Fast causal inference with non-random missingness by test-wise deletion

Eric V. Strobl1 · Shyam Visweswaran1 · Peter L. Spirtes2

Received: 24 May 2017 / Accepted: 28 December 2017 / Published online: 19 January 2018
© Springer International Publishing AG, part of Springer Nature 2018

Abstract
Many real datasets contain values missing not at random (MNAR). In this scenario, investigators often perform list-wise deletion, or delete samples with any missing values, before applying causal discovery algorithms. List-wise deletion is a sound and general strategy when paired with algorithms such as FCI and RFCI, but the deletion procedure also eliminates otherwise good samples that contain only a few missing values. In this report, we show that we can more efficiently utilize the observed values with test-wise deletion while still maintaining algorithmic soundness. Here, test-wise deletion refers to the process of list-wise deleting samples only among the variables required for each conditional independence (CI) test used in constraint-based searches. Test-wise deletion therefore often saves more samples than list-wise deletion for each CI test, especially when we have a sparse underlying graph. Our theoretical results show that test-wise deletion is sound under the justifiable assumption that none of the missingness mechanisms causally affect each other in the underlying causal graph. We also find that FCI and RFCI with test-wise deletion outperform their list-wise deletion and imputation counterparts on average when MNAR holds in both synthetic and real data.

Keywords Causal inference · Missing values · Missing not at random · MNAR

1 The problem

Many real observational datasets contain missing values, but modern constraint-based causal discovery (CCD) algorithms require complete data. These facts force many investigators to either perform list-wise deletion or imputation on their datasets. The first strategy can unfortunately result in the loss of many good samples just because of a few missing values. On the other hand, the second strategy can corrupt the joint distribution when the corresponding assumptions do not hold. Both of these approaches therefore can (and often do) degrade the performance of CCD algorithms. We thus seek a practical method which allows CCD algorithms to efficiently utilize the measured values while placing few assumptions on the missingness mechanism(s).

We specifically choose to tackle the most general case of values missing not at random (MNAR), where missing values may depend on other missing values. MNAR stands in contrast to values missing at random (MAR), where missing values can only depend on the measured values. MAR thus ensures recoverability of the underlying distribution from the measured values alone. In causal discovery, investigators usually deal with MAR by performing imputation and then running a CCD algorithm on the completed data [15,16]. Causal discovery under MAR therefore admits a straightforward solution, once an investigator has access to a sound imputation method.

Causal discovery under MNAR requires a more sophisticated approach than causal discovery under MAR. Investigators have developed three general strategies for handling the MNAR case thus far. The first approach assumes access to some background knowledge for modeling the missingness mechanism, typically encoded using graphs [4,10,14]. Investigators with deep knowledge about the dataset at hand can therefore use this strategy to convert the MNAR problem into a more manageable form. However, access to background knowledge is arguably scarce in many situations or prone to error. The second solution involves placing an extra...
assumption on the missingness mechanism(s) so that we may combine the results of multiple runs of a CCD algorithm; in particular, we assume that a dataset with missing values can be decomposed into multiple datasets with potentially non-overlapping variables subject to the same set of selection variables [22–25]. The problem of missing values therefore reduces to a problem of combining multiple datasets. Investigators nevertheless often find the assumption of identical selection bias across datasets hard to justify in practice. The third most general solution involves running a CCD algorithm that can handle selection bias on a list-wise deleted dataset, where investigators remove samples that contain any missing values [17]. List-wise deletion is nonetheless sample inefficient, because it eliminates samples with only a mild number of missing values. We therefore conclude that the three aforementioned strategies for the MNAR case can carry unsatisfactory limitations in real situations.

In this report, we propose to handle the MNAR case in CCD algorithms using a different strategy involving test-wise deletion. Here, test-wise deletion refers to the process of only performing list-wise deletion among the variables required for each conditional independence (CI) test. We develop the test-wise deletion procedure in detail throughout this report as follows. First, we provide background material in Sect. 2. We then characterize missingness using graphical models as follows. First, we develop the CCD algorithms using a different strategy involving test-wise deletion procedure in detail throughout this report. We develop the CCD algorithms using a different strategy involving test-wise deletion procedure in detail throughout this report. We develop the CCD algorithms using a different strategy involving test-wise deletion procedure in detail throughout this report. We develop the CCD algorithms using a different strategy involving test-wise deletion procedure in detail throughout this report. We develop the CCD algorithms using a different strategy involving test-wise deletion procedure in detail throughout this report.

Next, we justify test-wise deletion in Sects. 5 and 7 under the assumption that certain sets of missingness indicators do not causally affect each other in the underlying causal graph. These results lead to our final solution in Sect. 8. We also list experimental results in Sect. 9 which highlight the benefits of the Fast Causal Inference (FCI) algorithm and the Really Fast Causal Inference (RFCI) algorithm with test-wise deletion as opposed to the same algorithms with list-wise deletion or imputation. Finally, we conclude the paper with a short discussion in Sect. 10.

2 The causal interpretation of graphs

Let italicized capital letters such as A denote a single variable and bolded as well as italicized capital letters such as A denote a set of variables (unless specified otherwise). We write A = 1 to indicate that all members of A are set to one. We will also use the terms “variables” and “vertices” interchangeably. We assume that the reader is familiar with basic graphical terminology, which we cover briefly in Appendix 11.1.

We will interpret directed acyclic graphs (DAGs) in a causal fashion. To do this, we consider a stochastic causal process with a distribution P over X that satisfies the Markov property. A distribution satisfies the Markov property if it admits a density that “factorizes according to the DAG” as follows:

\[
f(X) = \prod_{i=1}^{p} f(X_i | Pa(X_i)).
\]

We can in turn relate (1) to a graphical criterion called d-connection. Specifically, if G is a directed graph in which A, B and C are disjoint sets of vertices in X, then A and B are d-connected by C in the directed graph G if and only if there exists an active path \( \pi \) between some vertex in A and some vertex in B given C. An active path between A and B given C refers to an undirected path \( \pi \) between some vertex in A and some vertex in B such that, for every collider \( X_i \) on \( \pi \), a descendant of \( X_i \) is in C and no non-collider on \( \pi \) is in C. A path is inactive when it is not active. Now A and B are d-separated by C in G if and only if they are not d-connected by C in G. For shorthand, we will write A \( \perp \perp \) B | C and A \( \perp \perp \perp \) B | C when A and B are d-separated or d-connected given C, respectively. The conditioning set C is called a minimal separating set if and only if A \( \perp \perp \) B | C but A and B are d-connected given any proper subset of C.

Now if we have A \( \perp \perp \) B | C, then A and B are conditionally independent given C, denoted as A \( \perp \perp \) B | C in any joint density factorizing according to (1); we refer to this property as the global directed Markov property. We also refer to the converse of the global directed Markov property as d-separation faithfulness; that is, if A \( \perp \perp \) B | C, then A and B are d-separated given C. One can in fact show that the factorization in (1) and the global directed Markov property are equivalent, so long as the distribution over X admits a density [7].

A maximal ancestral graph (MAG) is an ancestral graph where every missing edge corresponds to a conditional independence relation. We specifically partition X = O \( \cup \) L \( \cup \) S into observable, latent and selection variables, respectively. One can then transform a DAG G = (O \( \cup \) L \( \cup \) S, E) into a MAG \( \hat{G} = (O, \hat{E}) \) as follows. First, for any pair of vertices \{O_i, O_j\}, make them adjacent in \( \hat{G} \) if and only if there is an inducing path between O_i and O_j in G. We define an inducing path as follows:

Definition 1 A path \( \pi \) between O_i and O_j is called an inducing path with respect to L and S if and only if every collider on \( \pi \) is an ancestor of \{O_i, O_j\} \( \cup \) S, and every non-collider on \( \pi \) (except for the endpoints) is in L.

Note that two observables O_i and O_j are connected by an inducing path if and only if there are d-connected given any W \( \subseteq \) O \( \setminus \) \{O_i, O_j\} as well as S. Then, for each adjacency O_i \( \leftrightarrow \) O_j in \( \hat{G} \), place an arrowhead at O_i if O_i \( \notin \) An(O_j \( \cup \) S) and place a tail otherwise. The MAG of a DAG is therefore a kind of marginal graph that does not contain the latent or selection variables, but does contain information about the

\[\text{ Springer}\]
Fig. 1  A dataset in a subjected to selection bias according to \( X_2 \) in the DAG in b. In a, we can only view the samples in blue in practice. The dataset in c is the same dataset in a but with some missing values. The variable in pink is subject to selection bias according to \( X_2 \) and \( X_5 \) as in d due to the missing values, while the variable in green is subject only to \( X_2 \) as in b (color figure online)

| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) |
|----------|----------|----------|----------|----------|
| 101K     | 0        | 122      | 15.1     | 1        |
| 52K      | 0        | 132      | 0.5      | 1        |
| 48K      | 1        | 111      | 29.8     | 0        |
| 35K      | 1        | 141      | 2.5      | 1        |
| 123K     | 0        | 109      | 4.7      | 1        |
| 17K      | 1        | 125      | 10.8     | 1        |
| 42K      | 1        | 119      | 14.6     | 0        |

(a)

| \( X_1 \) | \( X_2 \) | \( X_3 \) | \( X_4 \) | \( X_5 \) |
|----------|----------|----------|----------|----------|
| 101K     | 0        | 122      | 15.1     | 1        |
| 52K      | 0        | 132      | 0.5      | 1        |
| 48K      | 1        | 111      | 29.8     | 0        |
| 35K      | 1        | 141      | 2.5      | 1        |
| 123K     | 0        | 109      | 4.7      | 1        |
| 17K      | 1        | 125      | 10.8     | 1        |
| 42K      | 1        | 119      | 14.6     | 0        |

(c)

ancestral relations between the observable and selection variables in the DAG. The MAG also has the same d-separation relations as the DAG, specifically among the observable variables conditional on the selection variables [20].

3 Selection bias

Selection bias refers to the preferential selection of samples from \( P_X \), potentially due to some unknown factors \( S \subseteq X \). Such preferential selection occurs in a variety of real-world contexts. For example, a psychologist may wish to discover principles of the mind that apply to the general population, but he or she may only have access to data collected from college students. A medical investigator may similarly wish to elucidate a disease process occurring in all patients with the disease, but he or she may only have samples collected from low income patients in Chicago who chose to enroll in the investigator’s study.

We can represent selection bias graphically using a DAG over \( X = \{O \cup L \cup S\} \). We specifically let \( S \) denote a set of binary indicator variables taking values in \( \{0, 1\} \). Without loss of generality, we then say that a sample is selected if and only if all of the indicator variables in \( S \) take on a value of one. The preferential selection of samples due to selection bias therefore amounts to conditioning on \( S = 1 \); in other words, we no longer have access to i.i.d. samples from \( P_O \) but rather i.i.d samples from \( P_{O|S=1} \).

As an example, let \( X = \{X_1, \ldots, X_5\}, O = \{X_1, X_3\} \) and \( L = \{X_4, X_5\} \). Also let \( S = X_2 \) correspond to a binary variable taking the value of 1 when \( X_1 \) is less than 50K and 0 otherwise. Consider drawing i.i.d. samples from a joint distribution \( P_X \) as shown in Fig. 1a; here, each sample corresponds to a row in the table. The caveat, however, is that we can only observe the values of \( \{X_1, X_3\} \) when \( X_1 \) is below 50K as highlighted in blue in Fig. 1a. We therefore observe \( \{X_1, X_3\} \) when \( X_2 = S \) takes on a value of 1, and otherwise we do not. In the real world, this situation may correspond to a physician who wants to measure the income \( X_1 \) and resting systolic blood pressure (SBP) \( X_3 \) of patients in the true patient population. The physician can nevertheless only measure \( \{X_1, X_3\} \) in patients with low income, since patients with low income tend to enroll in medical studies more often than patients with high income. Thus, we no longer have access to i.i.d. samples from \( P_{X_1,X_3} \) but rather i.i.d samples from \( P_{X_1,X_3|X_2=1} \), or equivalently \( P_{O|S=1} \).

We can represent the causal process in the above example using the probabilistic DAG represented in Fig. 1b. Here, we interpret \( X_2 \) as a child of \( X_1 \), since \( X_2 \) represents an indicator variable that takes on values according to the values of \( X_1 \). Notice also the double-sided vertex in Fig. 1b which denotes the conditioning on low income when \( X_2 = 1 \).

4 Missingness as selection bias on selection bias

We can informally interpret missing values as a type of “selection bias on selection bias.” Here, the first layer of selection bias due to \( S \) refers to the aforementioned measurement of all variables in \( O \) in a preferential selection of the samples. Missing values in turn represent the second
layer of selection bias because missing values arise due to the measurement of only a subset of the variables in \( O \) in a preferential subset of the available samples already subject to the selection bias of \( S \).

The missingness may more formally arise for many reasons as modeled by the factors \( S_{O_1} \supseteq S \) for \( O_1 \), \( S_{O_2} \supseteq S \) for \( O_2 \), and so on for all \( p \) variables in \( O \). We therefore encode the binary missingness status (measured or missing) of any observable \( O_i \in O \) using the binary missingness indicators \( S_{O_i} \setminus S \subseteq L \). Here, we measure the value of \( O_i \) if and only if \( S_{O_i} = 1 \) because we must select a sample when \( S = 1 \) and then measure the value of \( O_i \) when \( (S_{O_i} \setminus S) = 1 \). The preferential selection of samples due to some \( S_{O_i} \) thus amounts to conditioning on \( S_{O_i} = 1 \) similar to the original selection bias case; in other words, we no longer have access to i.i.d. samples from a set of conditional distributions for \( S \). Now, the variable \( X_3 \) highlighted in pink in Fig. 1c is subject to the selection variables \( S_{X_3} = \{X_2, X_3\} = 1 \) due to the unmeasured or missing values. On the other hand, the variable \( X_1 \) in green is only subject to the original \( S_{X_1} = \{X_2\} = S = 1 \) because \( X_1 \) contains no missing values. We can therefore represent these two situations graphically as in Fig. 1b for \( S_{X_1} \) and Fig. 1d for \( S_{X_3} \). Notice that Fig. 1d has an extra double-sided vertex \( X_5 \) representing the extra conditioning.

Returning to our medical example, \( X_4 \) may correspond to the number of miles from the hospital to a patient’s house. Individuals with low income may have a hard time commuting to the hospital, if they live far away. The physician therefore may not be able to measure SBP \( X_3 \) in low income patients who live far from the hospital. We thus no longer even have access to i.i.d. samples from \( P_{X_1,X_3 \mid S = 1} \) but instead have access to i.i.d. samples from a set of conditional distributions \{\( P_{X_1 \mid S_{X_1} = 1}, P_{X_3 \mid S_{X_1} = 1}, P_{X_1,X_3 \mid S_{X_1} = 1} \)\}.

More generally, we do not have access to i.i.d. samples from \( P_{O_i \mid S = 1} \) when missing values exist. Instead, we have access to i.i.d. samples from a set of conditional distributions \{\( P_{V \mid S_V = 1}, \forall V \subseteq O \)\}. In this sense, we must deal with heterogeneous selection bias induced by \( S_{O_1}, \ldots, S_{O_p} \) as opposed to homogeneous selection bias induced by just \( S \).

5 An assumption on the missingness mechanisms

Let \( S^u \) denote the set of \( q \leq p \) unique members of \( \{S_{O_1}, \ldots, S_{O_p}\} \). Note that we have so far imposed no restrictions on the causal relations involving \( S \) or any member of \( S^u \). From here on, we will continue to impose no restrictions on the causal relations involving \( S \), but we will impose restrictions on the causal relations involving the elements in the set \( M = \{S_{O_1}^u \setminus S, \ldots, S_{O_p}^u \setminus S\} \).

Recall that each \( M_i \in M \) corresponds to a set of missingness indicators, but we can colloquially call each \( M_i \in M \) a “missingness mechanism” because we obtain missing values for some subset of variables \( V \subseteq O \) when (at least) one member of \( M_i \) takes on a value of 0. Here, a missingness mechanism often corresponds to a practical issue. For example, we may have three variables in \( M_i \) corresponding to three instruments required to perform a measurement. We have \( M_i = 1 \) when three instruments can perform the measurement but one variable in \( M_i \) equals zero when one of the three instruments fails. We may similarly have \( M_3 = 1 \) when a subject can commute to the hospital and \( M_3 = 0 \) when the subject cannot commute to the hospital as in the running medical example; thus \( M_3 = X_5 \) in this case.

Now let \( I_i = \{1, \ldots, q \} \setminus i \) and consider the following assumption:

**Assumption 1 (Non-ancestral missingness)** Each \( M_i \in M \) does not contain an ancestor of any variable in \( O \cup S \) or \( \bigcup_{j \in I_i} M_j \).

The above non-ancestral missingness assumption appears technical at first glance, but we can justify it using an inductive argument that reads as follows. First note that we have no missing values if and only if all variables in the sets in \( M \) take on a value of one. Suppose then that we have a missing value but then manually set all variables in the sets in \( M \) to one in order to observe the value. Then we do not expect the mere act of observing a value, or equivalently intervening on the sets in \( M \), to induce changes in (or causally affect) (1) the values of the observable variables \( O \) in a dataset or (2) the set of available samples determined by \( S \). In other words, none of the variables in any set in \( M \) should be an ancestor of any of the variables in \( O \cup S \).

The non-ancestral missingness assumption, however, imposes the extra condition that no missingness mechanism \( M_i \in M \) contains an ancestor of any variable in \( \bigcup_{j \in I_i} M_j \). We find the extra assumption reasonable, if an attempt is made to measure each observable variable in \( O \) for each sample regardless of the missingness status of any other variable in \( S \). This means that the missingness statuses cannot causally affect each other. Now recall that the missingness status of any variable is determined by the variable’s missing mechanism. Hence, we can equivalently state that the missingness mechanisms cannot causally affect each other (i.e., each \( M_i \in M \) does not contain an ancestor of any vari-
ables in $\cup_{j \in I_t} M_j$). For example, we assume that a failure of any one of three instruments does not cause an investigator to potentially forgo the measurement of other variables which do not require the instruments but say rather cost a lot of money. Instead, the investigator attempts to measure the other variables regardless of whether or not an instrument fails. The instrument failures are thus not causes of the inability to pay or any other missingness mechanism. We conclude inductively that non-ancestral missingness is justified with "comprehensively measured observational data," where an attempt is made to measure each observable variable in $O$ for each sample regardless of the missingness status of any other variable in $O$.

6 The FCI and RFCI algorithms

We can justify test-wise deletion, if we can utilize the non-ancestral missingness assumption as well as CCD algorithms which can handle selection bias. We will focus on two such CCD algorithms in particular called FCI and RFCI. We therefore cover the two algorithms briefly in the next two subsections.

6.1 The FCI algorithm

The FCI algorithm considers the following problem: assume that the distribution of $X = O \cup L \cup S$ is d-separation faithful to an unknown DAG. Then, given oracle information about the conditional independencies between any pair of variables $O_i$ and $O_j$ given any $W \subseteq O \setminus \{O_i, O_j\}$ as well as $S$, reconstruct as much information about the underlying DAG as possible. The FCI algorithm ultimately accomplishes this goal by reconstructing a MAG up to its Markov equivalence class, or the set of the MAGs with the same conditional independence relations over $O$ given $S$.

The FCI algorithm represents the Markov equivalence class of MAGs, or the set of MAGs with the same conditional dependence and independence relations between variables in $O$ given $S$, using a completed partial maximal ancestral graph (CPMAG). A partial maximal ancestral graph (PMAG) is nothing more than a MAG with some circle endpoints. A PMAG is completed (and hence a CPMAG) when the following conditions hold: (1) every tail and arrowhead also exists in every MAG belonging to the Markov equivalence class of the MAG, and (2) there exists a MAG with a tail and a MAG with an arrowhead in the Markov equivalence class for every circle endpoint. Each edge in the CPMAG also has the following interpretations:

(i) An edge is absent between two vertices $O_i$ and $O_j$ if and only if there exists some $W \subseteq O \setminus \{O_i, O_j\}$ such that $O_i \perp \perp O_j | (W, S)$. That is, an edge is absent if and only if there does not exist an inducing path between $O_i$ and $O_j$ with respect to $L$ and $S$.

(ii) If an edge between $O_i$ and $O_j$ has an arrowhead at $O_j$, then $O_j \notin \text{An}(O_i \cup S)$.

(iii) If an edge between $O_i$ and $O_j$ has a tail at $O_j$, then $O_j \in \text{An}(O_i \cup S)$.

The FCI algorithm learns the CPMAG through a three-step procedure. Most of the algorithmic details are not important for this paper, so we refer the reader to [18] and [29] for algorithmic details. However, three components of FCI (R4) are important. V-structure discovery reads as follows: suppose $O_i$ and $O_k$ are adjacent, $O_j$ and $O_k$ are adjacent, but $O_i$ and $O_j$ are non-adjacent. Further assume that we have $O_i \perp \perp O_j | (W, S)$ with $W \subseteq O \setminus \{O_i, O_j\}$ and $O_k \notin W$. Then orient the triple $\langle O_i, O_k, O_j \rangle$ as $O_i \rightarrow O_k \rightarrow O_j$. R1 reads as follows: if we have $O_i \lor O_k \lor O_j$, $O_i \perp \perp O_j | (W, S)$ with $W \subseteq O \setminus \{O_i, O_j\}$ minimal, and $O_k \notin W$, then orient $O_i \lor O_k \lor O_j$ as $O_i \lor O_k \lor O_j$; here, the asterisk represents a placeholder for either a tail, arrowhead or circle. R4 involves the detection of additional colliders in certain shielded triples.

6.2 The RFCI algorithm

Discovering inducing paths can require large d-separating sets, so the FCI algorithm often takes too long to complete. The RFCI algorithm [2] resolves this problem by recovering a graph where the presence and absence of an edge have the following modified interpretations:

(i) The absence of an edge between two vertices $O_i$ and $O_j$ implies that there exists some $W \subseteq O \setminus \{O_i, O_j\}$ such that $O_i \perp \perp O_j | (W, S)$.

(ii) The presence of an edge between two vertices $O_i$ and $O_j$ implies that $O_i \not\perp \not\perp O_j | (W, S)$ for all $W \subseteq \text{Adj}(O_i) \setminus O_j$ for all $W \subseteq \text{Adj}(O_j) \setminus O_i$. Here $\text{Adj}(O_i)$ denotes the set of vertices adjacent to $O_i$ in RFCI’s graph.

We encourage the reader to compare these edge interpretations to the edge interpretations of FCI’s CPMAG.

The RFCI algorithm learns its graph (not necessarily a CPMAG) also through a three-step procedure. We refer the reader to [2] for algorithmic details.
7 Graph theory

We now analyze the effects of using test-wise deletion with FCI and RFCI in detail. We will consider the set of selection variables \( S_i = \bigcup_{i=1}^{p} S_{Oi} \). Notice that we obtain \( S_i = 1 \), when we perform list-wise deletion on the dataset. Also let \( S_{Oi,Oj,W} \) refer to the selection set induced by the complete samples among the variables \( O_i, O_j \) and \( W \subseteq O \setminus \{ O_i, O_j \} \) alone by setting \( V = \{ O_i, O_j, W \} \) in \( SV \); in other words, \( S_{Oi,Oj,W} \) corresponds to the selection variables obtained after performing test-wise deletion, or list-wise deletion only among the variables \( \{ O_i, O_j, W \} \).

The following important lemma now forms the basis of our arguments:

**Lemma 1** Assume non-ancestral missingness. If \( O_i \perp \! \! \! \perp_d O_j |(W, S_{Oi,Oj,W}) \) with \( W \subseteq O \setminus \{ O_i, O_j \} \) minimal, then \( O_i \perp \! \! \! \perp_d O_j |(A, S_{Oi,Oj,W}) \), where \( A \) denotes an arbitrary strict subset of \( W \). Now \( O_i \perp \! \! \! \perp_d O_j |(A, S_{Oi,Oj,W}) \) implies \( O_i \perp \! \! \! \perp_d O_j |(A, S_i) \) by Lemma 1. The conclusion follows because we chose \( A \) arbitrarily and assumed \( O_i \perp \! \! \! \perp_d O_j |(W, S_i) \).

**Proof** If we have \( O_i \perp \! \! \! \perp_d O_j |(W, S_{Oi,Oj,W}) \) with \( W \subseteq O \setminus \{ O_i, O_j \} \) minimal, then \( O_i \perp \! \! \! \perp_d O_j |(A, S_{Oi,Oj,W}) \), where \( A \) denotes an arbitrary strict subset of \( W \). Now \( O_i \perp \! \! \! \perp_d O_j |(A, S_{Oi,Oj,W}) \) implies \( O_i \perp \! \! \! \perp_d O_j |(A, S_i) \) by Lemma 1. The conclusion follows because we chose \( A \) arbitrarily and assumed \( O_i \perp \! \! \! \perp_d O_j |(W, S_i) \).

Note that FCI and RFCI require many calls to a CI oracle precisely because they search for minimal separating sets. We can therefore take advantage of Lemma 2 by searching for minimal separating sets with test-wise deletion and then only confirming the separating sets with list-wise deletion (rather than directly searching for minimal separating sets with list-wise deletion). This greatly reduces the number of CI tests performed with list-wise deletion.

We can now directly justify some desired conclusions. Let us examine the most difficult arguments in detail. We have the following conclusion for \( v \)-structure discovery and R1:

**Proposition 1** Assume non-ancestral missingness. Suppose \( O_i \perp \! \! \! \perp_d O_j |(W, S_{Oi,Oj,W}) \) with \( W \subseteq O \setminus \{ O_i, O_j \} \) minimal and \( O_i \perp \! \! \! \perp_d O_j |(W, S_i) \). Further assume \( O_i \perp \! \! \! \perp_d O_k |(W, S_{Oi,Oj,W}) \) and \( O_j \perp \! \! \! \perp_d O_k |(W, S_{Oi,Oj,W}) \). We have \( O_k \in W \) if and only if \( O_k \in An \{ O_i, O_j \} \cup S_i \).

**Proof** If \( O_i \perp \! \! \! \perp_d O_j |(W, S_{Oi,Oj,W}) \) with \( W \subseteq O \setminus \{ O_i, O_j \} \) minimal and \( O_i \perp \! \! \! \perp_d O_j |(W, S_i) \), then \( O_i \perp \! \! \! \perp_d O_j |(W, S_i) \) with \( W \) minimal by Lemma 1. By Lemma 2, we know that \( O_i \perp \! \! \! \perp_d O_j |(W, S_{Oi,Oj,W}) \) and \( O_j \perp \! \! \! \perp_d O_k |(W, S_{Oi,Oj,W}) \) imply \( O_i \perp \! \! \! \perp_d O_k |(W, S_i) \) and \( O_j \perp \! \! \! \perp_d O_k |(W, S_i) \), respectively. We may now invoke Lemma 3.1 of [2] with \( S_i \).

We can also justify the discriminating path rule (R4) with a similar argument:

**Proposition 2** Assume non-ancestral missingness. Let \( \pi_{ik} = \{ O_i, \ldots, O_i, O_j, O_k \} \) be a sequence of at least four vertices which satisfy the following:

1. \( O_i \perp \! \! \! \perp_d O_j |(W, S_j) \) with \( W \subseteq O \setminus \{ O_i, O_j \} \).
2. Any two successive vertices \( O_h \) and \( O_{h+1} \) on \( \pi_{ik} \) are \( d \)-connected given:
   \[(Y \setminus \{ O_h, O_{h+1} \}) \cup S_{Oh,O_{h+1}}(Y \setminus \{ O_h, O_{h+1} \})\]
   for all \( Y \subseteq W \).
3. All vertices \( O_h \) between \( O_i \) and \( O_j \) (not including \( O_i \) and \( O_j \)) satisfy \( O_h \in An(O_i) \) and \( O_h \notin An(O_{h-1}, O_{h+1} \cup S_i) \), where \( O_{h-1} \) and \( O_{h+1} \) denote the vertices adjacent to \( O_h \) on \( \pi_{ik} \).

Then, if \( O_j \in W \), then \( O_j \in An(O_k \cup S_i) \) and \( O_k \in An(O_j \cup S_i) \). On the other hand, if \( O_j \notin W \), then \( O_j \notin An(O_i, O_k) \cup S_i \) and \( O_k \notin An(O_j \cup S_i) \).
8 FCI and RFCI with test-wise deletion

We introduced some direct proofs in Propositions 1 and 2 of the previous section which capitalize on Lemmas 1 and 2. We can, however, actually prove the full soundness and completeness of FCI in one sweep by designing a CI oracle wrapper.

We introduce the CI oracle wrapper in Algorithm 1. Without loss of generality, we assume that the CI oracle outputs 0 when conditional dependence holds and 1 otherwise. The wrapper works by first querying the CI oracle with \( S_{O_i, O_j, W} \) in line 1. If the CI oracle outputs 1, then the wrapper also checks whether the CI oracle outputs 1 with \( S_l \) in line 3. If so, the wrapper outputs 1 and otherwise outputs 0 due to line 4. Hence, the wrapper claims conditional independence only when both the CI oracle with \( S_{O_i, O_j, W} \) and the CI oracle with \( S_l \) output 1 thus implementing Lemma 2. On the other hand, if the CI oracle with \( S_{O_i, O_j, W} \) outputs 0, then the wrapper immediately outputs 0 thus implementing Lemma 1. Notice that the finite sample version of Algorithm 1 follows immediately by replacing line 2 with an \( \alpha \) level cutoff.

**Algorithm 1**: CI oracle wrapper

```plaintext
Data: \( O_i, O_j, W \)

Result: \( p \)

1. \( p \leftarrow \) Ask CI oracle whether \( O_i \perp\!\!\!\perp O_j|(W, S_{O_i, O_j, W}) \)
2. **If** \( p = 1 \) **then**
3. \( q \leftarrow \) Ask CI oracle whether \( O_i \perp\!\!\!\perp O_j|(W, S_l) \)
4. \( p \leftarrow \min(p, q) \)
5. **end**

**Theorem 1** Assume non-ancestral missingness. Further assume d-separation faithfulness. Then FCI using Algorithm 1 outputs the same graph as FCI using a CI oracle with \( S_l \). The same result holds for RFCI.

**Proof** Recall that d-separation and conditional independence are equivalent under d-separation faithfulness, so we can talk about d-separation and conditional independence interchangeably. It then suffices to show that \( O_i \perp\!\!\!\perp O_j|(W, S_l) \) if and only if Algorithm 1 outputs zero. For the backward direction, if Algorithm 1 outputs zero, then we must have (1) \( O_i \not\perp\!\!\!\perp O_j|(W, S_{O_i, O_j, W}) \) or (2) \( O_i \not\perp\!\!\!\perp O_j|(W, S_l) \) (or both). If (2) holds, the conclusion follows immediately. If (1) holds, then the conclusion follows by Lemma 1. For the forward direction, assume for a contrapositive that Algorithm 1 outputs one. The conclusion follows because Algorithm 1 outputs one only if we have \( O_i \perp\!\!\!\perp O_j|(W, S_{O_i, O_j, W}) \) and \( O_i \perp\!\!\!\perp O_j|(W, S_l) \).

9 Experiments

We now describe the experiments used to assess the finite sample size performance of test-wise deletion as compared to existing approaches.
9.1 Algorithms

We compared the following algorithms with Fisher’s z-test and $\alpha$ set to 0.01:

1. FCI with test-wise deletion (i.e., equipped with Algorithm 1);
2. FCI with heuristic test-wise deletion, where we only run line 1 of Algorithm 1. We can justify this procedure under values missing completely at random (MCAR) as explained in detail in Appendix 11.2;
3. FCI with list-wise deletion [17];
4. FCI with five different imputation methods including hot deck [3], k-nearest neighbor ($k = 5$; k-NN) [6], Bayesian linear regression (BLR) [1,12,28], predictive mean matching (PMM) [8,11,27,28] and random forests (ntree = 10; RF) [5,13,26].

We then repeated the comparisons with RFCI in place of FCI. We therefore compared a total of 16 methods.

Note that FCI with heuristic test-wise deletion is not justified in the general MNAR case, but we find that it performs well with finite sample CI tests and hence report its results mainly in Appendix.

9.2 Synthetic data

9.2.1 Data generation

We used the following procedure in [2] to generate 400 different Gaussian DAGs with an expected neighborhood size of $E(N) = 2$ and $p = 20$ vertices. First, we generated a random adjacency matrix $A$ with independent realizations of Bernoulli($E(N)/(p – 1)$) random variables in the lower triangle of the matrix and zeroes in the remaining entries. Next, we replaced the ones in $A$ by independent realizations of a Uniform$([-1, -0.1] \cup [0.1, 1])$ random variable. We can interpret a nonzero entry $A_{ij}$ as an edge from $X_i$ to $X_j$ with coefficient $A_{ij}$ in the following linear model:

$$
X_1 = \varepsilon_1, \\
X_i = \sum_{r=1}^{p-1} A_{ir} X_r + \varepsilon_i,
$$

for $i = 2, \ldots, p$ where $\varepsilon_1, \ldots, \varepsilon_p$ are mutually independent $N(0, 1)$ random variables. We finally introduced nonzero means $\mu$ by adding $p$ independent realizations of a $N(0, 4)$ random variable to $X$. The variables $X_1, \ldots, X_p$ then have a multivariate Gaussian distribution with mean vector $\mu$ and covariance matrix $\Sigma = (I - A)^{-1}(I - A)^{-T}$, where $I$ is the $p \times p$ identity matrix.

We generated MNAR datasets using the following procedure. We first randomly selected a set of $0$–$4$ latent common causes $L$ without replacement. We then selected a set of $1$–$2$ additional latent variables $\tilde{L}$ without replacement from the set $X \setminus L$. Next, we randomly selected a subset of $3$–$6$ variables in $O = [X \setminus (L, \tilde{L})]$ without replacement for each $\tilde{L} \in \tilde{L}$ and then removed the bottom $r$ percentile of samples from those $3$–$6$ variables according to $\tilde{L}$; we drew $r$ according to independent realizations of a Uniform([0.1, 0.5]) random variable. Thus, the missing values depend directly on the unobservables $\tilde{L}$ in this MNAR case. The non-ancestral missingness assumption is also satisfied because none of the missingness indicators have children. We finally eliminated all of the instantiations of the latent variables $L \cup \tilde{L}$ from the dataset.

For the MAR case, we again randomly selected a set of $0$–$4$ latent common causes $L$ (at least two children) without replacement. We then selected a set of $1$–$2$ observable variables $\tilde{O} \subseteq O = [X \setminus L]$ without replacement and then randomly selected a subset of $3$–$6$ variables in $O \setminus \tilde{O}$ without replacement for each variable in $\tilde{O} \in \tilde{O}$. We next removed the bottom $r$ percentile of samples from those $3$–$6$ variables according to $\tilde{O}$; we again drew $r$ according to independent realizations of a Uniform([0.1, 0.5]) random variable. Thus, the missing values depend directly on the observables $\tilde{O}$ with no missing values in this MAR case. We finally again eliminated all of the instantiations of the latent variables $L$ from the dataset.

We ultimately created datasets with sample sizes of 100, 250, 500, 1000 and 5000 for each of the 400 DAGs for both the MNAR and MAR cases. We therefore generated a total of $400 \times 5 \times 2 = 4000$ datasets.

9.2.2 Metrics

We compared the algorithms using the structural Hamming distance (SHD) from the oracle graphs in the MNAR and MAR cases. We in particular computed the mean SHD difference which is equal to the mean of the SHD with test-wise deletion minus the SHD with a comparison method (e.g., list-wise or heuristic test-wise deletion).

We set the selection variables to $S_l$ for the oracle graphs in the MAR case, since FCI with test-wise and list-wise deletion recover these graphs in the sample limit. Here, we hope FCI and RFCI with test-wise deletion will outperform FCI and RFCI with list-wise deletion, respectively, by obtaining lower SHD scores on average.

We also set the selection variables to the empty set for the oracle graphs in the MAR case, since a sound imputation method should recover the underlying distribution without selection bias using the observed values. Note, however, that
Fig. 3 FCI and RFCI with test-wise deletion versus the same algorithms with list-wise deletion in the MNAR case. Test-wise deletion results in a decrease in the average SHD for FCI in (a) and RFCI in (b). The performance increase results because of a 25–35% increase in sample size per CI test on average for FCI in (c) and RFCI in (d).

test-wise deletion cannot eliminate the selection bias induced by the observed values in this case. Clearly then the imputation methods should outperform test-wise deletion under the metric of SHD to the oracle graph without selection bias. However, we still hope that the algorithms with test-wise deletion will perform reasonably well because none of variables in \( \tilde{O} \) induce dense MAGs by design when acting as selection variables.

9.2.3 Results

We have summarized the results for the MNAR case in Fig. 3. We focus on comparing FCI and RFCI with test-wise deletion against the same algorithms with list-wise deletion. FCI and RFCI with any of the five imputation methods expectedly performed much worse in this task, so we relegate the imputation results to Fig. 9 in Appendix. We have also summarized the excellent results of heuristic test-wise deletion in Fig. 6 in Appendix.

Figure 3a, b suggests that the algorithms with test-wise deletion consistently outperform their list-wise deletion counterparts across all sample sizes. In fact, most test-wise versus list-wise comparisons were significant using paired t tests at a Bonferroni corrected threshold of 0.05/5 for both FCI and RFCI (exceptions: FCI at sample size 5000, \( t = 2.018, p = 0.044 \); RFCI 1000, \( t = 2.108, p = 0.036 \); RFCI 5000, \( t = 1.055, p = 0.292 \)). We found the largest gains with smaller sample sizes, where the CCD algorithms are prone to error and greatly benefit from the sample size increase (sample size vs. SHD difference correlation; FCI: Pearson’s \( r = -0.0941, t = -4.226, p = 2.48E-5 \); RFCI: \( r = -0.111, t = -5.000, p = 6.24E-7 \)). Figure 3c, d lists the average sample size increase per executed CI test for test-wise deletion compared to list-wise deletion in percentage points. We see that test-wise deletion results in an approximately 25–35% increase in sample size than list-wise deletion regardless of the algorithm. RFCI in particular benefits the most at a steady 35% regardless of the sample size because the algorithm utilizes smaller conditioning set sizes than FCI, so test-wise deletion results in the deletion of even fewer samples per CI test than list-wise deletion for RFCI. We conclude that FCI and RFCI with test-wise deletion consistently outperform their list-wise deletion counterparts because Algorithm 1 allows the algorithms to more efficiently utilize the available samples.

We have also summarized the results for the MAR case in Fig. 4. Figure 4a, b lists the average ranked results against the five imputation methods. A rank of one denotes the best
Fig. 4 Test-wise deletion versus list-wise deletion as compared to five imputation methods in the MAR case. A rank of one denotes the best performance and a rank of six denotes the worst. Test-wise deletion has a smaller average rank than list-wise deletion for FCI in (a) and RFCI in (b). The performance increase in test-wise deletion again results because of increased sample efficiency for FCI in (c) and RFCI in (d).

performance whereas a rank of six denotes the worst. We see that test-wise deletion again outperforms list-wise deletion. The effect is significant for all sample sizes between 100 and 500 at a Bonferroni level of 0.05/5 for FCI (max $t = -5.223$, max $p = 2.84E-7$) and for RFCI (max $t = -4.147$, max $p = 4.12E-5$). The performance improvements result from the improved sample efficiency of test-wise deletion as compared to list-wise deletion in both FCI (approx. 15–30% increase; Fig. 4c) and RFCI (approx. 30% increase; Fig. 4d).

Test-wise deletion and list-wise deletion also perform very well overall even in the MAR case as compared to imputation methods. Both methods perform approximately middle of the road (rank approx. 2.5) and are only consistently outperformed by BLR and PMM, where the linear models are correctly specified. On the other hand, the nonparametric k-NN and random forest imputation methods often fall short of both test-wise and list-wise deletion. We conclude that FCI and RFCI with test-wise deletion are competitive against the same algorithms with imputation even when MAR strictly holds.

9.3 Real data

We finally ran the same algorithms using the nonparametric CI test called RCoT [21] at $\alpha = 0.01$ on a publicly available longitudinal dataset from the Cognition and Aging USA (CogUSA) study [9], where scientists measured the cognition of men and women above 50 years of age. The dataset contains three waves of data, but we specifically focused on the first two waves in this dataset. The first two waves are only separated by one week, and the investigators collected data for the first wave by telephone. Note that neuropsychological interventions are near impossible within a week after phone-based testing, so no missingness indicator should be an ancestor of $O \cup S$. Moreover, to the best of our knowledge, the investigators attempted to measure each variable regardless of the missingness statuses of the other variables for each sample. We can therefore justify the non-ancestral missingness assumption in this setting.

We used a cleaned version of the dataset containing 1514 samples over 16 variables; we specifically removed deterministic relations and variables related to administrative purposes as opposed to neuropsychological variables. Despite the cleaning, the dataset contains many missing values. List-wise deletion drops the number of samples from 1514 to 1106. However, this is also precisely the setting where we hope to use test-wise deletion in order to increase sample efficiency.

Note that we do not have access to a gold standard solution set in this case. However, we can develop an approximate solution set by utilizing two key facts. First, recall that we
cannot have ancestral relations directed backwards in time. Thus, a variable in wave 2 cannot be an ancestor of a variable in wave 1; we can therefore count the number of edges between wave 1 and wave 2 with both a tail and an arrowhead at a vertex in wave 2. Second, the mental status score is a composite score that includes backwards counting as well as some other metrics. Thus, there should exist an edge between backwards counting and mental status, and the edge ideally should have a tail at backwards counting as well as an arrowhead at mental status in both waves.

We used the above solution set to construct the following cost metric; we counted the number of incorrect ancestral relations \( w \) as well as counted the number of unoriented or incorrectly oriented endpoints between backwards counting and mental status \( v \). Allow \( w + v \) therefore indicates better performance.

We have summarized the results in Fig. 5 after generating 300 bootstrapped datasets. Test-wise deletion outperforms 6 of the 7 other methods at a Bonferroni corrected threshold of 0.05/7 when incorporated into FCI (max \( t = -3.210 \), max \( p = 1.47E-3 \); test-wise deletion also outperformed FCI with PMM but not by a significant margin (\( t = -1.331 \), \( p = 0.184 \)). Test-wise deletion did, however, outperform all of the other 7 methods with RFCI (max \( t = -3.482 \), max \( p = 5.73E-4 \)). Moreover, test-wise deletion conserves an average of 8.96% more samples per CI test (95% CI 7.95–9.98%) than list-wise deletion for FCI and similarly 8.82% (95% CI 7.82–9.82%) for RFCI. On the other hand, heuristic test-wise deletion conserves only 1.05% (95% CI 1.00–1.10%) more samples than test-wise deletion for FCI and only 0.98% (95% CI 0.94–1.02%) more samples for RFCI. We conclude that the real data results largely replicate the synthetic data results for the MNAR case.

10 Conclusion

We proposed test-wise deletion as a strategy to improve upon list-wise deletion for CCD algorithms even when MNAR holds. Test-wise deletion specifically involves running FCI or RFCI using Algorithm 1 without pre-processing the missing values. We proved soundness of the procedure so long as the missingness mechanisms do not causally affect each other in the underlying causal graph. Moreover, experiments highlighted the superior sample efficiency of test-wise deletion as compared to list-wise deletion. We conclude that test-wise deletion is a viable alternative to list-wise deletion when MNAR holds.

We ultimately hope that test-wise deletion will prove useful for investigators wishing to apply CCD algorithms on data with missing values. Test-wise deletion is easily implemented in a few lines of code via Algorithm 1. Here, we simply call a CCD algorithm equipped with Algorithm 1 in place of a normal CI test.

Acknowledgements Research reported in this publication was supported by Grant U54HG008540 awarded by the National Human Genome Research Institute through funds provided by the trans-NIH Big Data to Knowledge initiative. The research was also supported by the National Library of Medicine of the National Institutes of Health under award numbers T15LM007059 and R01LM012095. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

11 Appendix

11.1 Graphical terminology

We will represent causalit by Markovian graphs. We therefore require some basic graphical definitions.

A graph \( \mathcal{G} = (X, \mathcal{E}) \) consists of a set of vertices \( X = \{X_1, \ldots, X_p\} \) and a set of edges \( \mathcal{E} \) such that one edge exists between each pair of vertices. The edge set \( \mathcal{E} \) may contain the following six edge types: \( \rightarrow \) (directed), \( \leftrightarrow \) (bidirected), \( \quad\) (undirected), \( \quad\rightarrow \) (partially directed), \( \quad\) (partially undirected) and \( \quad\circ\circ \) (nondirected). Notice that these six edges utilize three types of endpoints including tails, arrowheads, and circles.
Fig. 6 Test-wise deletion versus heuristic test-wise deletion in the MNAR case. Notice that the y-axis is reversed in a, b. We find that test-wise deletion underperforms heuristic test-wise deletion by yielding slightly larger SHD values on average according to a, b. Subfigures c and d show the increase in average sample size per CI test for heuristic test-wise deletion as compared to test-wise deletion.

Fig. 7 Test-wise deletion versus heuristic test-wise deletion in skeleton discovery in the MNAR case. We find that test-wise deletion outperforms heuristic test-wise deletion by yielding smaller skeleton SHD values on average. Moreover, the margin gradually increases with sample size.

We call a graph containing only directed edges as a directed graph. We will only consider simple directed graphs in this paper, or directed graphs containing at most one directed edge between any two distinct vertices. On the other hand, a mixed graph contains directed, bidirected and undirected edges. We say that $X_i$ and $X_j$ are adjacent in a graph, if they are connected by an edge independent of the edge’s type. An (undirected) path $\pi$ between $X_i$ and $X_j$ is a set of consecutive edges (also independent of their type) connecting the variables such that no vertex is visited more than once. A directed path from $X_i$ to $X_j$ is a set of consecutive directed edges from $X_i$ to $X_j$ in the direction of the arrowheads. A cycle occurs when a path exists from $X_i$ to $X_j$, and $X_j$ and $X_i$ are adjacent. More specifically, a directed path from $X_i$ to $X_j$ forms a directed cycle with the directed edge $X_j \rightarrow X_i$ and an almost directed cycle with the bidirected edge $X_j \leftrightarrow X_i$.

Three vertices $\{X_i, X_j, X_k\}$ form an unshielded triple, if $X_i$ and $X_j$ are adjacent, $X_j$ and $X_k$ are adjacent, but $X_i$ and $X_k$ are not adjacent. We call a nonendpoint vertex $X_j$ on a
Fig. 8 Test-wise deletion versus heuristic test-wise deletion as compared to five imputation methods in the MAR case. Heuristic test-wise deletion has a smaller average rank than test-wise deletion for FCI in (a) and RFCI in (b). The performance increase in heuristic test-wise deletion results because of increased sample efficiency for FCI in (c) and RFCI in (d).

Fig. 9 Performance of test-wise deletion versus the best of five imputation methods in terms of the SHD when MNAR holds. Test-wise deletion outperforms the best imputation method (usually RF) by an increasing margin as sample size increases.

path π a collider on π, if both the edges immediately preceding and succeeding the vertex have an arrowhead at X_j. Likewise, we refer to a nonendpoint vertex X_j on π which is not a collider as a non-collider. Finally, an unshielded triple involving \{X_i, X_j, X_k\} is more specifically called a v-structure, if X_j is a collider on the subpath \langle X_i, X_j, X_k \rangle.

We say that X_i is an ancestor of X_j (and X_j is a descendant of X_i) if and only if there exists a directed path from X_i to X_j or X_i = X_j. We write X_i ∈ An(X_j) to mean X_i is an ancestor of X_j and X_j ∈ De(X_i) to mean X_j is a descendant of X_i. We also apply the definitions of an ancestor and descendant to a set of vertices Y ⊆ X as follows:

\[ An(Y) = \{ X_i | X_i ∈ An(X_j) \text{ for some } X_j ∈ Y \} \]

\[ De(Y) = \{ X_i | X_i ∈ De(X_j) \text{ for some } X_j ∈ Y \} \]

We call a directed graph a directed acyclic graph (DAG), if it does not contain directed cycles. Every DAG is a type of...
ancestral graph, or a mixed graph that (1) does not contain directed cycles, (2) does not contain almost directed cycles, and (3) for any undirected edge $X_i - X_j$ in $E$, $X_i$ and $X_j$ have no parents or spouses.

### 11.2 Heuristic test-wise deletion

We consider running FCI or RFCI with only line 1 of Algorithm 1; we therefore do not query the CI oracle with $S_l$ when the CI oracle with $S_{Oi O_j W}$ outputs one.

We refer to the above test-wise deletion strategy as heuristic test-wise deletion because the procedure is not sound in general, even when non-ancestral missingness holds. The problem lies in the inability to query the CI oracle with a consistent set of selection variables either directly (as with list-wise deletion) or indirectly (as with Algorithm 1). We thus often cannot soundly execute FCI or RFCI’s orientation rules. For example, for FCI’s R1, if we have the unshielded triple $O_i \rightarrow O_k \rightarrow O_j$ with $O_k \notin \text{An}(O_i, S_{O_i O_j W})$, (1) $O_i \perp_d O_j | (W_2, S_{O_i O_j W})$ with $W_2 \subseteq O \setminus \{O_i, O_j\}$ minimal and (2) $O_k \in W_2$, then we may claim that $O_k$ is an ancestor of $O_i, O_j$ or $S_{O_i O_j W_2}$ with (1) and (2) (but not $S_{O_i O_j W}$); see Lemma 14 in [19]). We thus cannot conclude in general that we have $O_k \in \text{An}(O_j)$ by using the arrowhead at $O_k$; we can only conclude that $O_k \in \text{An}(O_j, S_{O_i O_j W})$; this fact in turn prevents us from executing R1 by orienting $O_i \rightarrow O_k \rightarrow O_j$ as $O_i \rightarrow O_k \rightarrow O_j$.

We can, however, justify heuristic test-wise deletion under MCAR, where missing values do not depend on any other measured or missing values. One interpretation of MCAR in terms of a causal graph reads as follows:

**Assumption 2** There does not exist an undirected path between any member of $O$ and any member of $\cup_{i=1}^q M_i$ in the underlying DAG.6

Now Assumption 2 states that the set $\cup_{i=1}^q M_i$ plays no role in the conditional dependence relations between the observables. Specifically:

**Lemma 3** Consider Assumption 2. Then $O_i \not\perp_d O_j | (W, S_{O_i O_j W})$ if and only if $O_i \not\perp_d O_j | (W, S)$.

**Proof** The proof follows trivially if $S_{O_i O_j W} = S$, so assume that we have $S_{O_i O_j W} \supset S$. Let $T_{O_i O_j W} = \{S \setminus S\}$. Then no member of $T_{O_i O_j W}$ is on any undirected path between

---

6 This MCAR interpretation implies that $\cup_{i=1}^q M_i \not\perp_d O$, so the interpretation is similar to the MCAR interpretation introduced in [10], where we have $\cup_{i=1}^q M_i \not\perp_d ((O \cup L \cup S) \setminus \cup_{i=1}^q M_i)$. 

Springer
Fig. 11  Test-wise deletion versus list-wise deletion as compared to five imputation methods in the MAR case when \( E(N) = 3 \). Same format as Fig. 4

\[ O_1 \text{ and } O_j \text{ by Assumption 2. Hence, no subset of } T_{O_1, O_j} W \text{ can be used to block an active path } \pi \text{ between } O_i \text{ and } O_j \text{. This proves the backward direction. Moreover, no subset of } T_{O_1, O_j} W \text{ can be used to activate any inactive path } \pi \text{ between } O_i \text{ and } O_j \text{. This proves the forward direction by contrapositive.} \]

\[ \square \]

The corresponding statement to Theorem 1 then reads as follows:

**Proposition 3**  Consider Assumption 2. Further assume \( d\)-separation faithfulness. Then FCI using only line 1 of Algorithm 1 outputs the same graph as FCI using a CI oracle with \( S \). The same result holds for RFCI.

**Proof**  It suffices to show that \( O_i \perp \!\!\!\!\perp O_j | (W, S) \) if and only if line 1 of Algorithm 1 outputs zero. This follows directly by Lemma 3 and \( d\)-separation faithfulness. \( \square \)

Notice, however, that Assumption 2 is much more difficult to justify in practice than the non-ancestral missingness assumption. We therefore do not recommend FCI or RFCI with only line 1 of Algorithm 1 in general, because these algorithms may not be sound when dealing with real data.

Heuristic test-wise deletion can nonetheless perform very well in the finite sample size case even when Assumption 2 is violated due to the extra boost in sample size provided by avoiding list-wise deletion altogether. We have summarized the simulation results in Figs. 6 and 7 in the MNAR case. Heuristic test-wise deletion outperforms test-wise deletion slightly by at most 0.203 SHD points on average (Fig. 6a, b). We could account for the increase in performance by a 5-15% increase in the average sample size per CI test compared to test-wise deletion (Fig. 6c, d). However, heuristic test-wise deletion generally underperforms test-wise deletion in skeleton discovery by a margin gradually increasing in sample size. This dichotomy between the overall SHD and the skeleton SHD occurs because accurate endpoint orientation requires more samples than accurate skeleton discovery in general. We conclude that while heuristic test-wise deletion usually outperforms test-wise deletion when taking endpoint orientations into account, the performance improvement is small.

The results for the MAR case follow similarly, as summarized in Fig. 8. Heuristic test-wise deletion claims an average lower rank than test-wise deletion due to a 5-20% increase in sample size in this scenario.

### 11.3 Test-wise deletion versus imputation

We have summarized the results of test-wise deletion versus imputation for the MNAR case in Fig. 9. Test-wise deletion

---

**International Journal of Data Science and Analytics (2018) 6:47–62 61**

**Fig. 11** Test-wise deletion versus list-wise deletion as compared to five imputation methods in the MAR case when \( E(N) = 3 \). Same format as Fig. 4
outperforms all imputation methods by a large margin in this regime.

11.4 Results with larger neighborhood sizes

We also report the synthetic data results for the MNAR and MAR cases when we set the expected neighborhood size to 3 in Figs. 10 and 11, respectively. Results are near identical to the results with $|E(N)| = 2$ as reported in Figs. 3 and 4.

References

1. Brand, J.: Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets. The Author (1999). https://books.google.com/books?id=Y0TywAAACAAJ
2. Colombo, D., Maathius, M., Kalisch, M., Richardson, T.: Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann. Stat. 40(1), 294–321 (2012). https://doi.org/10.1214/11-AOS940. http://projecteuclid.org/euclid.aos/133567191
3. Cranmer, S.J., Gill, J.: We have to be discrete about this: a non-parametric imputation technique for missing categorical data. Br. J. Polit. Sci. 43, 425–449 (2013). https://doi.org/10.1017/s0007123412000312
4. Daniel, R.M., Kenward, M.G., Cousens, S.N., De Stavola, B.L.: Using causal diagrams to guide analysis in missing data problems. Stoch. Models 21(3), 243–256 (2012)
5. Dooce, L., Van Buuren, S., Dusseldorp, E.: Recursive partitioning for missing data imputation in the presence of interaction effects. Comput. Stat. Data Anal. 72(C), 92–104 (2014)
6. Kowarik, A., Templ, M.: Imputation with the R package VIM. J. Stat. Softw. 74, 1–16 (2016). https://doi.org/10.18637/jss.v074.i07
7. Lauritzen, S.L., Dawid, A.P., Larsen, B.N., Leimer, H.G.: Independence properties of directed Markov fields. Networks 20(5), 491–505 (1990). https://doi.org/10.1002/net.323200503
8. Little, R.J.A.: Missing data adjustments in large surveys. J. Bus. Econ. Stat. 6, 287–296 (1988)
9. McCandl, J., Rodgers, W., Willis, R.: Cognition and aging in the USA (CogUSA) 2007–2009. Inter-university Consortium for Political and Social Research, Ann Arbor, MI (2015). https://doi.org/10.3886/ICPSR3053.v1
10. Mohan, K., Pearl, J., Tian, J.: Graphical models for inference with missing data. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26, pp. 1277–1285. Curran Associates, Inc., New York (2013)
11. Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. Wiley, New York (1987)
12. Schafer, J.: Analysis of Incomplete Multivariate Data. Chapman and Hall, London (1997)
13. Shah, A.D., Bartlett, J.W., Carpenter, J., Nicholas, O., Hemingway, H.: Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. Am. J. Epidemiol. 179(6), 764–774 (2014). https://doi.org/10.1093/aje/kwt312
14. Shipitser, I., Mohan, K., Pearl, J.: Missing data as a causal and probabilistic problem. In: Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, 12–16 July 2015, Amsterdam, The Netherlands, pp. 802–811 (2015)
15. Sokolova, E., Groot, P., Claassen, T., von Rhein, D., Buitelaar, J., Heskes, T.: Causal discovery from medical data: dealing with missing values and a mixture of discrete and continuous data. In: Artificial Intelligence in Medicine—Proceedings of 15th Conference on Artificial Intelligence in Medicine, AIME 2015, Pavia, Italy, 17–20 June 2015, pp. 177–181 (2015). https://doi.org/10.1007/978-3-319-19551-3_23
16. Sokolova, E., von Rhein, D., Naaijen, J., Groot, P., Claassen, T., Buitelaar, J., Heskes, T.: Handling hybrid and missing data in constraint-based causal discovery to study the etiology of ADHD. Int. J. Data Sci. Anal. 3(2), 105–119 (2017). https://doi.org/10.1007/s41060-016-0034-x
17. Spirtes, P.: An anytime algorithm for causal inference. In: In the Presence of Latent Variables and Selection Bias in Computation, Causation and Discovery, pp. 121–128. MIT Press, Cambridge (2001)
18. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, 2nd edn. MIT Press, Cambridge (2000)
19. Spirtes, P., Meek, C., Richardson, T.: An algorithm for causal inference in the presence of latent variables and selection bias. Computation, Causation, and Discovery, pp. 211–252. AAAI Press, Menlo Park, CA (1999)
20. Spirtes, P., Richardson, T.: A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In: Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics, Fort Lauderdale, pp. 489–500 (1996)
21. Strobl, E.V., Zhang, K., Visweswaran, S.: Approximate Kernel-Based Conditional Independence Tests for Fast Non-Parametric Causal Discovery (2017). http://arxiv.org/abs/1702.03877
22. Tillman, R.E., Danks, D., Glymour, C.: Integrating locally learned causal structures with overlapping variables. In: Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 8–11 Dec 2008, pp. 1665–1672 (2008)
23. Tillman, R.E., Eberhardt, F.: Learning causal structure from multiple datasets with similar variable sets. Behaviormetrika 41(1), 41–64 (2014)
24. Tillman, R.E., Spirtes, P.: Learning equivalence classes of acrylic models with latent and selection variables from multiple datasets with overlapping variables. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, 11–13 April 2011, pp. 3–15 (2011). http://www.jmlr.org/proceedings/papers/v15/tillman11a/tillman11a.pdf
25. Triantafilou, S., Tsamardinos, I., Tollis, I.G.: Learning causal structure from overlapping variable sets. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010, pp. 860–867 (2010). http://www.jmlr.org/proceedings/papers/v9/triantafilou10a.html
26. van Buuren, S.: Flexible Imputation of Missing Data (Chapman and Hall, CRC Interdisciplinary Statistics), 1st edn. Chapman and Hall, London (2012)
27. van Buuren, S., Brand, J.P.L., Groothuis-Oudshoorn, K.C., Rubin, D.B.: Fully conditional specification in multivariate imputation. J. Stat. Comput. Simul. (in press) (2005)
28. van Buuren, S., Groothuis-Oudshoorn, K.: Mice: multivariate imputation by chained equations in R. J. Stat. Softw. 45(3) (2011). https://www.jstatsoft.org/article/view/v045i03
29. Zhang, J.: On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. Artif. Intell. 172(16–17), 1873–1896 (2008). https://doi.org/10.1016/j.artint.2008.08.001