ and $P^{M_2}(V,R)$ compatible with G with $P^{M_1}(V_o, S, R_{V_m \setminus S} = 1, R_S = 0) = P^{M_2}(V_o, S, R_{V_m \setminus S} = 1, R_S = 0) > 0$ for all $S \subseteq V_m$.*

This collection of observed probabilities $\{P(V_o, S, R_{V_m \setminus S} = 1, R_S = 0) : S \subseteq V_m\}$ has been called the *manifest distribution* and the problem of recovering probabilistic queries from the manifest distribution has been studied in Mohan et al. (2013); Mohan and Pearl (2014b,a); Shpitser et al. (2015).

Example 1 *In Fig. 1, the manifest distribution is the collection $\{P(X, Y, R_X = 0, R_Y = 0), P(X, R_X = 0, R_Y = 1), P(Y, R_X = 1, R_Y = 0), P(R_X = 1, R_Y = 1)\}$.*

3.1. Previous work

When data are MAR, it is known that the joint distribution is recoverable. We have $R \perp\!\!\!\perp V_m | V_o$[2] (see Mohan et al. (2013); Tian (2015) for graphical definition of MAR), and the joint is recoverable as $P(V) = P(V_m|V_o)P(V_o) = P(V_m|V_o, R = 0)P(V_o)$.

When data are MNAR, the joint $P(V)$ may or may not be recoverable depending on the m-graph G. Mohan and Pearl (2014a) presented a sufficient condition for recovering probabilistic queries by using sequential factorizations (extending ordered factorizations in Mohan et al. (2013)). The basic idea is to find an order of variables, called *admissible sequence*, in $V \cup R$ such that $P(V)$ could be decomposed into an ordered factorization or sum of it such that every factor is recoverable by using conditional independence relationships.

2. We use $X \perp\!\!\!\perp Y | Z$ to denote that X is conditionally independent of Y given Z.

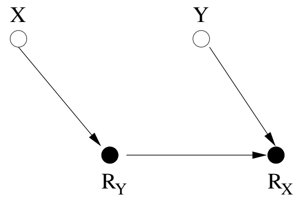

Figure 1: A m-graph that is MNAR. $P(X, Y)$ is recoverable. We use solid circles to represent always observed variables in V_o and R, and hollow circles to represent partially observed variables in V_m.

Example 2 *We want to recover* $P(X, Y)$ *given the m-graph in Fig. 1. The order* $X < R_Y < Y$ *induces the following sum of factorization:*

$$P(x, y) = \sum_{r_Y} P(x|r_Y, y)P(r_Y|y)P(y) = P(y)\sum_{r_Y} P(x|r_Y)P(r_Y), \tag{1}$$

where both $P(y) = P(y|R_Y = 0)$ *and* $P(x|r_Y) = P(x|r_Y, R_X = 0)$ *are recoverable.*

The main issue with the sequential factorization approach is that it is not clear in general whether an admissible sequence exists or how to find an admissible sequence (even deciding whether a given order is admissible or not does not appear to be easy). Several sufficient conditions for recovering the joint $P(V)$ and queries such as $P(x|y)$ are given in Mohan et al. (2013); Mohan and Pearl (2014a) which may handle problems for which no admissible sequence exists. For example, one condition says $P(V)$ is recoverable if no variable X is a parent of its corresponding R_X and there are no edges between the R variables. Shpitser et al. (2015) formulated the problem as a causal inference problem and used techniques developed for the problem of identification of causal effects to develop a general algorithm for recovering the joint $P(V)$. In this paper we develop algorithms for systematically recovering the joint $P(V)$ and marginal distributions. We treat the problem in a purely probabilistic framework without appealing to causality theory.

4. Utility Lemmas

In this section, we present some basic results that will facilitate developing algorithms for recovering probability distributions. We will then use these results to develop algorithms for recovering the joint $P(V)$ and any target marginal distributions. We believe these results provide a foundation for solving other recoverability related tasks (such as recovering a target conditional distribution $P(y|x)$ or causal query $P(y|do(x))$).

First we reformulate the given observed probabilities.

Proposition 2 *Given the manifest distribution* $\{P(V_o, S, R_{V_m \setminus S} = 1, R_S = 0) : S \subseteq V_m\}$, *the set of probabilities* $P^* = \{P(V_o, S, R_{V_m \setminus S}, R_S = 0) : S \subseteq V_m\}$ *are recoverable.*

Proof For any $r_{V_m \setminus S}$ values, let T be the set of variables in $V_m \setminus S$ for which $r_T = 0$, then $r_{V_m \setminus (S \cup T)} = 1$. We have $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ is recoverable as

$$P(v_o, s, r_{V_m \setminus S}, R_S = 0) = \sum_t P(v_o, s \cup t, R_{V_m \setminus (S \cup T)} = 1, R_{S \cup T} = 0) \tag{2}$$

■

It turns out that it is much easier to work with the set of probabilities $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ than with $P(V_o, S, R_{V_m \setminus S} = 1, R_S = 0)$. Therefore in the following, to recover a probabilistic query q, we attempt to express q in terms of the set of observed probabilities $P^* = \{P(V_o, S, R_{V_m \setminus S}, R_S = 0) : S \subseteq V_m\}$.

Example 3 *In Fig. 1, instead of the manifest distribution given in Example 1, we work with the set of observed distributions:*
$P^* = \{P(X, Y, R_X = 0, R_Y = 0), P(X, R_X = 0, R_Y), P(Y, R_X, R_Y = 0), P(R_X, R_Y)\}$.

In order to determine the recoverability of a probabilistic query from the set of observed distributions, the basic idea is to express observed $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ in a "canonical" form of the chain rule factorization of Bayesian networks. For the joint distribution we have

$$P(v) = \prod_i P(v_i | pa_i), \tag{3}$$

and for the observed complete data case, we have

$$P(v_o, v_m, R = 0) = P(v)P(R = 0|v) = \prod_i P(v_i|pa_i) \prod_j P(R_{V_j} = 0 | pa_{r_{V_j}}) \mid_{R=0}, \tag{4}$$

where Pa_i and $Pa_{r_{V_j}}$ represent the parents of V_i and R_{V_j} in G respectively. We obtain that $P(V)$ is recoverable if every factor $P(v_i|pa_i)$ is recoverable (or every factor $P(R_{V_j} = 0|pa_{r_{V_j}})$ is recoverable). In general, for $S \subseteq V_m$ we have

$$P(v_o, s, r_{V_m \setminus S}, R_S = 0) = \sum_{v_m \setminus s} \prod_i P(v_i|pa_i) \prod_j P(r_{V_j}|pa_{r_{V_j}}) \mid_{R_S=0}. \tag{5}$$

We ask the question: given the expression in Eq. (5), is a factor $P(r_{V_j}|pa_{r_{V_j}})$ or $P(v_i|pa_i)$ computable in terms of given $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$? What functions of those factors $P(r_{V_j}|pa_{r_{V_j}})$ and $P(v_i|pa_i)$ are computable in terms of given $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$? It turns out similar questions have been studied in dealing with Bayesian networks with unobserved latent variables in Tian and Pearl (2002a,b). Here we assume the variables in $V_m \setminus S$ are latent variables and extend the results in Tian and Pearl (2002a,b) to our situation. The basic idea is that the summation of products in (5) may be decomposed into a product of summations based on so-called c-components such that each summation is computable from $P(V_o, S, R_{V_m \setminus S}, R_S)$.

In the sequel, we present basic results regarding what quantities are recoverable in terms of $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$ as expressed in Eq. (5). The results are not presented in the framework of m-graphs, rather in the (more general) framework of Bayesian networks with unobserved latent variables (since we believe they are potentially useful in other related problems.). These results (lemmas) will facilitate developing algorithms for recovering probability distributions.

First we introduce some useful concepts mostly following the notation in Tian and Pearl (2002b). Let G be a Bayesian network structure over $O \cup L$ where $O = \{O_1, \ldots, O_n\}$ is the set of observed variables and $L = \{L_1, \ldots, L_{n'}\}$ is the set of unobserved latent variables. We will often use the notation $G(O, L)$ when we want to make it clear which set of variables in G are latent. The observed probability distribution $P(O)$ can be expressed as:

$$P(o) = \sum_l \prod_{\{i|O_i \in O\}} P(o_i|pa_{O_i}) \prod_{\{i|L_i \in L\}} P(l_i|pa_{L_i}), \tag{6}$$

where the summation ranges over all the L variables. The set of latent variables L can be partitioned into a set of *c-components* $H_1, \ldots, H_m$ by assigning two variables L_i and L_j to the same c-component H_k if and only if there exists a path or edge between L_i and L_j in G such that (i)every internal node of the path is in L, *or* (ii) every node in O on the path is head-to-head ($\rightarrow O_i \leftarrow$) Tian and Pearl (2002b). The partition of latent L variables then defines a partition of the observed O variables into *c-components* $S_1, \ldots, S_k$ as follows. If O_i has a latent parent in H_j then O_i is in c-component S_j corresponding to H_j. If O_i has no latent parent, then it forms a c-component $S_j = \{O_i\}$ by itself, and we set the corresponding latent c-component H_j to be an empty set. We will say that the Bayesian network $G(O, L)$ is partitioned into c-components $(S_1, H_1), \ldots, (S_k, H_k)$. For any c-component (S, H) we define the quantity $Q[S, H]$ to denote the following function [3]

$$Q[S, H](o) = \sum_h \prod_{\{i|O_i \in S\}} P(o_i|pa_{O_i}) \prod_{\{i|L_i \in H\}} P(l_i|pa_{L_i}), \tag{7}$$

where the summation ranges over all the latent variables in H. $Q[S, H](o)$ is a function of some subset of variables in O. Note for $S = \{O_i\}$ and $H = \emptyset$, we have $Q[\{O_i\}, \emptyset] = P(o_i|pa_{O_i})$. For convenience, we will often write $Q[S, H](o)$ as $Q[S, H]$.

The importance of the c-components partition lies in that each $Q[S_i, H_i]$, called a *c-factor*, is computable in terms of $P(O)$. The following result is from Tian and Pearl (2002a,b):

Lemma 3 *Assuming a Bayesian network $G(O, L)$ is partitioned into c-components $(S_1, H_1), \ldots, (S_k, H_k)$, we have*

(i) $P(O)$ is decomposed into

$$P(o) = \prod_i Q[S_i, H_i].$$

3. Note Tian and Pearl (2002a,b) defined quantity $Q[S]$ for any set $S \subseteq O$ to denote the following function

$$Q[S](o) = \sum_l \prod_{\{i|O_i \in S\}} P(o_i|pa_{O_i}) \prod_{\{i|L_i \in L\}} P(l_i|pa_{L_i}),$$

where the summation ranges over all the L variables. It can be shown that $Q[S, H] = Q[S, L] = Q[S]$. The reason we introduce a new notation in this paper is because we are considering situations in which the latent variables L vary (we treat $V_m \setminus S$ as latent when considering $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$) and therefore for the same S its corresponding H varies, while in Tian and Pearl (2002a,b), the latent variables L are fixed and given S the corresponding H is fixed.

(ii) Let a topological order over O be $O_1 < \ldots < O_n$, and let $O^{\leq i} = \{O_1, \ldots, O_i\}$ be the set of variables ordered before O_i (including O_i), for $i = 1, \ldots, n$, and $O^{\leq 0} = \emptyset$. Then each $Q[S_j, H_j]$, $j = 1, \ldots, k$, is computable from $P(O)$ and is given by

$$Q[S_j, H_j] = \prod_{\{i|O_i \in S_j\}} P(o_i|o^{\leq i-1}). \tag{8}$$

Example 4 *Assume that X and Y are latent variables in Fig. 1. We have*

$$P(r_X, r_Y) = [\sum_x P(r_Y|x)P(x)][\sum_y P(r_X|r_Y, y)P(y)] = Q[R_Y, X]Q[R_X, Y], \tag{9}$$

and with the topological order being $R_Y < R_X$, $Q[R_Y, X]$ and $Q[R_X, Y]$ are computable from $P(R_X, R_Y)$ as

$$Q[R_Y, X](r_Y) = P(r_Y), \tag{10}$$

$$Q[R_X, Y](r_X, r_Y) = P(r_X|r_Y). \tag{11}$$

In this paper, the observed distributions are in the form of $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$ as expressed in Eq. (5). To utilize Lemma 3 we consider variables in $V_m \setminus S$ as latent variables, and assume that $G(V_o \cup S \cup R, V_m \setminus S)$ is partitioned into c-components $(C_1, H_1), \ldots, (C_m, H_m)$. Then we obtain

$$P(v_o, s, r_{V_m \setminus S}, R_S = 0) = \prod_i Q[C_i, H_i] \mid_{R_S=0} \tag{12}$$

Example 5 *In Fig. 1, we have*

$$\begin{aligned} P(y, r_X, R_Y = 0) &= \sum_x P(r_X|R_Y = 0, y)P(R_Y = 0|x)P(y)P(x) \\ &= P(r_X|R_Y = 0, y)P(y)[\sum_x P(R_Y = 0|x)P(x)] \\ &= P(r_X|R_Y = 0, y)P(y)Q[R_Y = 0, X]. \end{aligned} \tag{13}$$

With X being latent, $G(\{Y, R_X, R_Y\}, \{X\})$ is partitioned into three c-components $(R_X, \emptyset)$, $(Y, \emptyset)$, and (R_Y, X).

In lieu of Lemma 3(ii), we ask a similar question: given the expression in Eq. (12), is a factor $Q[C_i, H_i]$ computable in terms of given $P(v_o, s, r_{V_m \setminus S}, R_S = 0)$? The main difference with the situation in Lemma 3 is that variables in R_S are assuming a fixed value. Next we extend Lemma 3 to the situation that we are not given $P(O)$ but $P(O \setminus S, S = 0)$ for some $S \subset O$. For any set C, let $An(C)$ denote the union of C and the set of ancestors of the variables in C.

Lemma 4 *Assuming a Bayesian network $G(O, L)$ is partitioned into c-components $(S_1, H_1), \ldots, (S_k, H_k)$, we have, for any $S \subseteq O$,*
(i)

$$P(O \setminus S, S = 0) = \prod_i Q[S_i, H_i] \mid_{S=0}. \tag{14}$$

(ii) If $S_j \cap An(S) = \emptyset$, that is, S_j contains no ancestors of S (nor variables in S), then $Q[S_j, H_j] \big|_{S=0}$ is computable from $P(O \setminus S, S = 0)$. In this case, letting a topological order over O be $O_1 < \ldots < O_n$ such that non-acestors of S is ordered after ancesters of S, i.e., $An(S) < O \setminus An(S)$, then $Q[S_j, H_j] \big|_{S=0}$ is given by

$$Q[S_j, H_j] \big|_{S=0} = \prod_{\{i|O_i \in S_j\}} P(o_i|o^{\leq i-1}) \big|_{S=0}. \tag{15}$$

Proof By Lemma 3, (i) holds and each $Q[S_j, H_j] \big|_{S=0}$ can be expressed by Eq. (15). If S_j contains no ancestors of S, then all variables in S_j are ordered after S. As a consequence $S \subseteq O^{\leq i-1}$ and therefore $P(o_i, o^{\leq i-1}) \big|_{S=0}$ is computable from $P(O \setminus S, S = 0)$ by marginalization and each term $P(o_i|o^{\leq i-1}) \big|_{S=0}$ in Eq. (15) is computable from $P(O \setminus S, S = 0)$. ■

Example 6 *Consider the m-graph in Fig. 1. Eq. (12) becomes, for $S = \{Y\}$, Eq. (13). By Lemma 4, c-factors $P(r_X|R_Y = 0, y)$ and $P(y)$ are computable from $P(y, r_X, R_Y = 0)$ because neither of R_X or Y is an ancestor of R_Y. We also obtain that $Q[R_Y = 0, X]$ is recoverable by virtue of both $P(r_X|R_Y = 0, y)$ and $P(y)$ being recoverable.*

Now for $S = \{X\}$ with Y considered a latent variable, we have

$$P(x, r_Y, R_X = 0) = P(r_Y|x)P(x)[\sum_y P(R_X = 0|r_Y, y)P(y)], \tag{16}$$

where none of the three c-factors is computable from $P(x, r_Y, R_X = 0)$ because R_Y, X, and R_X are all ancestors of R_X.

For $S = \emptyset$ with both X and Y considered latent variables, $P(r_X, r_Y)$ can be expressed as in Eq. (9) and both $Q[R_Y, X]$ and $Q[R_X, Y]$ are computable from $P(R_X, R_Y)$ based on Lemma 4 (or 3 as shown in Example 4).

5. Recovering the Joint

Equipped with Lemmas 3 and 4, we are now ready to develop a systematic algorithm for recovering the joint distribution $P(V)$. We have that $P(V)$ is recoverable if every factor $P(v_i|pa_i)$ in Eq. (3) is recoverable. Alternatively, from Eq. (4) $P(V)$ is recoverable if every factor $P(R_{V_j} = 0|pa_{r_{V_j}})$ is recoverable.

There exist simple graphical conditions by which we can quickly recover $P(v_i|pa_i)$ or $P(R_{V_j}|pa_{r_{V_j}})$. For example, the following proposition is straightforward.

Proposition 5 *$P(S|S')$ is recoverable if $S \cup S' \subseteq V_o \cup R$.*

A necessary condition for recovering $P(R_{V_j} = 0|pa_{r_{V_j}})$ is also known.

Proposition 6 *Mohan et al. (2013) $P(R_{V_j} = 0|pa_{r_{V_j}})$ is not recoverable if V_j is a parent of R_{V_j}*

We summarize several easy-to-use recoverability conditions in the following proposition.

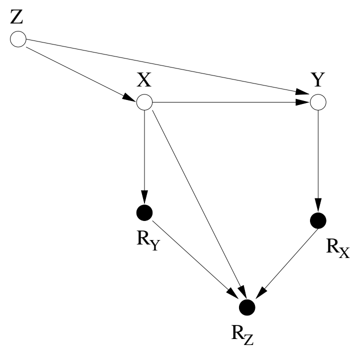

Figure 2: $P(X, Y, Z)$ is recoverable.

Proposition 7 *$P(R_{V_j}|pa^o_{r_{V_j}}, pa^m_{r_{V_j}}, pa^r_{r_{V_j}})$, where $Pa^o_{r_{V_j}}$, $Pa^m_{r_{V_j}}$, and $Pa^r_{r_{V_j}}$ are the parents of R_{V_j} in G that are V^o variables, V^m variables, and R variables respectively, is recoverable if one of the following holds:*

1. *$Pa^m_{r_{V_j}} = \emptyset$.*
2. *$Pa^m_{r_{V_j}}$ is a subset of the set of V_m variables corresponding to $Pa^r_{r_{V_j}}$.*
3. *R_{V_j} has no child.*
4. *None of $R_{Pa^m_{r_{V_j}}}$ is a descendant of R_{V_j}.*

Proof Conditions 1 and 2 are straightforward and used extensively in Mohan et al. (2013); Mohan and Pearl (2014b,a).

Conditions 3 and 4: $P(R_{V_j}|pa^o_{r_{V_j}}, pa^m_{r_{V_j}}, pa^r_{r_{V_j}})$ is a c-factor in $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ for $S = Pa^m_{r_{V_j}}$. Then $P(R_{V_j}|pa^o_{r_{V_j}}, pa^m_{r_{V_j}}, pa^r_{r_{V_j}})$ is recoverable by Lemma 4 since R_{V_j} is not an ancestor of R_S. ■

Based on the conditions 3 and 4 in Proposition 7, we present the following novel sufficient condition for recovering $P(V)$.

Theorem 8 *$P(V)$ is recoverable if no variable X is a parent of its corresponding R_X, and for each R_X, either it has no child, or none of the R variables correponding to its V_m parents are descendants of R_X.*

Example 7 *$P(X, Y, Z)$ is recoverable in Fig. 2 by Theorem 8.*

In general when these simple recoverability conditions are not applicable, we would like to develop a systematic algorithm for recovering the joint $P(V)$. The basic idea is to attempt to recover each $P(v_i|pa_i)$ or $P(R_{V_j}|pa_{r_{V_j}})$ by applying Lemma 4 to observed probabilities $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$. First we study an example.

Example 8 *In the m-graph in Fig. 1, $P(x,y) = P(x)P(y)$ is recovered if both $P(x)$ and $P(y)$ are recovered. $P(y)$ can be recovered from $P(y, r_X, R_Y = 0)$ as shown in Example 6. However $P(x)$ is not computable from $P(x, r_Y, R_X = 0)$ (see Eq. (16)) because X is an ancestor of R_X. On the other hand $Q[R_X, Y]$ in Eq. (16) has been shown to be computable from $P(R_X, R_Y)$ in Example 4. We rewrite Eq. (16) as:*

$$\frac{P(x, r_Y, R_X = 0)}{Q[R_X, Y] \mid_{R_X=0}} = P(r_Y|x)P(x). \tag{17}$$

Now $P(x)$ is computable from the recoverable quantity on the left-hand-side of the above equation as

$$P(x) = \sum_{r_Y} \frac{P(x, r_Y, R_X = 0)}{Q[R_X, Y] \mid_{R_X=0}}. \tag{18}$$

Intuitively, $P(r_Y|x)$ and $P(x)$ are c-factors of the subgraph over $\{R_Y, X\}$ formed by removing the variables R_X and Y from the original m-graph, and both are recoverable by Lemma 3.

In order to recover $P(v_i|pa_i)$ or $P(R_{V_j}|pa_{r_{V_j}})$, we will often need to first recover some other c-factors. We propose a systematic algorithm REQ for recovering an arbitrary c-factor $Q[C, H]$ presented in Fig. 3. REQ works by (i) attempting to recover $Q[C, H]$ from an observed $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ (in which $Q[C, H]$ potentially forms a c-factor) by using algorithm REQP, and (ii) attempting to recover every factor $P(x|pa_X)$ within $Q[C, H] = \sum_h \prod_{X \in C \cup H} P(x|pa_X)$. The algorithm REQP attempts to recover $Q[C, H]$ from a given $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ based on Lemma 4 by systematically reducing the problem to simpler one in subgraphs, e.g. by removing irrelevant non-ancestors or removing other recoverable c-factors. The detailed explanations and motivation of the steps in REQ and REQP are given in the soundness proof of the algorithms (following Theorem 9).

We can recover the joint $P(v)$ by recovering every $P(v_i|pa_i)$ using REQ. However, we observe that it is often the case that to recover all $P(v_i|pa_i)$, we will often first need to recover many $P(R_{V_j} = 0|pa_{r_{V_j}})$ for R_{V_j} being a descendant of some V_i (the opposite is not true). Therefore for the purpose of recovering the joint $P(v)$ it is often more efficient to recover all $P(R_{V_j} = 0|pa_{r_{V_j}})$ instead. Our algorithm REJ-M for recovering the joint $P(V)$ is presented in Fig. 4. REJ-M works by attempting to recover each $P(R_{V_j} = 0|pa_{r_{V_j}})$ using REQ and finally computing $P(v)$ from Eq. (4).

Theorem 9 *Algorithm REJ-M is sound. That is, if REJ-M returns an expression for $P(v)$ then $P(v)$ is correctly recovered.*

Proof From Eq. (4), REJ-M is sound if REQ is sound. Next we show the soundness of REQ.

Step 3 attempts to use simple conditions in Propositions 5-7 to recover $Q[C, H]$.

Step 4 specifies the smallest S such that $Q[C, H]$ could potentially form a c-factor in $G(O, V_m \setminus S)$. Step 5 then attempts to recover $Q[C, H]$ from observed $P(V_o, S, R_{V_m \setminus S}, R_S = 0)$ using algorithm REQP.

```
Procedure REQ($Q[C,H], goals$)
INPUT: m-graph $G$, observed distributions $P^*$, query c-factor $Q[C,H]$,
        $goals$ stores the list of queried c-factors in the recursive calls to avoid loops,
        $proved$: a global variable storing the list of recovered c-factors.
OUTPUT: Expression for $Q[C,H]$ or FAIL

  1. IF $Q[C,H]$ is in $proved$, THEN RETURN $Q[C,H]$ expression.

  2. IF $Q[C,H]$ is in $goals$, THEN RETURN FAIL; ELSE add $Q[C,H]$ to $goals$.

  3. IF any of Propositions 5-7 is applicable, THEN add $Q[C,H]$ to $proved$ and RETURN
     expression for $Q[C,H]$.

  4. Assume that $Q[C,H]$ is a function over $W$. Let $S = W \cap V^m$, $O = V_o \cup S \cup R$.

  5. CALL REQP($G(O, V_m \setminus S), P(V_o, S, R_{V_m \setminus S}, R_S = 0), Q[C,H], goals$).
     IF $Q[C,H]$ is recovered, add $Q[C,H]$ to $proved$ and RETURN $Q[C,H]$ expression.

  6. IF $H$ is empty, RETURN FAIL; ELSE for every $P(x|pa_X), X \in C \cup H$,
           CALL REQ($P(x|pa_X), goals$)
           IF $P(x|pa_X)$ is not recovered, RETURN FAIL
     Add $Q[C,H]$ to $proved$ and RETURN $Q[C,H] = \sum_h \prod_{X \in C \cup H} P(x|pa_X)$.


Procedure REQP($G', P', Q[C,H], goals$)
INPUT: Bayesian network $G'(O,L)$, distribution $P'(O \setminus S, S = 0)$, query c-factor $Q[C,H]$,
        list of queried c-factors $goals$.
OUTPUT: Expression for $Q[C,H]$ or FAIL

  1. IF $(C,H)$ forms a c-component in $G'$ and $C \cap An(S) = \emptyset$, THEN RETURN $Q[C,H]$
     recoverable as given in Lemma 4.

  2. Let $T = (An(C) \cup An(S)) \cap O$ and $D = O \setminus T$. IF $D \neq \emptyset$, THEN let $G''$ be the graph
     resulting from removing $D$ from $G'$ and RETURN REQP($G'', \sum_D P', Q[C,H], goals$).

  3. For each c-component $(C_i, H_i)$ $i = 1, \ldots, k$ of $G'$ such as $C_i \cap An(S) = \emptyset$: $Q[C_i, H_i]$ is
     recovered by Lemma 4. Let $G''$ be the graph resulting from removing all $C_i, H_i, i =$
     $1, \ldots, k$ from $G'$. RETURN REQP($G'', P' / \prod_{i=1}^{k} Q[C_i, H_i], Q[C,H], goals$).

  4. For each c-component $(C_i, H_i)$ of $G'$ that does not contain $C$:
        IF $Q[C_i, H_i]$ is recovered by REQ($Q[C_i, H_i], goals$), THEN let $G''$ be the graph
        resulting from removing $C_i, H_i$ from $G'$ and
        RETURN REQP($G'', P'/Q[C_i, H_i], Q[C,H], goals$).

  5. RETURN FAIL
```

Figure 3: Algorithm for recovering a c-factor.

Algorithm REJ-M
INPUT: m-graph G, observed distributions P^*
OUTPUT: Expression for the joint $P(V)$ or FAIL

```
1. For every $V_j \in V_m$:
       call REQ($P(R_{V_j} = 0|pa_{r_{V_j}}), [\,]$). IF FAIL, THEN RETURN FAIL
2. RETURN $P(v)$ recovered as $P(v) = P(v,R)/\prod_j P(R_{V_j}|pa_{r_{V_j}})\,|_{R=0}$.
```

Figure 4: Algorithm for recovering the joint $P(V)$.

Step 6 attempts to recover every factor $P(x|pa_X)$ within $Q[C,H] = \sum_h \prod_{X\in C\cup H} P(x|pa_X)$. If all factors $P(x|pa_X)$ are recovered, then $Q[C,H]$ is recovered.

Next we show the soundness of REQP which attempts to recover $Q[C,H]$ from observed $P(V_o, S, R_{V_m\setminus S}, R_S = 0)$.

Step 1 is the base case which is sound based on Lemma 4.

In Step 2, summing out D from both sides of Eq. (12) is graphically equivalent to removing variables in D, based on the chain rule of Bayesian networks since all D variables could be ordered after T variables.

In Step 3 we attempt to recover every c-factor $Q[C_i,H_i]$ recoverable from $P(V_o, S, R_{V_m\setminus S}, R_S = 0)$ by Lemma 4. In Step 4 we attempt to recover c-factor $Q[C_i,H_i]$ from observed distributions P^* by calling REQ. If $Q[C_i,H_i]$ is recovered, then given

$$P(o, S=0) = Q[C_i,H_i]\prod_{j\neq i} Q[C_j,H_j]\,|_{S=0}, \tag{19}$$

we obtain

$$\frac{P(o,S=0)}{Q[C_i,H_i]} = \prod_{j\neq i} Q[C_j,H_j]\,|_{S=0}. \tag{20}$$

Now the problem of recovering $Q[C,H]$ is reduced to a problem of recovering $Q[C,H]$ from distribution $P(o,S=0)/Q[C_i,H_i]$ in the subgraph resulting from removing C_i, H_i from G'.

We fail to recover $Q[C,H]$ if $Q[C,H]$ cannot be recovered by Lemma 4 in Step 1 and the problem cannot be reduced to a smaller one by Steps 2, 3, or 4. ■

Example 9 *We demonstrate algorithm REJ-M by recovering $P(A,B,C,D)$ in the m-graph in Fig. 5(a). REJ-M calls for recovering both $P(R_A = 0|d,b,R_B = 0)$ and $P(R_B = 0|d,a)$ using REQ. $P(R_A = 0|d,b,R_B = 0)$ is easily recovered by condition 2 or 3 in Proposition 7. We have $S = \{A\}$ in Step 4 in REQ($P(R_B = 0|d,a), [\,]$), which attempts to recover $P(R_B = 0|d,a)$ from $P(c,d,a,r_B,R_A = 0)$ using REQP. We have*

$$P(c,d,a,r_B,R_A=0) = P(d|c)P(a)P(r_B|d,a)[\sum_b P(b)P(c|a,b)P(R_A=0|d,b,r_B)]. \tag{21}$$

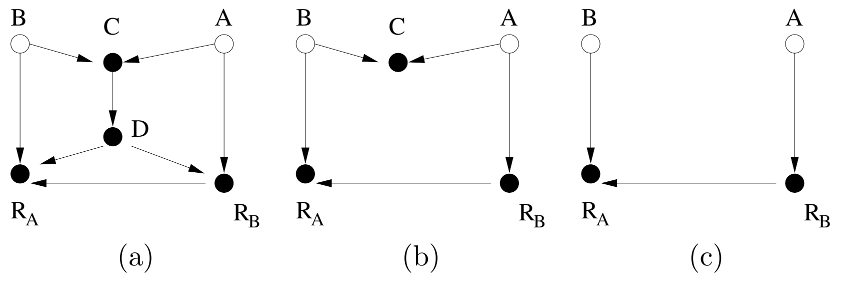

Figure 5: $P(A, B, C, D)$ is recoverable in the m-graph in (a).

Step 1 of REQP is not applicable as R_B is an ancestor of R_A. Steps 2 and 3 are not applicable either. In Step 4 of REQP, REQ($P(d|c)$, goals) recovers $P(d|c)$ by Proposition 5. The problem is reduced to using REQP to recover $P(R_B = 0|d, a)$ in Fig. 5(b) from

$$\frac{P(c, d, a, r_B, R_A = 0)}{P(d|c)} = P(a)P(r_B|d, a)[\sum_b P(b)P(c|a, b)P(R_A = 0|d, b, r_B)]. \tag{22}$$

C is not an ancestor of R_B or R_A in Fig. 5(b), and Step 2 of REQP reduces the problem to using REQP to recover $P(R_B = 0|d, a)$ in Fig. 5(c) from

$$\sum_c \frac{P(c, d, a, r_B, R_A = 0)}{P(d|c)} = P(a)P(r_B|d, a)[\sum_b P(b)P(R_A = 0|d, b, r_B)]. \tag{23}$$

Fig. 5(c) is the same as Fig. 1 for which $P(r_B|d, a)$ can be recoverable as shown in Example 8. In fact, in Step 4 of REQP, REQ($Q[R_A = 0, B]$, goals) will recover $Q[R_A = 0, B] = \sum_b P(b)P(R_A = 0|d, b, r_B)$ by using REQP to recover it from $P(c, d, r_A, r_B)$ from the following

$$\sum_c \frac{P(c, d, r_A, r_B)}{P(d|c)} = [\sum_a P(a)P(r_B|d, a)][\sum_b P(b)P(R_A|d, b, r_B)], \tag{24}$$

from which $Q[R_A, B]$ is recoverable by Lemma 4 (or 3). Finally the problem is reduced to recovering $P(R_B = 0|d, a)$ from the following

$$\frac{1}{Q[R_A = 0, B]} \sum_c \frac{P(c, d, a, r_B, R_A = 0)}{P(d|c)} = P(a)P(r_B|d, a), \tag{25}$$

from which $P(R_B = 0|d, a)$ is recoverable by Lemma 4.

This example is used to demonstrate the MID algorithm in Shpitser et al. (2015), which is the most general algorithm in the literature for recovering the joint. However, there exist examples (Figure 1a in Shpitser and Robins (2016)) where MID fails to recover a recoverable joint [4], while we have tested that our algorithm REJ-M is able to recover the joint in Figure 1a in Shpitser and Robins (2016) (due to the space limitation, we will not show the derivation process in this paper).

4. Shpitser and Robins (2016) claims they have an improved algorithm over MID that can cover the example, but does not present it in their article.

```
Algorithm REM-M
INPUT: m-graph $G$, observed distributions $P^*$, query distribution $P(C)$
OUTPUT: Expression for $P(C)$ or FAIL

1. Let the c-components of $G(C, An(C) \setminus C)$ be $(A_1, B_1), \ldots, (A_k, B_k)$.

2. For every $Q[A_i, B_i]$:
       Call REQ($Q[A_i, B_i]$,[ ]). IF FAIL, THEN RETURN FAIL.

3. RETURN $P(C)$ recoverable as $P(c) = \prod_i Q[A_i, B_i]$
```

Figure 6: Algorithm for recovering marginal distribution $P(C)$ for $C \subset V$.

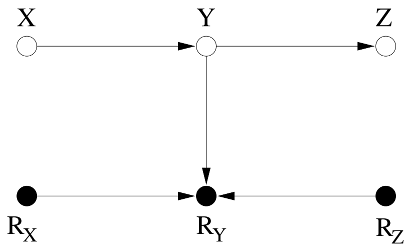

Figure 7: $P(X, Z)$ is recoverable although $P(X, Y, Z)$ is not recoverable.

6. Recovering Marginal Distributions

In this section we present a systematic algorithm for recovering an arbitrary marginal distribution $P(C)$ for $C \subset V$. The basic idea is again to express $P(C)$ in its canonical form of c-factor factorization. We have

$$P(c) = \sum_{v \setminus c} \prod_i P(v_i|pa_i) = \sum_{an(C) \setminus c} \prod_{i, V_i \in An(C)} P(v_i|pa_i) = \prod_i Q[A_i, B_i], \tag{26}$$

where we assume $An(C) \setminus C$ are latent variables and $G(C, An(C) \setminus C)$ is partitioned into a set of c-components (A_i, B_i). We obtain that $P(C)$ is recoverable if each c-factor $Q[A_i, B_i]$ is recoverable. The result is summarized as a systematic algorithm REM-M for recovering $P(C)$ presented in Fig. 6.

Example 10 *We demonstrate algorithm REM-M by recovering $P(X, Z)$ in the m-graph in Fig. 7. Note that since R_Y is a child of Y, the joint $P(X, Y, Z)$ is not recoverable. Following REM-M we have*

$$P(x, z) = P(x) \sum_y P(z|y) P(y|x) \tag{27}$$

We call REQ attempting to recover $P(x)$ and $Q[Z, Y] = \sum_y P(z|y)P(y|x)$ respectively. $P(x)$ is recovered by calling REQP to recover $P(x)$ from $P(x, r_Y, r_Z, R_X = 0)$ and then *using Lemma 4. REQ calls REQP attempting to recover $Q[Z, Y]$ from*

$$P(x, z, r_Y, R_X = 0, R_Z = 0)$$
$$= P(x)[\sum_y P(z|y)P(y|x)P(r_Y|y, R_X = 0, R_Z = 0)]P(R_X = 0)P(R_Z = 0). \tag{28}$$

Since R_Y is not an ancestor of other variables, Step 2 of REQP reduces the problem to recovering $Q[Z, Y]$ from

$$P(x, z, R_X = 0, R_Z = 0) = P(x)Q[Z, Y]P(R_X = 0)P(R_Z = 0), \tag{29}$$

which says $Q[Z, Y]$ is recoverable by Lemma 4.

We believe REM-M is the first systematic algorithm for recovering marginal distributions. It is a significant advance over the existing sufficient conditions in the literature Mohan et al. (2013); Mohan and Pearl (2014a).

7. Conclusion

It is of theoretical interest and importance in practice to determine in principle whether a probabilistic query is estimable from missing data or not when the data are not MAR. In this paper we present algorithms for systematically determining the recoverability of the joint distribution and marginal distributions from observed data with missing values given an m-graph. We have also developed new simple sufficient conditions that could be used to quickly recover the joint. It is natural to ask whether the algorithms REJ-M and REM-M are complete, that is whether the output of FAIL corresponds to that $P(V)$ or $P(C)$ is not recoverable. We are not able to answer this difficult question at this point. We find the algorithms promising in that they have pinned down situations in which recoverability seems not possible. The results are significantly more general than the sufficient conditions in Mohan et al. (2013); Mohan and Pearl (2014a). Compared to the MID algorithm in Shpitser et al. (2015), we treat the problem in a purely probabilistic framework without appealing to causality theory. There also exist models where our REJ-M algorithm works but MID fails. Future work includes investigating the completeness of REJ-M and REM-M, and developing algorithms for recovering arbitrary probabilistic queries such as $P(x|y)$, and algorithms for recovering causal queries such as $P(y|do(x))$. We believe the results in Section 4 provide a solid foundation for pursuing these tasks. It is also an interesting research direction how to actually estimate distribution parameters from finite amount of data if the joint is determined to be recoverable den Broeck et al. (2015).

Acknowledgments

This research was partly supported by NSF grant IIS-1704352.

References

G. Van den Broeck, K. Mohan, A. Choi, A. Darwiche, and J. Pearl. Efficient algorithms for bayesian network parameter learning from incomplete data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Manfred Jaeger. The ai&m procedure for learning from incomplete data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 225–232, 2006.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data.* Wiley, 2002.

Benjamin Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. Collaborative filtering and the missing at random assumption. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

Benjamin M Marlin, Richard S Zemel, Sam T Roweis, and Malcolm Slaney. Recommender systems, missing data and statistical model estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

K. Mohan and J. Pearl. Graphical models for recovering probabilistic and causal queries from missing data. In M. Welling, Z. Ghahramani, C. Cortes, and N. Lawrence, editors, *Advances of Neural Information Processing 27 (NIPS)*, pages 1520–1528. 2014a.

K. Mohan and J. Pearl. On the testability of models with missing data. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2014b.

Karthika Mohan, Judea Pearl, and Jin Tian. Graphical models for inference with missing data. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

I. Shpitser and J. M. Robins. Towards a complete identification algorithm for missing data problems. In *NIPS 2016 "What If" Workshop*, 2016. URL http://www.homepages.ucl.ac.uk/~ucgtrbd/whatif/Paper10.pdf.

I. Shpitser, K. Mohan, and J. Pearl. Missing data as a causal and probabilistic problem. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

J. Tian. Missing at random in graphical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

J. Tian and J. Pearl. A general identification condition for causal effects. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 567–573, Menlo Park, CA, 2002a. AAAI Press/The MIT Press.

J. Tian and J. Pearl. On the testable implications of causal models with hidden variables. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002b.