



Greedy structure learning from data that contain systematic missing values

Yang Liu¹ · Anthony C. Constantinou¹

Received: 15 July 2021 / Revised: 13 May 2022 / Accepted: 19 May 2022 /

Published online: 10 August 2022

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2022

Abstract

Learning from data that contain missing values represents a common phenomenon in many domains. Relatively few Bayesian Network structure learning algorithms account for missing data, and those that do tend to rely on standard approaches that assume missing data are missing at random, such as the Expectation-Maximisation algorithm. Because missing data are often systematic, there is a need for more pragmatic methods that can effectively deal with data sets containing missing values not missing at random. The absence of approaches that deal with systematic missing data impedes the application of BN structure learning methods to real-world problems where missingness are not random. This paper describes three variants of greedy search structure learning that utilise pairwise deletion and inverse probability weighting to maximally leverage the observed data and to limit potential bias caused by missing values. The first two of the variants can be viewed as subversions of the third and best performing variant, but are important in their own in illustrating the successive improvements in learning accuracy. The empirical investigations show that the proposed approach outperforms the commonly used and state-of-the-art Structural EM algorithm, both in terms of learning accuracy and efficiency, as well as both when data are missing at random and not at random.

Keywords Expectation-maximisation · Inverse probability weighting · Missing data · Score-based learning · Structure learning

Editor: Manfred Jaeger.

✉ Yang Liu
yangliu@qmul.ac.uk

Anthony C. Constantinou
a.constantinou@qmul.ac.uk

¹ Bayesian Artificial Intelligence Research Lab, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

1 Introduction

The field of Bayesian Network (BN) structure learning represents a set of approaches that focus on recovering the conditional or causal relationships between variables from data. Structure learning can be divided into two main categories known as constraint-based and score-based methods. Constraint-based methods such as PC (Spirtes et al. 2000) and IAMB (Tsamardinos et al. 2003) recover a graph by ruling out the structures that violate the conditional independencies discovered from data, and orientating edges by determining colliders. Score-based algorithms such as GES (Chickering 2002) and GOBNILP (Cussens 2011) recover a graph by exploring the search space of possible graphs and returning the graph with the highest objective score. While numerous BN structure learning algorithms have been proposed in the literature over the past few decades, most of them do not efficiently learn from data that contain systematic missing values. This hinders the application of structure learning to real-world problems, since missing data represents a common issue in most applied areas including medicine and healthcare (Constantinou et al. 2016), clinical epidemiology (Pedersen et al. 2017), traffic flow prediction (Tian et al. 2018), anomaly detection (Zemichael and Dietterich 2019), and financial analysis (John et al. 2019). Therefore, there is a greater need for structure learning algorithms that account for potential data bias due to systematic missing values, without having significant impact on the computational efficiency of structure learning.

According to Rubin (1976), missing data problems can be categorised into three classes. These are the Missing Completely At Random (MCAR), the Missing At Random (MAR) and the Missing Not At Random (MNAR). Specifically, MCAR denotes that the missing values are purely random and independent of other observed variables or parameters. This type of missingness is usually caused by technical error that would not bias the analysis. The definition of MAR, on the other hand, is somewhat counterintuitive in its name and assumes the missing values are dependent on observed data. For example, in an investigation between age and frequency of smoking, missing data are MAR if younger respondents are more likely to not disclose their smoking frequency. Lastly, data missingness are said to be MNAR if it is neither MCAR nor MAR. In the above example, the missingness are MNAR if data on respondent's age also contains missing values.

Methods that deal with missing data typically include naïve approaches such as the complete case analysis (a.k.a list-wise deletion) and multiple imputation (Rubin 2004). Complete case analysis involves removing the data cases that contain missing values and hence, restricting learning to complete data cases. Clearly, while this approach is easy to implement, it can be sample inefficient and may yield bias when missingness are not MCAR (Graham 2009). Multiple imputation, on the other hand, fills - rather than ignoring - the missing values and takes the uncertainty of imputation into consideration by repeating imputation over different possible values (Azur et al. 2011). However, multiple imputation is built under the assumption of MAR which means it may also produce biased outcomes when data are MNAR.

One of the earliest advanced approaches for dealing with missing data is the Expectation-Maximisation (EM) algorithm, which was also later adopted by the structure learning community. The Structural EM algorithm (Friedman 1997) is an iterative process which consists of two steps: the Expectation (E) step and the Maximisation (M) step. In E step, Structural EM makes inferences on the missing values and computes the expected sufficient statistics based on the graph learned in previous iteration. The M step follows where the current state of the learned graph is revised based on the sufficient statistics obtained

at step E. An advantage of Structural EM is that it can be combined with different structure learning algorithms. A disadvantage, however, is that it is computationally inefficient due to the inference process that takes place at step E. Therefore, in practice, the E step of the Structural EM algorithm is usually implemented with single imputation, i.e., imputing the expectation of the missing values derived from the observed values. Ruggieri et al. (2020) compared the performance of the original Structural EM to that of the imputed-based Structural EM, and found that the latter achieves better performance in most of the simulation scenarios.

An increasing number of algorithms are recently proposed to improve structure learning from data containing missing values. In the case of score-based learning, two model selection methods have been proposed based on the likelihood function called Node-Average Likelihood (NAL) for discrete (Balov 2013) and conditional Gaussian BNs (Bodewes and Scutari 2021). While these methods are consistent with MCAR, they are not consistent with MAR or MNAR cases. In constraint-based learning, Strobl et al. (2018) treated missing values as a type of selection bias and showed that performing test-wise deletion during conditional independence (CI) tests represents a sound solution for the FCI algorithm (Spirtes et al. 2000). In the context of constraint-based learning, test-wise deletion is a process that deletes the data cases with missing values amongst the variables involved in a given CI test. Gain and Shpitser (2018) later show that replacing the standard CI test in PC with an Inverse Probability Weighting (IPW) (Horvitz and Thompson 1952) based CI test, enables PC to be applied to data sets which contain systematic missing values without loss of consistency. IPW is an approach to alleviate bias in data distributions by reweighting the data cases which we will describe in detail in Sect. 3. However, IPW CI testing assumes sufficient information of missingness, such as information about the parents of missingness and the total ordering of the missing indicators, which is unlikely to be known in practise. Tu et al. (2019) tried to address this issue by first predicting the parents of missingness using constraint-based learning, for every observed variable that contained missing values, and applying the IPW CI tests using the sufficient information obtained during the constraint-based learning phase.

In this paper, we propose three variants of the greedy search Hill-Climbing algorithm to investigate how they handle missing data values under different assumptions of missingness. These variants can be viewed as fusions between greedy search score-based learning, and the pairwise deletion and IPW methods discussed above that have been previously applied to constraint-based learning. The contribution of this paper is a novel structure learning algorithm suitable for structural learning from data that contain systematic missingness. The empirical results show that, under systematic missingness, the proposed algorithm outperforms the current state-of-the-art Structural EM algorithm, both in terms of learning accuracy and efficiency.

The paper is organised as follows: Sect. 2 provides necessary preliminary information that includes notation and background information, Sect. 3 describes the proposed algorithm, Sect. 4 presents the results, and we provide our concluding remarks in Sect. 5.

2 Preliminaries

In this paper, we consider discrete variables which we denote with uppercase letters (e.g., U , V), and the assignment of variable states with lowercase letters (e.g., u , v). We denote a set of variables with bold uppercase letters (e.g., \mathbf{U} , \mathbf{V}), and the assignment of a set of variable states with bold lowercase letters (e.g., \mathbf{u} , \mathbf{v}).

2.1 Bayesian network

A BN $\langle \mathcal{G}, P \rangle$ is a probabilistic graphical model that can be represented by a Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and a joint distribution P defined over \mathbf{V} , where $\mathbf{V} = \{V_1, \dots, V_n\}$ represents a set of random variables and \mathbf{E} represents a set of directed edges between pairs of variables. A BN entails the *Markov Condition* which states that for every variable V_i in \mathcal{G} , V_i is independent of all its non-descendants conditional on its parents. Given the Markov Condition, the joint distribution P can be factorised as follows:

$$P(V_1, \dots, V_n) = \prod_{i=1}^n P(V_i \mid \mathbf{Pa}_i), \quad (1)$$

where \mathbf{Pa}_i represents the parent-set of V_i in \mathcal{G} . Since this study focuses on discrete BNs, we assume that every variable follows an independent multinomial distribution given their parents. We also assume that the set of observed variables \mathbf{V} is *causally sufficient* (Spirtes et al. 2000) and this means that we assume there are no unobserved common causes between any of the variables in \mathbf{V} . In practice, this means that even though measurement error can be viewed as a hidden variable problem where nodes that contain any form of error must have a hidden parent that causes that error, we assume causal sufficiency such that the graphs reconstructed by SEM are DAGs that contain the observed variables only.

Because an observed distribution can be represented by multiple different DAGs, we work under the assumption that multiple DAGs can be statistically indistinguishable. A collection of DAGs that are statistically indistinguishable, and express the same joint distribution, is also known as a set of Markov equivalent DAGs often referred to as a Completely Partial DAG (CPDAG) (Spirtes et al. 2000). A CPDAG can be obtained from a DAG by (a) preserving all its v-structures, (b) preserving all the directed edges that would create a cycle or a new v-structure if reversed, and (c) converting the residual directed edges to undirected edges.

2.2 Hill climbing algorithm

For simplicity, we focus on the Hill-Climbing (HC) structure learning algorithm (Heckerman et al. 1995) which is a classic score-based learning algorithm that greedily searches the space of neighbouring graphs. It typically starts from an empty graph and explores the search space of graphs via edge additions, deletions and reversals that maximally improve the objective score. HC terminates when no neighbouring graph increases the objective score. HC is an approximate learning algorithm that returns a local maximum solution. However, it is acknowledged to be a computationally efficient algorithm that often outperforms other more

complex algorithms (Gámez et al. 2011; Constantinou et al. 2021). The pseudo-code of the standard HC structure learning algorithm is provided in Algorithm 1.

Algorithm 1 The Hill-Climbing structure learning algorithm

```

Input data set  $D$ 
Output learned DAG  $\mathcal{G}$ 
1: procedure HILL CLIMBING
2:    $\mathcal{G} \leftarrow$  empty graph
3:   repeat
4:      $\delta \leftarrow 0$ 
5:     repeat
6:       construct a neighbouring DAG  $\mathcal{G}_{nei}$  by adding, reversing or deleting an edge
       from  $\mathcal{G}$ 
7:       if  $S(\mathcal{G}_{nei} | D) - S(\mathcal{G} | D) > \delta$  then
8:          $\delta \leftarrow S(\mathcal{G}_{nei} | D) - S(\mathcal{G} | D)$ 
9:          $\mathcal{G}_{update} \leftarrow \mathcal{G}_{nei}$ 
10:      end if
11:    until all possible edge operations have been attempted
12:    if  $\delta > 0$  then
13:       $\mathcal{G} \leftarrow \mathcal{G}_{update}$ 
14:    end if
15:  until  $\delta = 0$ 
16: end procedure

```

As with most other structure learning algorithms, HC is usually paired with a decomposable score function to evaluate each graph explored relative to the input data. A score function $S(\mathcal{G}, D)$ is decomposable if it can be written as the sum over a set of local scores, each of which corresponds to a variable and its parents in \mathcal{G} . While all score-based algorithms can use a decomposable score, this property is particular efficient in the case of HC search since it explores one or two graphical modifications at a time; i.e., one in case of edge addition or removal, and two in the case of edge reversal. Therefore, the objective score for each neighbouring graph \mathcal{G}_{nei} can be obtained efficiently by only recomputing the local scores of up to two nodes whose parent-set has changed, and obtaining the local scores of the remaining nodes whose parent-set remains intact from the current best graph \mathcal{G} .

Many score functions offer the decomposable property, and most commonly include the *Bayesian Information Criterion* (BIC) (Schwarz 1978), the *Bayesian Dirichlet equivalent* (BDe) (Heckerman et al. 1995) and the *quotient Normalized Maximum Likelihood* (qNML) (Silander et al. 2018). In this paper, we employ BIC as the score function in all of our experiments. The formal definition of BIC is:

$$\begin{aligned}
 S_{BIC}(\mathcal{G} | D) &= \log L(\mathcal{G} | D) - \frac{\log(N)}{2} \cdot |\mathcal{G}| \\
 &= \sum_{i=1}^n \left(\log P(V_i | \mathbf{Pa}_i, \hat{\Theta}_i) - \frac{\log(N)}{2} \cdot |\hat{\Theta}_i| \right),
 \end{aligned} \tag{2}$$

where N is the sample size, \mathbf{Pa}_i is the parent set of V_i in \mathcal{G} , $\hat{\Theta}_i$ is the maximum likelihood estimates of the parameters over the local distribution of V_i , and $|\hat{\Theta}_i|$ is the number of free parameters in $\hat{\Theta}_i$. If \mathcal{G} is defined over a set of discrete multinomial variables $V = \{V_1, \dots, V_n\}$, then the BIC score has the following form:

$$S_{BIC}(\mathcal{G} \mid D) = \sum_{i=1}^n \left(\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \cdot \log \frac{N_{ijk}}{N_{ij}} - \frac{\log(N)}{2} \cdot (r_i - 1) q_i \right), \quad (3)$$

where N_{ijk} is the number of cases in data set D in which the variable V_i takes its k^{th} value and the parents of V_i take the j^{th} configuration. Similarly, N_{ij} is the number of cases in data set D where the parents of V_i take their j^{th} configuration and, therefore, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Lastly, r_i represents the number of distinct values of V_i and q_i represents the number of configurations of the parents of V_i .

2.3 Missing data assumptions

We adopt the graphical descriptions of missing data introduced by Mohan et al. (2013) and Mohan and Pearl (2021). In this paper, we denote the set of fully observed variables (i.e., variables without missing values) as V_o and the set of partially observed variables (i.e., variables with at least one missing values) as V_m . For every partially observed variable $V_i \in V_m$, we define an auxiliary variable R_i called missing indicator to reflect the missingness in V_i , where R_i takes the value of 0 when V_i is recorded and the value of 1 when V_i is missing.

Further, we define the *missingness graph* (m-graph (Mohan et al. 2013)) $\mathcal{G}(\mathbb{V}, E)$ that captures the relationships between observed variables V and missing indicators R , where $\mathbb{V} = V_o \cup V_m \cup R$. Based on m-graph, we define missing data as MCAR if $R \perp\!\!\!\perp V_o \cup V_m$, MAR if $R \perp\!\!\!\perp V_m \mid V_o$, otherwise MNAR. Figure 1 presents the three possible m-graphs assuming three observed variables with structure $V_1 \rightarrow V_2 \rightarrow V_3$, depicting the MCAR, MAR and MNAR assumptions respectively.

To ensure the population distributions are recoverable from the observed data, some assumptions need to be employed for the missing indicators. These are:

Assumption 1 Variables in R neither can be the parent of an observed variables in V nor other variables in R .

Assumption 2 No partially observed variable can be the parent of its own missing indicator.

Assumption 1 states that a missing indicator in R can only be an effect (leaf) node in an m-graph, whereas Assumption 2 states that the missing value is independent of the variable

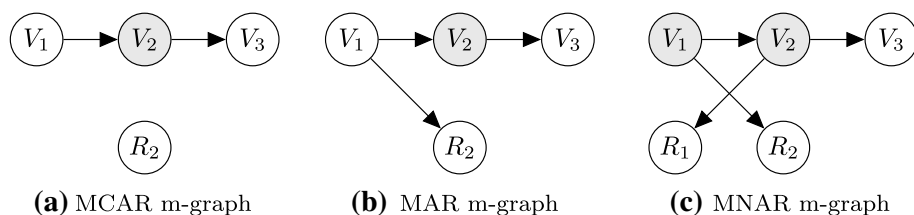


Fig. 1 The three possible m-graphs assuming three observed variables with structure $V_1 \rightarrow V_2 \rightarrow V_3$. Shaded nodes represent partially observed variables

value. When both Assumptions 1 and 2 hold, the joint distribution of the observed variables is recoverable from the observed data (Mohan et al. 2013, Theorem 2).

3 Handling systematic missing data with hill-climbing

This section describes the three HC variants that we explore in extending the learning process towards dealing with systematic missing data. Specifically, Sect. 3.1 describes HC with pairwise deletion which we call HC-pairwise, Sect. 3.2 describes HC with both pairwise deletion and Inverse Probability Weighting which we call HC-IPW, and Sect. 3.3 describes an improved version of HC-IPW, the HC-aIPW, that prunes less data samples compared to HC-IPW. The first two HC-variants can be viewed as sub-versions of HC-aIPW, but are important in their own in illustrating the successive improvements in learning accuracy.

3.1 Hill-climbing with pairwise deletion


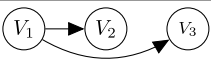
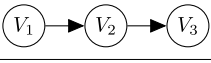
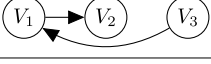
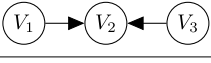
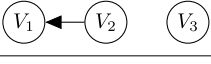
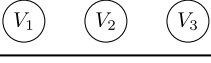
Recall that, at each iteration, HC moves to the neighbouring graph that maximally improves the objective score, and that performing HC search with a decomposable scoring function means that there is no need to recompute the local score of variables whose parent-set remains unchanged across graphs. Therefore, an efficient (but not necessarily effective) way of applying HC to missing data is to ignore data cases that contain missing values in variables that form part of the set of variables considered when exploring local score changes to a DAG. We refer to this process as *pairwise deletion*, where “pair” refers to the current pair of candidate DAGs (the current best DAG and neighbouring DAG), and this deletion process may involve more than two variables. When comparing the current best DAG against a neighbouring DAG, the *necessary variables* would be the nodes with unequal parent-sets between the two graphs, plus the parents of those nodes in the two graphs. Formally, when exploring a neighbouring DAG \mathcal{G}_{nei} from the current best DAG \mathcal{G} , the set of necessary variables W between \mathcal{G} and \mathcal{G}_{nei} can be described as:

$$W = \cup_{V_i \in V_d} \{V_i, \mathbf{Pa}_i, \mathbf{Pa}_i^{nei}\}, \quad (4)$$

where V_d is the set of variables that have different parent-sets between \mathcal{G} and \mathcal{G}_{nei} , and \mathbf{Pa}_i and \mathbf{Pa}_i^{nei} are the parent-sets of V_i in \mathcal{G} and \mathcal{G}_{nei} respectively. For simplicity, we refer to the data set obtained after applying pairwise deletion as the pairwise deleted data set.

Example 1 Assume that, during HC, the current state of DAG \mathcal{G} is a graph containing three variables $\{V_1, V_2, V_3\}$ and the edge $V_1 \rightarrow V_2$, as illustrated in Table 1. Given DAG \mathcal{G} , there are six possible edge operations each of which produces a neighbouring graph \mathcal{G}_{nei} . Operation add $V_1 \rightarrow V_3$, for example, can be evaluated by assessing the change in the local score of V_3 , i.e., $S(V_3 | V_1) - S(V_3)$, since V_3 is the only variable with different parents between \mathcal{G} and \mathcal{G}_{nei} . When the data set contains missing values, we can apply pairwise deletion to data given $\{V_1, V_3\}$ in order to obtain a complete data set that will enable us to assess the neighbouring graph resulting from this edge operation. However, there is a risk that this action may lead to biased estimates when missingness is not MCAR.

Table 1 Examples of necessary variables for each edge operation in HC, which we define as the variables with different parent-sets between the current best and neighbouring graphs, plus the parents that make up those parent-sets

current DAG state \mathcal{G}	edge operation	neighbouring DAG \mathcal{G}_{nei}	necessary variables
	add $V_1 \rightarrow V_3$		$\{V_1, V_3\}$
	add $V_2 \rightarrow V_3$		$\{V_2, V_3\}$
	add $V_3 \rightarrow V_1$		$\{V_1, V_3\}$
	add $V_3 \rightarrow V_2$		$\{V_1, V_2, V_3\}$
	reverse $V_1 \rightarrow V_2$		$\{V_1, V_2\}$
	delete $V_1 \rightarrow V_2$		$\{V_1, V_2\}$

Because pairwise deletion leads to edge operations that are assessed based on different subsets of the data, it is possible to get stuck in an infinite loop where previous neighbouring graphs are constantly revisited and re-selected as a higher scoring graph. This can happen when, for example, DAG \mathcal{G}_2 returns a higher score than \mathcal{G}_1 based on pairwise deleted data set D_1 , \mathcal{G}_3 returns a higher score than \mathcal{G}_2 based on pairwise deleted data set D_2 , and \mathcal{G}_1 returns a higher score than \mathcal{G}_3 based on pairwise deleted data set D_3 . In this example, HC with pairwise deletion would identify the graphical scores as $\mathcal{G}_1 < \mathcal{G}_2 < \mathcal{G}_3 < \mathcal{G}_1$ and never converge to a maximal solution. We address this issue by restricting HC search to neighbours not previously identified as the optimal graph. We call this variant of HC as HC-pairwise, and present its pseudo-code in Algorithm 2.

Algorithm 2 HC-pairwise algorithm

Input data set D
Output learned DAG \mathcal{G}

```

1: procedure HC-PAIRWISE
2:    $\mathcal{G} \leftarrow$  empty graph
3:    $\mathcal{G}_{record} \leftarrow \{\mathcal{G}\}$ 
4:   repeat
5:      $\delta \leftarrow 0$ 
6:     repeat
7:       construct a neighbouring DAG  $\mathcal{G}_{nei}$  by adding, reversing or deleting an edge
       from  $\mathcal{G}$ 
8:       if  $\mathcal{G}_{nei} \notin \mathcal{G}_{record}$  then
9:         construct  $D_{pw}$  by pairwise deleting  $D$  given the necessary variables  $\mathbf{W}$ 
10:        if  $S(\mathcal{G}_{nei} \mid D_{pw}) - S(\mathcal{G} \mid D_{pw}) > \delta$  then
11:           $\delta \leftarrow S(\mathcal{G}_{nei} \mid D_{pw}) - S(\mathcal{G} \mid D_{pw})$ 
12:           $\mathcal{G}_{update} \leftarrow \mathcal{G}_{nei}$ 
13:        end if
14:      end if
15:    until all possible edge operations have been attempted
16:    if  $\delta > 0$  then
17:       $\mathcal{G} \leftarrow \mathcal{G}_{update}$ 
18:       $\mathcal{G}_{record} = \mathcal{G}_{record} \cup \{\mathcal{G}\}$ 
19:    end if
20:  until  $\delta = 0$ 
21: end procedure

```

When data are MCAR, on the basis of $\mathbf{R} \perp\!\!\!\perp \mathbf{V}$ the distribution entailed by any pairwise deleted data set is an unbiased estimate of the underlying true distribution:

$$P(V_i \mid \mathbf{Pa}_i, \mathbf{R}_s = \mathbf{0}) = P(V_i \mid \mathbf{Pa}_i), \quad (5)$$

where \mathbf{R}_s can be any subset of \mathbf{R} .

From this, we derive Proposition 1, which states that, when the missingness is MCAR, the DAG learned by HC-pairwise is a local maximum graph, at least when BIC is used as the objective function. We define the local maximum graph as the graph with an objective score not lower than the scores of all its valid neighbouring graphs, when these scores are derived from the fully observed data set; i.e., it is independent of missingness generated.

Proposition 1 Assume data D is MCAR and sample size $N \rightarrow \infty$, for any DAG \mathcal{G} and one of its neighbouring DAG \mathcal{G}_{nei}

$$S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}) > S_{BIC}(\mathcal{G} \mid D_{pw}), \text{ iff } S_{BIC}(\mathcal{G}_{nei} \mid D_f) > S_{BIC}(\mathcal{G} \mid D_f),$$

where D_{pw} is the pairwise deleted data set which is derived from D by removing the data cases with missing values amongst the necessary variables \mathbf{W} , and D_f is the corresponding fully observed data set.

3.2 Hill-climbing with inverse probability weighting

Although HC-pairwise will progressively learn a better DAG after each iteration when missingness is MCAR, this property does not necessarily hold when missingness is MAR or MNAR, since systematic bias in the data might produce

$$P(V_i | \mathbf{Pa}_i, \mathbf{R} = \mathbf{0}) \neq P(V_i | \mathbf{Pa}_i). \quad (6)$$

To diminish data biases caused by potential dependencies between missing and observed data, we further explore applying the IPW method on the pairwise deleted data set.

According to Mohan et al. (2013, Theorem 2) and Tu et al. (2019), when Assumptions 1 and 2 hold, the joint distribution of variables V can be fully recovered from the observed part of the data set (i.e., the data after applying pairwise deletion) by

$$P(V) = P(V | \mathbf{R} = \mathbf{0}) \cdot \underbrace{\frac{P(\mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} P(R_i = 0 | \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}}_c \prod_{R_i \in \mathbf{R}} \underbrace{\frac{P(\mathbf{Pa}_{R_i} | \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}{P(\mathbf{Pa}_{R_i} | R_i = 0, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}}_{\beta_{R_i}}, \quad (7)$$

where \mathbf{Pa}_{R_i} is the set of parents of missing indicator R_i , and $\mathbf{R}_{\mathbf{Pa}_{R_i}}$ is the set of missing indicator of the partially observed variables in \mathbf{Pa}_{R_i} . We further discuss and provide the derivation of Eq. (7) in Appendix B.

Since the term c in Eq. (7) represents a constant value, we can apply pairwise deletion to the missing data cases of variables V and weight the pairwise deleted data set by $\prod_{R_i \in \mathbf{R}} \beta_{R_i}$. This will produce a weighted data set that approximates the unbiased distribution $P(V)$. We call this HC variant HC-IPW, and can be viewed as an extension of HC-pairwise that incorporates both the pairwise deletion and IPW methods. Unlike HC-pairwise, the HC-IPW algorithm can be used under the assumption the input data are MAR or MNAR, in addition to MCAR, to diminish data bias caused by systematic missing values.

It should be noted that when \mathbf{Pa}_{R_i} contains partially observed variables, Eq. (7) implies that $\mathbf{Pa}_{R_i} \subseteq V$; otherwise, the columns of \mathbf{Pa}_{R_i} in the pairwise deleted data set may contain missing values that will render the calculation of β_{R_i} invalid. The following example shows that it might be impossible to recover the underlying true distribution if any $\mathbf{Pa}_{R_i} \not\subseteq V$.

Example 2 Consider that Figure 1c is the true m-graph, the current best DAG \mathcal{G} in HC search is the one shown in Fig. 2a, and b presents one of its neighbouring DAGs, \mathcal{G}_{nei} . Since the difference in score between \mathcal{G}_{nei} and \mathcal{G} is $S(V_3 | V_1) - S(V_3)$, we need to ensure that missingness does not bias the estimate of distribution $P(V_1, V_3)$ when computing distributional score difference. If we apply pairwise deletion directly on the necessary variables $\{V_1, V_3\}$ and use Eq. (7) to recover $P(V_1, V_3)$. This will result in the following equation:

$$P(V_1, V_3) = P(V_1, V_3 | R_1 = 0) \frac{P(R_1 = 0)}{P(R_1 = 0)} \cdot \frac{P(V_2 | R_2 = 0)}{P(V_2 | R_1 = 0, R_2 = 0)}.$$

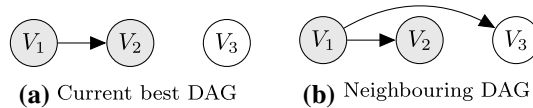


Fig. 2 A hill-climbing illustration of the DAG considered in Example 2, discussed in the main text. Shaded nodes represent partially observed variables

However, the problem in the above equation is that we cannot compute the weight term $\frac{P(V_2|R_2=0)}{P(V_2|R_1=0, R_2=0)}$ for data cases that contain missing values in V_2 .

To avoid this, when assessing the edge operations from \mathcal{G} to \mathcal{G}_{nei} in HC-IPW, the pairwise deletion for Eq. (7) should be performed on *sufficient variables* U , which is a variable set that contains the necessary variables W plus the parents of missing indicators of all variables in U :

$$U = \cup_{V_i \in V_d} \{V_i, \mathbf{Pa}_i, \mathbf{Pa}_i^{nei}\} \cup \mathbf{Pa}_{R_U}, \quad (8)$$

where V_d is the set of variables that have different parent-sets between \mathcal{G} and \mathcal{G}_{nei} , and \mathbf{Pa}_i and \mathbf{Pa}_i^{nei} are the parent-sets of V_i in \mathcal{G} and \mathcal{G}_{nei} respectively. It is worth noting that Eq. (8) represents a recursive process that iterates over the parents of missing indicators for all involved variables, i.e., not only W but also $U \setminus W$ should be included in U in order to resolve the issue illustrated in Example 2.

Another potential issue with Eq. (7) is that the parents \mathbf{Pa}_{R_i} of each missing indicator R_i are generally unknown. Tu et al. (2019) used constraint-based learning to discover the parents of each missing indicator, and this approach has been proven to be sound when both Assumptions 1 and 2 hold. We have, therefore, adopted the constraint-based approach proposed by Tu et al. (2019) to discover the parents of the missing indicators in applying HC-IPW. The intention here is that this approach can be used to exclude variable V_j as the parent of R_i , if R_i is found to be independent of V_j given any variable set S , given the pairwise deleted data set for $\{V_j\} \cup S$. Algorithm 3 provides the pseudo-code.

Algorithm 3 Discovering the parents of the missing indicators using constraint-based learning

Input data set D

Output the parents of missing indicators \mathbf{Pa}_R

```

1: procedure DETECTING PARENTS OF MISSING INDICATORS
2:   for each  $V_i \in V_m$  do
3:      $\mathbf{Pa}_{R_i} \leftarrow V \setminus V_i$ 
4:     for each  $V_j \in V \setminus V_i$  do
5:       remove  $V_j$  from  $\mathbf{Pa}_{R_i}$  if  $R_i \perp\!\!\!\perp V_j \mid S, R_j = 0, R_S = 0$ , for any  $S \subseteq \mathbf{Pa}_{R_i}$ 
6:     end for
7:   end for
8:   return  $\mathbf{Pa}_R$ 
9: end procedure
```

Algorithm 4 HC-IPW algorithm

Input data set D
Output learned DAG \mathcal{G}

```

1: procedure HC-IPW
2:    $\mathcal{G} \leftarrow$  empty graph
3:    $\mathcal{G}_{record} \leftarrow \{\mathcal{G}\}$ 
4:   retrieve the parents of missing indicators via Algorithm 3
5:   repeat
6:      $\delta \leftarrow 0$ 
7:     repeat
8:       construct a neighbouring DAG  $\mathcal{G}_{nei}$  by adding, reversing or deleting an edge
       from  $\mathcal{G}$ 
9:       if  $\mathcal{G}_{nei} \notin \mathcal{G}_{record}$  then
10:        construct  $D_{pw}$  by pairwise deleting  $D$  given the sufficient variables  $U$ 
11:        compute weight  $\beta$  by Equation 7 for  $D_{pw}$ 
12:        if  $S(\mathcal{G}_{nei} \mid D_{pw}, \beta) - S(\mathcal{G} \mid D_{pw}, \beta) > \delta$  then
13:           $\delta \leftarrow S(\mathcal{G}_{nei} \mid D_{pw}, \beta) - S(\mathcal{G} \mid D_{pw}, \beta)$ 
14:           $\mathcal{G}_{update} \leftarrow \mathcal{G}_{nei}$ 
15:        end if
16:      end if
17:    until all possible edge operations have been attempted
18:    if  $\delta > 0$  then
19:       $\mathcal{G} \leftarrow \mathcal{G}_{update}$ 
20:       $\mathcal{G}_{record} = \mathcal{G}_{record} \cup \{\mathcal{G}\}$ 
21:    end if
22:  until  $\delta = 0$ 
23: end procedure

```

Algorithm 4 describes the HC-IPW algorithm, where lines coloured in blue represent the difference in pseudo-code between HC-IPW and HC-pairwise. Note that when computing the objective score for HC-IPW, the weighted statistics $\tilde{N}_{ijk}, \tilde{N}_{ij}$ are used instead of the standard N_{ijk}, N_{ij} used in HC, and which are defined as follows:

$$\tilde{N}_{ijk} = \sum_{s=1}^{|D_{pw}|} 1_{ijk}(d^s) \cdot \beta^s, \quad (9)$$

$$\tilde{N}_{ij} = \sum_{k=1}^{r_i} \tilde{N}_{ijk}, \quad (10)$$

where 1_{ijk} is the indicator function of the event $(V_i = k, \mathbf{Pa}_i = j)$ which returns 1 when the combination of $V_i = k, \mathbf{Pa}_i = j$ appears in the input data case, and returns 0 otherwise, d^s is the s^{th} record in pairwise deleted data set D_{pw} , and β^s is the weight corresponding to d^s . Therefore, we define the BIC score for pairwise deleted data set D_{pw} given β as follows:

$$S_{BIC}(\mathcal{G} \mid D_{pw}, \beta) = \sum_{i=1}^n \left(\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \tilde{N}_{ijk} \cdot \log \frac{\tilde{N}_{ijk}}{\tilde{N}_{ij}} - \frac{\log(N_{pw})}{2} \cdot (r_i - 1)q_i \right),$$

where N_{pw} represents the sample size of D_{pw} , \tilde{N}_{ijk} and \tilde{N}_{ij} represent the weighted statistics as defined in Eqs. (9) and (10), and β is used for computing the weighted \tilde{N}_{ijk} and \tilde{N}_{ij} .

The following proposition shows that HC-IPW converges to a local optima when BIC is used as the score function, when both Assumptions 1 and 2 hold, and when sample size $N \rightarrow \infty$.

Proposition 2 *Given Assumptions 1 and 2, assume data D is partially observed and sample size $N \rightarrow \infty$, for any DAG \mathcal{G} and one of its neighbouring DAG \mathcal{G}_{nei}*

$$S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}, \beta) > S_{BIC}(\mathcal{G} \mid D_{pw}, \beta), \text{ iff } S_{BIC}(\mathcal{G}_{nei} \mid D_f) > S_{BIC}(\mathcal{G} \mid D_f),$$

where D_{pw} is the pairwise deleted data set which is derived from D by removing data cases with missing values among sufficient variables U , $\beta = \prod_{R_i \in \mathbf{R}_U} \beta_{R_i}$, and D_f is the corresponding fully observed data set.

3.3 Hill-climbing with adaptive inverse probability weighting

Although HC-IPW diminishes potential data bias caused by systematic missing values, the learning approach achieves this by removing a greater number of data cases compared to those removed by HC-pairwise when \mathbf{Pa}_{R_W} contains partially observed variables, which is likely to happen when the missingness are MNAR. This can be a problem when data cases are limited. We illustrate this phenomenon with an example.

Example 3 Suppose graph (a) in Fig. 3 represents the ground truth m-graph in which the variables in shaded backcolour V_1, V_4 and V_6 are partially observed whose missingness are caused by V_4, V_5 and V_1 respectively, as illustrated with the missing indicators R_1, R_4 and R_6 corresponding to the missingness of V_1, V_4 and V_6 . Let us assume graph (b) represents the current state of the optimal DAG in the HC-pairwise/HC-IPW search process, and that graphs (c) and (d) represent two of the possible neighbouring graphs. When HC-pairwise compares \mathcal{G} with \mathcal{G}_{n1} , it applies pairwise deletion on cases in which the necessary variables $\mathbf{W} = \{V_5, V_2, V_6\}$ contain missing values. Since only V_6 is partially observed out of the three necessary variables, HC-pairwise removes data cases when the value of V_6 is missing. In contrast, when HC-IPW is applied to this case, and assuming it correctly learns the parents of missingness via Algorithm 3, it computes the weights of the pairwise deleted data set through pairwise deletion based on the sufficient variables $\mathbf{U} = \{V_5, V_2, V_6\} \cup \{V_1, V_4, V_5\}$. Thus, HC-IPW removes data cases whenever any of the variables in \mathbf{U} contain a missing value (in this example, V_1, V_4 and V_6 do). Therefore, HC-IPW performs learning on a smaller set of data cases compared to those in the case of HC-pairwise.

When \mathbf{Pa}_{R_W} (refer to Eq. 4 and Algorithm 4) does not contain any partially observed variables, the HC-IPW algorithm will perform learning on the same number of data cases as in HC-pairwise. This can happen in cases such as when comparing neighbouring DAG \mathcal{G}_{n2} against \mathcal{G} in Fig. 3, where the set of necessary variables \mathbf{W} in HC-pairwise contains $\{V_4, V_2, V_5\}$ and the set of sufficient variables \mathbf{U} in HC-IPW is $\{V_4, V_2, V_5\} \cup \{V_5\}$. In this case, because V_5 is fully observed, applying pairwise deletion given \mathbf{W} and \mathbf{U} would result in the same pairwise deleted data set.

Because the effectiveness of a scoring function increases with sample size, the scoring efficiency of HC-IPW can decrease considerably when missingness are MNAR for

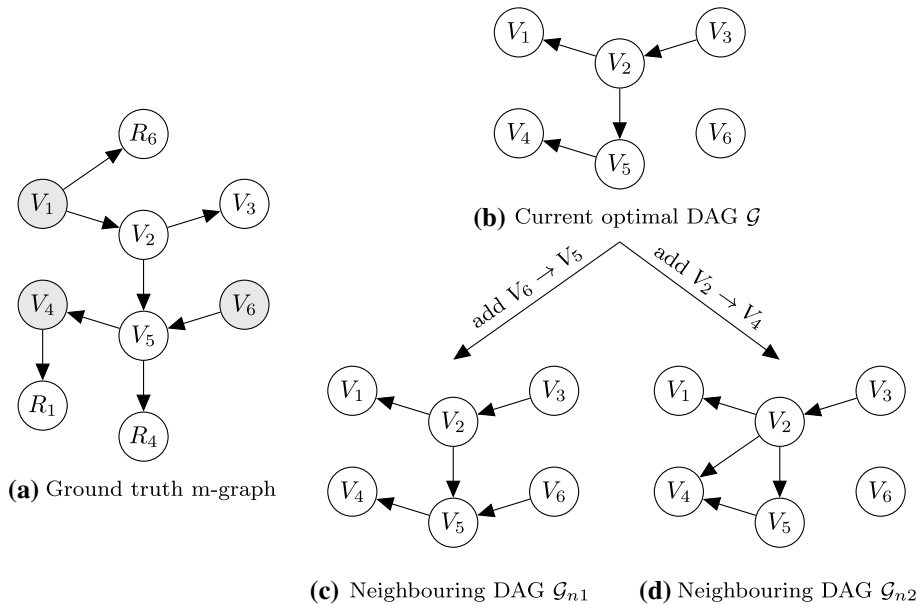


Fig. 3 Example of a searching step in HC-pairwise/HC-IPW

multiple variables. This is because both the number of partially observed variables and MNAR missingness increase the number of data cases removed during the learning process. It is on this basis we investigated a third variant, called the adaptive IPW-based HC (HC-aIPW), and which can be viewed as an extension of HC-IPW. The pseudo-code of HC-aIPW is shown in Algorithm 5. The highlighted section represents the part of the code that differs from HC-IPW.

In essence, HC-aIPW aims to maximise the samples taken into consideration during the learning process. When there are partially observed variables in \mathbf{Pa}_{R_w} , HC-aIPW applies pairwise deletion given \mathbf{W} and computes the difference in score between the current optimal DAG and the neighbouring DAG using the original pairwise deleted data set and standard scoring function. This is the only difference between HC-aIPW and HC-IPW. When there are no partially observed variables in \mathbf{Pa}_{R_w} , HC-aIPW uses the same IPW procedure as in HC-IPW to compute the difference in score between the current optimal DAG and the neighbouring DAG given the weighted pairwise deleted data set.

Algorithm 5 HC-aIPW algorithm

Input data set D
Output learned DAG \mathcal{G}

```

1: procedure HC-aIPW
2:    $\mathcal{G} \leftarrow$  empty graph
3:    $\mathcal{G}_{record} \leftarrow \{\mathcal{G}\}$ 
4:   retrieve the parents of missing indicators via Algorithm 3
5:   repeat
6:      $\delta \leftarrow 0$ 
7:     repeat
8:       construct a neighbouring DAG  $\mathcal{G}_{nei}$  by adding, reversing or deleting an edge
       from  $\mathcal{G}$ 
9:       if  $\mathcal{G}_{nei} \notin \mathcal{G}_{record}$  then
10:        if  $Pa_{RW} \cap V_m \neq \emptyset$  then
11:          construct  $D_{pw}$  by pairwise deleting  $D$  given the necessary variables  $W$ 
12:          if  $S(\mathcal{G}_{nei} | D_{pw}) - S(\mathcal{G} | D_{pw}) > \delta$  then
13:             $\delta \leftarrow S(\mathcal{G}_{nei} | D_{pw}) - S(\mathcal{G} | D_{pw})$ 
14:             $\mathcal{G}_{update} \leftarrow \mathcal{G}_{nei}$ 
15:          end if
16:        else
17:          construct  $D_{pw}$  by pairwise deleting  $D$  given the sufficient variables  $U$ 
18:          compute weight  $\beta$  by Equation 7 for  $D_{pw}$ 
19:          if  $S(\mathcal{G}_{nei} | D_{pw}, \beta) - S(\mathcal{G} | D_{pw}, \beta) > \delta$  then
20:             $\delta \leftarrow S(\mathcal{G}_{nei} | D_{pw}, \beta) - S(\mathcal{G} | D_{pw}, \beta)$ 
21:             $\mathcal{G}_{update} \leftarrow \mathcal{G}_{nei}$ 
22:          end if
23:        end if
24:      end if
25:    until all possible edge operations have been attempted
26:    if  $\delta > 0$  then
27:       $\mathcal{G} \leftarrow \mathcal{G}_{update}$ 
28:       $\mathcal{G}_{record} = \mathcal{G}_{record} \cup \{\mathcal{G}\}$ 
29:    end if
30:  until  $\delta = 0$ 
31: end procedure

```

4 Experiments

The learning accuracy of each of the three algorithms described in Sect. 3 is investigated and evaluated with reference to the Structural EM algorithm when applied to the same data. The Structural EM algorithm represents a state-of-the-art score-based approach for structure learning from missing data, and also explores the search space of graphs using HC. Since all the involved algorithms are based on HC, we measure their learning accuracy with reference to the results obtained when applying standard HC on complete, rather than incomplete, data. Results from complete data give us the empirical maximum performance we can achieve on these data sets using HC, before making part of the data missing. The HC and Structural EM algorithms used in this paper are those available in the *bnlearn* R package (Scutari 2010). It is worth noting that the Structural EM algorithm implemented in *bnlearn* R package is based on single imputation rather than belief propagation.

Therefore, the results presented in this paper approximate the difference between the proposed methods and Friedman's Structural EM. The implementations of the three HC variants described in Sect. 3 are available online at <https://github.com/Enderlogic/HC-missing-data>.

4.1 Generating synthetic data and missingness

To illustrate the performance of the algorithms under different settings, we consider three types of ground truth DAGs: sparse networks, dense networks and real-world networks. We have constructed 50 random sparse and 50 random dense DAGs. Each network contains 20 to 50 nodes with two to six states per node. A sparse DAG \mathcal{G} with n variables is generated from a randomly ordered variable set $V_1 < V_2 < \dots < V_n$, where directed edges are sampled from lower ordered variables to higher ordered variables with probability $2/(n-1)$. Dense DAGs are generated with the same procedure, but the probability of drawing an edge between variables increases to $4/(n-1)$. The conditional probability distribution of variable V_i in sparse and dense DAGs is parameterised, given any configuration of its parents, by drawing a random number from the Dirichlet distribution $\text{Dir}(\alpha)$, where $\alpha = \underbrace{\{1, \dots, 1\}}_{r_i}$, and r_i is the number of states in V_i . For real-world DAGs, we use the six

real-world BNs investigated in (Constantinou et al. 2021). The structure and parameters of these BNs are set by either real data observations or prior knowledge as defined in the original studies. The properties of these BNs are provided in Table 2.

We generate complete and incomplete synthetic data using the DAGs introduced above. The complete data sets are provided as input to the standard HC algorithm, whereas the corresponding incomplete data sets are provided as input to the Structural EM and the three HC variants described in Sect. 3. We generate five complete data sets per DAG with sample sizes $N \in \{100, 500, 1000, 5000, 10000\}$. Each complete data set is then used to construct further three data sets with missing values; one per missingness assumption, MCAR, MAR or MNAR. For the MCAR case, we randomly select 50% of the variables to represent the partially observed variables, and we then remove observed data of these variables with probability p , where p represents a random value between 0.1 and 0.6. For case MAR, we had to ensure missingness are dependent on a subset of the fully observed variables, and this is done as follows:

1. Randomly select 50% of the variables as partially observed variables (same process as in MCAR);

Table 2 The properties of the six real-world BNs

Name	Number of variables	Average degree	Number of states
Asia	8	2.00	2
Alarm	37	2.49	2 ~ 4
Pathfinder	109	3.58	2 ~ 63
Sports	9	3.33	3 ~ 8
ForMed	88	3.14	2 ~ 10
Property	27	2.30	2 ~ 7

2. Randomly assign a fully observed variable as the parent of missingness of a partially observed variable (repeat for all partially observed variables);
3. Remove observations in partially observed variables with probability $p = 0.6$ when the parent of their missingness is at its highest occurring state; otherwise, remove the observation with probability $p = 0.1$.

Generating MNAR data also involves the above 3-step procedure, but step 2 is modified as follows:

2. Randomly select 50% of the partially observed variables and randomly assign a fully observed variable as the parent of their missingness. For the remaining 50% partially observed variables, randomly assign another partially observed variable as the parent of their missingness.

4.2 Evaluation metrics

The structure learning performance is assessed using two metrics that are fully oriented towards graphical discovery. The first metric is the classic F_1 score, composed of *Precision* and *Recall*. The formal definition of the F_1 score is:

$$F_1 = 2 \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2 TP}{2 TP + FP + FN} \quad (11)$$

where TP is the number of edges that exist in both the learned graph and true graph, FP is the number of edges that exist in the learned graph but not in true graph, and FN is the number of edges that exist in the true graph but not in the learned graph.

The second metric considered is the Structural Hamming Distance (SHD) which measures graphical differences between the learned graph and the true graph (Tsamardinos et al. 2006). Specifically, the SHD score represents the number of edge operations needed to convert the learned graph to the true graph, where the edge operations involve arc addition, deletion and removal. Therefore, in contrast to the F_1 score, a lower SHD score indicates a better performance. Because the SHD score is sensitive to the number of edges and variables present in the true graph, we divide the SHD score by the number of edges in the true DAG to reduce bias.

Because the experiments are based on observational data, multiple DAGs can be statistically indistinguishable due to being part of the same Markov Equivalence class. On this basis, we compare the CPDAGs between the learned and true graphs to measure both the F_1 and SHD graphical scores.

4.3 Results when the true DAG is sparse

Figure 4 presents the average accuracy of the algorithms when the true DAGs are sparse. Each averaged score is derived from 50 CPDAGs, corresponding to each of the 50 randomly generated sparse DAGs. Appendix C provides the mean and standard deviation of the scores. The results suggest that the two evaluation metrics are generally consistent in ranking the algorithms from best to worst performance. Both metrics suggest that all of the three proposed HC variants outperform the Structural EM algorithm when the sample size is greater than 1,000, under all three missingness scenarios MCAR, MAR and MNAR.

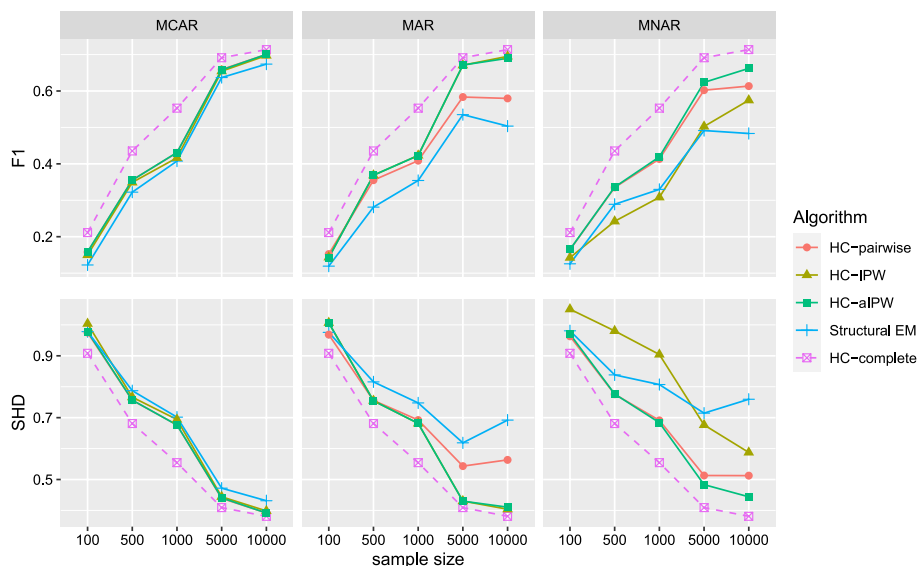


Fig. 4 Average F_1 and normalised SHD scores learned by HC-pairwise, HC-IPW, HC-aIPW and Structural EM for sparse networks, under different assumptions of missingness and sample sizes. Each score represents the average score over 50 CPDAGs. Note the scores of HC-complete are based on complete data for benchmarking purposes; i.e., the same scores are superimposed in all three missingness cases as a dashed line

Interestingly, the HC-aIPW algorithm almost matches the performance of HC which is applied to complete data (denoted as HC-complete in Fig. 4), particularly for experiments with 10,000 sample size, and this observation is consistent across all three missingness assumptions.

The three variants, HC-pairwise, HC-IPW and HC-aIPW, produce very similar results under MCAR, and this is because missingness under MCAR has no pattern that could be identified by the HC-IPW and HC-aIPW variants. That is, when HC-IPW and HC-aIPW do not discover any parent of missingness, they follow the search process of HC-pairwise. Under MAR, however, both HC-IPW and HC-aIPW outperform HC-pairwise as well as Structural EM when the sample size is larger than 100 and the improvement in performance increases with sample size. From this observation, we can conclude that the IPW method successfully eliminate most of the distributional bias. Interestingly, although the construction of the Structural EM algorithm is based on the MAR assumption, its performance under MAR is considerably lower than its performance under MCAR. A possible explanation is that the single imputation process the *bnlearn* R package employs during the E step of Structural EM, instead of belief propagation, is unable to capture the uncertainty of the missing values.

Lastly, the results under MNAR suggest that HC-IPW generally performs worse than HC-pairwise across most sample sizes. This observation can be explained by the reduced sample size on which HC-IPW operates, relative to HC-pairwise, as discussed in Sect. 3.3. Specifically, when the parents of missingness of necessary variables W contain partially observed variables (i.e., MNAR case), HC-IPW applies pairwise deletion by taking into consideration a higher number of variables compared to those considered by HC-pairwise. This means that, compared to HC-pairwise, the HC-IPW algorithm typically evaluates

edge operations based on smaller samples when missingness are MNAR, which tends to yield less accurate results. From this, we can also conclude that the negative effect resulting from HC-IPW further pruning samples has not been offset by the data bias adjustments applied by the IPW method. On the other hand, the HC-aIPW algorithm which is designed to apply the IPW method only when no additional samples would be deleted compared to HC-pairwise, generally outperforms all other algorithms under MNAR, particularly under higher sample sizes.

Figure 5 presents the relative execution time between (a) the four algorithms applied to data with missing values, and (b) the HC algorithm applied to the complete data. Because the three HC variants are implemented in Python, we measure their execution time relative to our Python version of HC. On the other hand, Structural EM is implemented in *bnlearn* R package and makes use of the HC implementation of that package. Therefore, the execution time of Structural EM is measured relative to the HC implementation in *bnlearn* R package. The mean and standard deviation of the results can be found in Appendix D.

Overall, the results show that HC-pairwise is the most efficient algorithm for missingness. Specifically, HC-pairwise increases execution time relative to HC by approximately 50%, while HC-IPW and HC-aIPW are anywhere between 8 and 15 times slower than HC dependent on sample size, and the relative difference in execution time tends to increase with sample size. This is because a higher number of parents of missingness are likely to be detected in larger sample sizes, and these discoveries increase execution time for IPW-based variants. Still, both the HC-IPW and HC-aIPW variants are more efficient than Structural EM which increases execution time relative to HC by 100 to 700 times.

4.4 Results when the true DAG is dense

In this subsection we investigate the performance of the algorithms when applied to data sets sampled from dense networks. The performance of each algorithm is depicted in Fig. 6, and detailed results are provided in Appendix C. An important distinction between sparse and dense networks is that learning from data sampled from dense networks makes it more likely that local parts of the graph will involve learning from partially observed variables. In other words, the effect of missing values is more severe on dense, compared to sparse, networks as shown in Sect. 4.3.

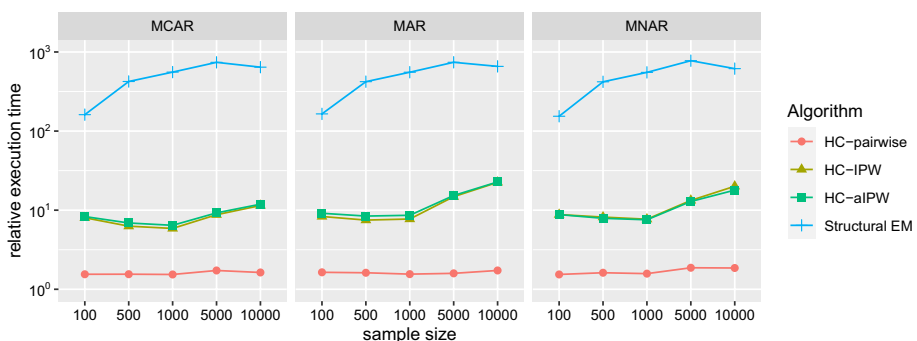


Fig. 5 Average ratio of the execution time between the algorithms running on missing data sets and HC running on complete data sets

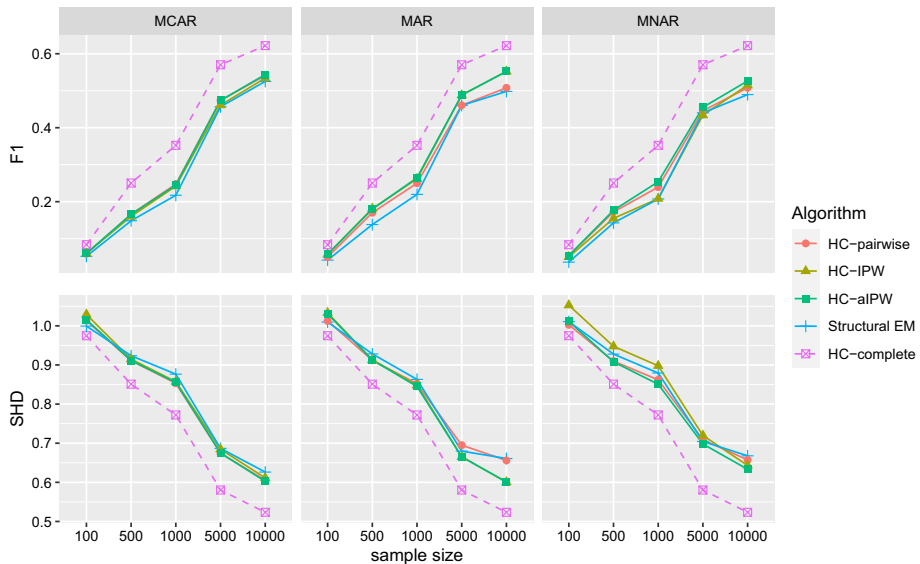


Fig. 6 Average F_1 and normalised SHD scores learned by HC-pairwise, HC-IPW, HC-aIPW and Structural EM for dense networks, under different assumptions of missingness and sample sizes. Each score represents the average score over 50 CPDAGs. Note the scores of HC-complete are based on complete data for benchmarking purposes; i.e., the same scores are superimposed in all three missingness cases as a dashed line

The results show that the HC-aIPW algorithm continues to perform best in the case of denser graphs, in terms of overall performance and over the different missingness and sample size assumptions. Specifically, HC-aIPW achieves the highest accuracy in 11 and 8 cases in terms of F_1 and SHD measures respectively, out of the 15 experiments conducted in this subsection. In contrast, the Structural EM algorithm performs best only in two experiments and only in SHD score. However, compared with the results in Sect. 4.3, the divergence in score between Structural EM and HC-based variants is much smaller.

The performance across the three HC-based variants appears to be similar to that obtained under sparse graphs. When data are MCAR, HC-IPW and HC-aIPW produce scores that are similar to those produced by HC-pairwise, and this is expected since no observed variables should be detected as the parents of missing indicators when missingness is MCAR. When data are MAR, both HC-IPW and HC-aIPW outperform HC-pairwise since, unlike HC-pairwise, they can detect and reduce bias caused by missing values. Lastly, when data are MNAR, HC-IPW performs worst amongst all algorithms, particularly when the sample size is lowest, and this is because it tends to remove a large number of data cases when computing the local scores. On the other hand, HC-aIPW (which aims to resolve this specific drawback of HC-IPW) performs best in almost all MNAR experiments. The consistency of the results across sparse and dense networks suggests that the performance of HC-aIPW, relative to the other algorithms considered in this study, is not sensitive to the sparsity of the network that generates the input data.

4.5 Results when the true DAG is a real-world network

Lastly, we apply the algorithms to data sets sampled from the six real-world networks. Figure 7 shows the average performance of the algorithms across all the six real-world networks and over all the five sample sizes. When the missingness is MCAR, the three HC-based variants achieve similar accuracy, as expected, and generally outperform the Structural EM algorithm when the sample size is larger than 500. When the missingness is MAR or MNAR, the performance of HC-aIPW improves over the other algorithms, especially when the sample size is larger than 500. These results are consistent with those obtained from the randomised sparse and dense networks presented in Sects. 4.3 and 4.4 respectively.

5 Conclusion

Learning accurate BN structure from incomplete data remains a challenging task. Most BN structure learning algorithms do not support learning from incomplete data, and this is partly explained by the considerable increase in computational complexity when dealing with incomplete data. The increased computational complexity caused by missing data adds to a problem that is NP-hard even when data are complete. This challenge is even greater when missing values are systematic rather than random.

In this paper, we have investigated three novel HC-based variants that employ pairwise deletion and IPW strategies to deal with random and systematic missing data. The

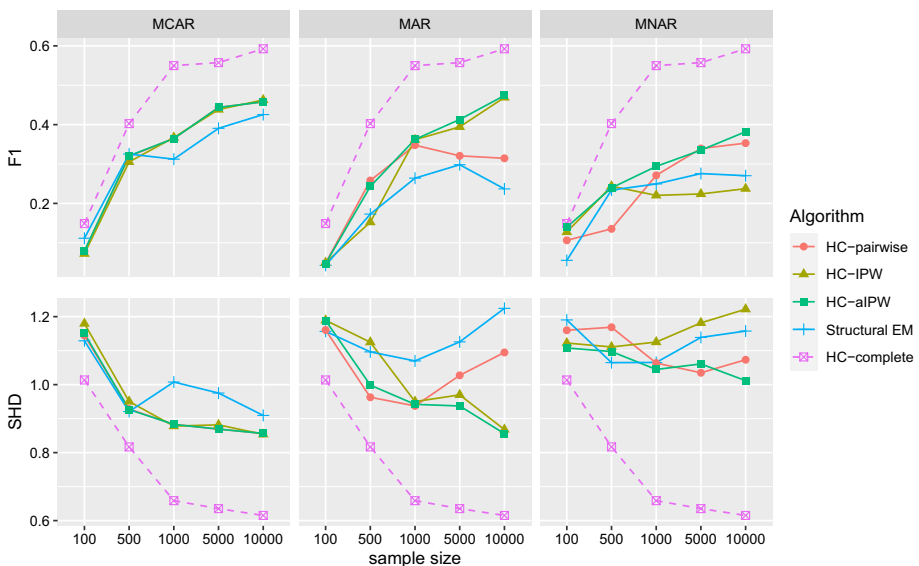


Fig. 7 Average F_1 and normalised SHD scores learned by HC-pairwise, HC-IPW, HC-aIPW and Structural EM for real-world networks, under different assumptions of missingness and sample sizes. Each score represents the average score over 50 CPDAGs. Note the scores of HC-complete are based on complete data for benchmarking purposes; i.e., the same scores are superimposed in all three missingness cases as a dashed line

HC-pairwise and HC-IPW variants can be viewed as subversions of HC-aIPW, which is the most complete and best performing variant described in this paper. All of the three variants have been applied to different cases of data missingness, and their performance was compared to the state-of-the-art Structural EM algorithm that is available in the *bnlearn* R package. Moreover, all performances under missingness have been compared to HC when applied to the corresponding complete data sets. The empirical results show:

1. Pairing HC with pairwise deletion (i.e., the HC-pairwise variant) is enough to learn graphs that are more accurate, as well as less computationally expensive, compared to the graphs produced by the Structural EM algorithm.
2. Combining HC with both pairwise deletion and IPW techniques (i.e., the HC-IPW variant) further improves learning accuracy under MCAR and MAR, in general, but decreases accuracy under MNAR due to aggressive pruning employed by HC-IPW on the data cases (refer to Sect. 3.3). Moreover, HC-IPW becomes considerably slower than HC-pairwise, although it remains an order of magnitude faster than Structural EM.
3. The HC-aIPW takes advantage of both strategies, as in HC-IPW, but relaxes the pruning strategy on the data cases and returns the overall best performance, especially under MNAR which represents the most difficult case of missingness.
4. All three HC variants described in this paper outperform Structural EM in most cases. Importantly, the performance of HC-aIPW on missing data approaches the performance of HC on complete data when sample size is 10,000 and the ground truth graph is sparse, and this observation is consistent under all three cases of missingness.

Future research will investigate the application of these learning strategies to search algorithms that are more complex than HC, such as Tabu, or other variants of HC such as the GES algorithm (Chickering 2002) which explores the CPDAG, rather than DAG space. Another possible research direction would be to combine the IPW method with the NAL score (Balov 2013), which is a scoring function intended for missingness under MCAR, and further investigate the possibility of a new decomposable scoring function under systematic missingness cases of MAR and MNAR.

Appendix A Proofs of propositions

In this section, we provide proofs of the propositions discussed in Sect. 3. We define the variables used in proofs as follows: V_d is the set of variables with different parent-sets between a given DAG \mathcal{G} and its neighbouring DAG \mathcal{G}_{nei} , \mathbf{W} is a set of the necessary variables as defined in Eq. (4), \mathbf{U} is a set of the sufficient variables defined in Eq. (8), and N and N_{pw} are the sample sizes of the partially observed data set D and pairwise deleted data set D_{pw} respectively.

Proposition 1 Assume data D is MCAR and sample size $N \rightarrow \infty$, for any DAG \mathcal{G} and one of its neighbouring DAG \mathcal{G}_{nei}

$$S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}) > S_{BIC}(\mathcal{G} \mid D_{pw}), \text{ iff } S_{BIC}(\mathcal{G}_{nei} \mid D_f) > S_{BIC}(\mathcal{G} \mid D_f),$$

where D_{pw} is the pairwise deleted data set which is derived from D by removing the data cases with missing values amongst the necessary variables \mathbf{W} , and D_f is the corresponding fully observed data set.

$$\begin{aligned}
& S_{BIC}(\mathcal{G}_{nei} \mid D_f) - S_{BIC}(\mathcal{G} \mid D_f) \\
&= \sum_{i=1}^n (S_{BIC}(V_i \mid \mathbf{Pa}_i^{nei}) - S_{BIC}(V_i \mid \mathbf{Pa}_i)) \\
&= \sum_{i: V_i \in V_d} (S_{BIC}(V_i \mid \mathbf{Pa}_i^{nei}) - S_{BIC}(V_i \mid \mathbf{Pa}_i)) \\
&= \sum_{i: V_i \in V_d} \left(\sum_{D_f} (\log P(V_i \mid \mathbf{Pa}_i^{nei}) - \log P(V_i \mid \mathbf{Pa}_i)) \right. \\
&\quad \left. + \frac{\log(N)}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&= \frac{N}{N_{pw}} \sum_{i: V_i \in V_d} \left(\sum_{D_{pw}} (\log P(V_i \mid \mathbf{Pa}_i^{nei}, \mathbf{R}_W = \mathbf{0}) - \log P(V_i \mid \mathbf{Pa}_i, \mathbf{R}_W = \mathbf{0})) \right. \\
&\quad \left. + \frac{\log(N_{pw})}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) + \frac{\log(N/N_{pw})}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right)
\end{aligned} \tag{12}$$

Proof

$$\begin{aligned}
&= \frac{N}{N_{pw}} \left(S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}) - S_{BIC}(\mathcal{G} \mid D_{pw}) \right. \\
&\quad \left. + \frac{\log(N/N_{pw})}{2} \sum_{i: V_i \in V_d} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&\propto S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}) - S_{BIC}(\mathcal{G} \mid D_{pw}) + O(1)
\end{aligned} \tag{13}$$

Equation (12) follows from Eq. (5) given the MCAR assumption and large sample limit. Equation (13) is due to the missing rate of data D , i.e., N_{pw}/N , does not relate to the sample size N and remains constant with the increase of N . \square

Proposition 2 Given Assumptions 1 and 2, assume data D is partially observed and sample size $N \rightarrow \infty$, for any DAG \mathcal{G} and one of its neighbouring DAG \mathcal{G}_{nei}

$$S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}, \beta) > S_{BIC}(\mathcal{G} \mid D_{pw}, \beta), \text{ iff } S_{BIC}(\mathcal{G}_{nei} \mid D_f) > S_{BIC}(\mathcal{G} \mid D_f),$$

where D_{pw} is the pairwise deleted data set which is derived from D by removing data cases with missing values among sufficient variables \mathbf{U} , $\beta = \prod_{R_i \in \mathbf{R}_U} \beta_{R_i}$, and D_f is the corresponding fully observed data set.

$$\begin{aligned}
& S_{BIC}(\mathcal{G}_{nei} \mid D_f) - S_{BIC}(\mathcal{G} \mid D_f) \\
&= \sum_{i: V_i \in V_d} \left(\sum_{D_f} (\log P(V_i \mid \mathbf{Pa}_i^{nei}) - \log P(V_i \mid \mathbf{Pa}_i)) \right. \\
&\quad \left. + \frac{\log(N)}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&= \sum_{i: V_i \in V_d} \left(\sum_{D_f} \left(\log \frac{P(V_i, \mathbf{Pa}_i^{nei})}{\sum_{V_i} P(V_i, \mathbf{Pa}_i^{nei})} - \log \frac{P(V_i, \mathbf{Pa}_i)}{\sum_{V_i} P(V_i, \mathbf{Pa}_i)} \right) \right. \\
&\quad \left. + \frac{\log(N)}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&= \frac{N}{N_{pw}} \sum_{i: V_i \in V_d} \left(\sum_{D_{pw}} \left(\log \frac{P(V_i, \mathbf{Pa}_i^{nei} \mid \mathbf{R}_U = \mathbf{0})\beta}{\sum_{V_i} P(V_i, \mathbf{Pa}_i^{nei} \mid \mathbf{R}_U = \mathbf{0})\beta} \right. \right. \\
&\quad \left. \left. - \log \frac{P(V_i, \mathbf{Pa}_i \mid \mathbf{R}_U = \mathbf{0})\beta}{\sum_{V_i} P(V_i, \mathbf{Pa}_i \mid \mathbf{R}_U = \mathbf{0})\beta} \right) + \frac{\log(N)}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&= \frac{N}{N_{pw}} \sum_{i: V_i \in V_d} \left(\sum_{j=1}^{|\mathbf{Pa}_i^{nei}|} \sum_{k=1}^{|V_i|} \tilde{N}_{ijk} \log \frac{\tilde{N}_{ijk}}{\tilde{N}_{ij}} - \sum_{j=1}^{|\mathbf{Pa}_i|} \sum_{k=1}^{|V_i|} \tilde{N}_{ijk} \log \frac{\tilde{N}_{ijk}}{\tilde{N}_{ij}} \right. \\
&\quad \left. + \frac{\log(N)}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \tag{14} \\
&= \frac{N}{N_{pw}} \sum_{i: V_i \in V_d} \left(\sum_{j=1}^{|\mathbf{Pa}_i^{nei}|} \sum_{k=1}^{|V_i|} \tilde{N}_{ijk} \log \frac{\tilde{N}_{ijk}}{\tilde{N}_{ij}} - \sum_{j=1}^{|\mathbf{Pa}_i|} \sum_{k=1}^{|V_i|} \tilde{N}_{ijk} \log \frac{\tilde{N}_{ijk}}{\tilde{N}_{ij}} \right. \\
&\quad \left. + \frac{\log(N_{pw})}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) + \frac{\log(N/N_{pw})}{2} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&= \frac{N}{N_{pw}} \left(S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}, \beta) - S_{BIC}(\mathcal{G} \mid D_{pw}, \beta) \right. \\
&\quad \left. + \frac{\log(N/N_{pw})}{2} \sum_{i: V_i \in V_d} (|\hat{\Theta}_i^{nei}| - |\hat{\Theta}_i|) \right) \\
&\propto S_{BIC}(\mathcal{G}_{nei} \mid D_{pw}, \beta) - S_{BIC}(\mathcal{G} \mid D_{pw}, \beta) + O(1)
\end{aligned}$$

Proof

In the above equations, $\beta = \prod_{R_i \in \mathbf{R}_U} \beta_{R_i}$, \tilde{N}_{ijk} and \tilde{N}_{ij} are defined by Eqs. (9) and (10). Equation (14) is a consequence of the recoverability of $P(U)$ given Eq. (7). \square

Appendix B Derivation of Eq. (7)

Based on Mohan et al. (2013), Theorem 2, given Assumptions 1 and 2, the joint distribution $P(V)$ can be fully recovered from the observed data via the following equation:

$$P(V) = \frac{P(V, \mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} P(R_i = 0 \mid \mathbf{Pa}_{R_i}, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}$$

where \mathbf{Pa}_{R_i} is the set of parents of missing indicator R_i , and $\mathbf{R}_{\mathbf{Pa}_{R_i}}$ is the set of missing indicator of the partially observed variables in \mathbf{Pa}_{R_i} . Then,

$$\begin{aligned} P(V) &= \frac{P(V, \mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} P(R_i = 0 \mid \mathbf{Pa}_{R_i}, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})} \\ &= \frac{P(V \mid \mathbf{R} = \mathbf{0})P(\mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} P(R_i = 0 \mid \mathbf{Pa}_{R_i}, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})} \\ &= P(V \mid \mathbf{R} = \mathbf{0}) \cdot \frac{P(\mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} \frac{P(\mathbf{Pa}_{R_i} \mid R_i = 0, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})P(R_i = 0 \mid \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}{P(\mathbf{Pa}_{R_i} \mid \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}} \\ &= P(V \mid \mathbf{R} = \mathbf{0}) \cdot \underbrace{\frac{P(\mathbf{R} = \mathbf{0})}{\prod_{R_i \in \mathbf{R}} P(R_i = 0 \mid \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}}_c \underbrace{\prod_{R_i \in \mathbf{R}} \frac{P(\mathbf{Pa}_{R_i} \mid \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}{P(\mathbf{Pa}_{R_i} \mid R_i = 0, \mathbf{R}_{\mathbf{Pa}_{R_i}} = \mathbf{0})}}_{\beta_{R_i}} \end{aligned}$$

In the above equation, the term c depends only on the missing indicators \mathbf{R} and remains constant with respect to the observed variables \mathbf{V} . The product $\prod_{R_i \in \mathbf{R}} \beta_{R_i}$ represents the relative probability of a data case from the pairwise deleted data set being observed in the complete data set. For example, if a pairwise deleted data case has $c \prod_{R_i \in \mathbf{R}} \beta_{R_i}$ out of 0.8, then its occurrence rate is assumed to drop by 20% in the complete data set compared to its occurrence rate in the pairwise deleted data set. Therefore, we use Eq. (7) to reweight the pairwise deleted data and estimate the underlying true distribution given the pairwise deleted data set.

Appendix C Supplementary results from the structure learning experiments

Refer to Tables 3, 4, 5, 6, 7 and 8.

Table 3 Mean and standard deviation of F_1 scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for sparse networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	0.122 ± 0.088	0.158 ± 0.108	0.150 ± 0.104	0.159 ± 0.107
	500	0.325 ± 0.139	0.356 ± 0.139	0.349 ± 0.133	0.355 ± 0.139
	1000	0.410 ± 0.141	0.430 ± 0.149	0.417 ± 0.143	0.431 ± 0.150
	5000	0.642 ± 0.149	0.659 ± 0.144	0.654 ± 0.160	0.658 ± 0.144
	10000	0.682 ± 0.135	0.700 ± 0.140	0.697 ± 0.143	0.700 ± 0.137
MAR	100	0.117 ± 0.097	0.152 ± 0.102	0.143 ± 0.102	0.142 ± 0.101
	500	0.281 ± 0.119	0.355 ± 0.117	0.369 ± 0.140	0.369 ± 0.138
	1000	0.354 ± 0.136	0.409 ± 0.152	0.423 ± 0.150	0.423 ± 0.149
	5000	0.543 ± 0.119	0.583 ± 0.137	0.671 ± 0.147	0.671 ± 0.150
	10000	0.505 ± 0.141	0.580 ± 0.121	0.695 ± 0.136	0.690 ± 0.138
MNAR	100	0.127 ± 0.094	0.164 ± 0.098	0.143 ± 0.103	0.165 ± 0.099
	500	0.285 ± 0.122	0.335 ± 0.129	0.242 ± 0.091	0.336 ± 0.131
	1000	0.328 ± 0.123	0.413 ± 0.142	0.308 ± 0.113	0.419 ± 0.148
	5000	0.488 ± 0.137	0.602 ± 0.164	0.503 ± 0.124	0.624 ± 0.150
	10000	0.473 ± 0.148	0.613 ± 0.144	0.575 ± 0.157	0.662 ± 0.146

Table 4 Mean and standard deviation of normalised SHD scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for sparse networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	0.978 ± 0.063	0.975 ± 0.095	1.004 ± 0.107	0.978 ± 0.093
	500	0.784 ± 0.121	0.756 ± 0.127	0.766 ± 0.120	0.757 ± 0.127
	1000	0.700 ± 0.133	0.677 ± 0.147	0.694 ± 0.140	0.676 ± 0.147
	5000	0.465 ± 0.181	0.439 ± 0.178	0.444 ± 0.191	0.441 ± 0.179
	10000	0.424 ± 0.178	0.392 ± 0.182	0.398 ± 0.183	0.392 ± 0.178
MAR	100	0.975 ± 0.086	0.969 ± 0.094	1.007 ± 0.101	1.007 ± 0.101
	500	0.814 ± 0.103	0.756 ± 0.108	0.755 ± 0.131	0.754 ± 0.129
	1000	0.748 ± 0.123	0.692 ± 0.160	0.681 ± 0.156	0.682 ± 0.154
	5000	0.609 ± 0.158	0.543 ± 0.174	0.430 ± 0.186	0.430 ± 0.189
	10000	0.690 ± 0.188	0.563 ± 0.161	0.404 ± 0.181	0.411 ± 0.184
MNAR	100	0.984 ± 0.068	0.963 ± 0.078	1.051 ± 0.129	0.970 ± 0.082
	500	0.836 ± 0.102	0.776 ± 0.114	0.980 ± 0.116	0.777 ± 0.115
	1000	0.810 ± 0.119	0.691 ± 0.140	0.904 ± 0.163	0.684 ± 0.149
	5000	0.721 ± 0.184	0.513 ± 0.201	0.676 ± 0.177	0.483 ± 0.185
	10000	0.774 ± 0.201	0.513 ± 0.185	0.588 ± 0.203	0.444 ± 0.184

Table 5 Mean and standard deviation of F_1 scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for dense networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	0.052 ± 0.043	0.059 ± 0.046	0.060 ± 0.045	0.060 ± 0.046
	500	0.148 ± 0.070	0.166 ± 0.077	0.160 ± 0.071	0.166 ± 0.076
	1000	0.217 ± 0.094	0.248 ± 0.089	0.242 ± 0.088	0.245 ± 0.092
	5000	0.457 ± 0.136	0.474 ± 0.124	0.461 ± 0.115	0.474 ± 0.124
	10000	0.525 ± 0.140	0.541 ± 0.151	0.534 ± 0.147	0.544 ± 0.148
MAR	100	0.042 ± 0.048	0.050 ± 0.052	0.058 ± 0.053	0.058 ± 0.051
	500	0.138 ± 0.071	0.170 ± 0.081	0.181 ± 0.081	0.180 ± 0.078
	1000	0.220 ± 0.109	0.250 ± 0.095	0.262 ± 0.086	0.265 ± 0.088
	5000	0.460 ± 0.124	0.461 ± 0.121	0.488 ± 0.129	0.488 ± 0.128
	10000	0.498 ± 0.118	0.508 ± 0.126	0.552 ± 0.132	0.552 ± 0.133
MNAR	100	0.036 ± 0.040	0.054 ± 0.053	0.051 ± 0.054	0.054 ± 0.052
	500	0.143 ± 0.066	0.172 ± 0.083	0.155 ± 0.064	0.177 ± 0.084
	1000	0.207 ± 0.087	0.239 ± 0.093	0.208 ± 0.077	0.254 ± 0.100
	5000	0.440 ± 0.123	0.446 ± 0.118	0.434 ± 0.126	0.455 ± 0.124
	10000	0.490 ± 0.125	0.508 ± 0.115	0.515 ± 0.127	0.526 ± 0.127

Table 6 Mean and standard deviation of normalised SHD scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for dense networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	0.999 ± 0.033	1.013 ± 0.036	1.029 ± 0.046	1.015 ± 0.037
	500	0.924 ± 0.047	0.911 ± 0.053	0.915 ± 0.050	0.911 ± 0.053
	1000	0.876 ± 0.068	0.853 ± 0.072	0.857 ± 0.071	0.855 ± 0.074
	5000	0.687 ± 0.137	0.674 ± 0.131	0.685 ± 0.118	0.675 ± 0.131
	10000	0.626 ± 0.156	0.605 ± 0.175	0.611 ± 0.169	0.603 ± 0.173
MAR	100	1.010 ± 0.032	1.013 ± 0.031	1.033 ± 0.050	1.031 ± 0.045
	500	0.928 ± 0.049	0.912 ± 0.056	0.913 ± 0.062	0.913 ± 0.062
	1000	0.863 ± 0.089	0.852 ± 0.076	0.848 ± 0.067	0.845 ± 0.070
	5000	0.680 ± 0.133	0.695 ± 0.133	0.665 ± 0.141	0.665 ± 0.140
	10000	0.661 ± 0.144	0.656 ± 0.154	0.601 ± 0.157	0.601 ± 0.158
MNAR	100	1.011 ± 0.035	1.003 ± 0.037	1.053 ± 0.064	1.011 ± 0.041
	500	0.928 ± 0.044	0.910 ± 0.052	0.948 ± 0.052	0.908 ± 0.053
	1000	0.879 ± 0.064	0.862 ± 0.073	0.898 ± 0.062	0.851 ± 0.079
	5000	0.704 ± 0.129	0.709 ± 0.119	0.720 ± 0.134	0.698 ± 0.127
	10000	0.668 ± 0.142	0.657 ± 0.135	0.643 ± 0.144	0.633 ± 0.151

Table 7 Mean and standard deviation of F_1 scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for real-world networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	0.112 ± 0.103	0.078 ± 0.048	0.072 ± 0.046	0.079 ± 0.050
	500	0.325 ± 0.268	0.319 ± 0.273	0.305 ± 0.279	0.321 ± 0.272
	1000	0.312 ± 0.265	0.365 ± 0.220	0.367 ± 0.216	0.365 ± 0.220
	5000	0.391 ± 0.248	0.444 ± 0.213	0.438 ± 0.219	0.444 ± 0.213
	10000	0.426 ± 0.246	0.458 ± 0.196	0.463 ± 0.192	0.458 ± 0.196
MAR	100	0.043 ± 0.037	0.048 ± 0.039	0.050 ± 0.041	0.047 ± 0.039
	500	0.173 ± 0.149	0.258 ± 0.338	0.152 ± 0.121	0.243 ± 0.342
	1000	0.264 ± 0.303	0.348 ± 0.352	0.362 ± 0.313	0.363 ± 0.311
	5000	0.298 ± 0.340	0.321 ± 0.256	0.394 ± 0.266	0.413 ± 0.261
	10000	0.237 ± 0.326	0.315 ± 0.247	0.469 ± 0.349	0.474 ± 0.343
MNAR	100	0.055 ± 0.047	0.106 ± 0.095	0.127 ± 0.153	0.140 ± 0.149
	500	0.234 ± 0.240	0.135 ± 0.079	0.244 ± 0.234	0.239 ± 0.236
	1000	0.249 ± 0.271	0.271 ± 0.256	0.220 ± 0.120	0.294 ± 0.243
	5000	0.276 ± 0.253	0.339 ± 0.236	0.224 ± 0.107	0.335 ± 0.241
	10000	0.270 ± 0.217	0.353 ± 0.231	0.237 ± 0.105	0.382 ± 0.242

Table 8 Mean and standard deviation of normalised SHD scores produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for real-world networks, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	1.129 ± 0.098	1.145 ± 0.137	1.180 ± 0.160	1.151 ± 0.136
	500	0.921 ± 0.287	0.928 ± 0.284	0.950 ± 0.303	0.927 ± 0.284
	1000	1.008 ± 0.434	0.883 ± 0.303	0.878 ± 0.298	0.883 ± 0.303
	5000	0.975 ± 0.437	0.869 ± 0.310	0.882 ± 0.321	0.869 ± 0.310
	10000	0.909 ± 0.368	0.856 ± 0.296	0.854 ± 0.291	0.856 ± 0.296
MAR	100	1.157 ± 0.137	1.161 ± 0.150	1.189 ± 0.129	1.189 ± 0.128
	500	1.096 ± 0.217	0.963 ± 0.438	1.125 ± 0.189	0.999 ± 0.451
	1000	1.070 ± 0.443	0.937 ± 0.472	0.950 ± 0.415	0.942 ± 0.411
	5000	1.126 ± 0.597	1.027 ± 0.453	0.970 ± 0.444	0.937 ± 0.436
	10000	1.224 ± 0.551	1.095 ± 0.466	0.868 ± 0.609	0.856 ± 0.599
MNAR	100	1.190 ± 0.115	1.160 ± 0.155	1.122 ± 0.229	1.108 ± 0.214
	500	1.065 ± 0.313	1.169 ± 0.154	1.111 ± 0.288	1.097 ± 0.277
	1000	1.065 ± 0.385	1.064 ± 0.356	1.125 ± 0.179	1.045 ± 0.344
	5000	1.139 ± 0.424	1.035 ± 0.389	1.182 ± 0.206	1.061 ± 0.379
	10000	1.158 ± 0.359	1.073 ± 0.394	1.222 ± 0.218	1.012 ± 0.413

Appendix D Supplementary results of execution time

See results in Table 9.

Table 9 Mean and standard deviation of execution times produced by Structural EM, HC-pairwise, HC-IPW and HC-aIPW for sparse networks and relative to HC when applied to complete data, under the different assumptions of missingness and sample sizes

Data	Sample size	Structural EM	HC-pairwise	HC-IPW	HC-aIPW
MCAR	100	161.29 ± 54.37	1.55 ± 0.68	8.02 ± 2.27	8.30 ± 1.95
	500	423.61 ± 146.51	1.55 ± 0.43	6.28 ± 1.50	6.89 ± 1.47
	1000	556.72 ± 178.34	1.54 ± 0.36	5.88 ± 1.68	6.44 ± 1.78
	5000	739.52 ± 269.94	1.73 ± 0.70	8.76 ± 2.48	9.24 ± 2.38
	10000	642.91 ± 254.91	1.63 ± 0.42	11.46 ± 3.45	11.90 ± 3.57
MAR	100	164.96 ± 46.26	1.64 ± 0.51	8.33 ± 2.15	9.14 ± 2.29
	500	421.37 ± 144.99	1.62 ± 0.42	7.50 ± 2.00	8.42 ± 2.09
	1000	554.72 ± 186.89	1.55 ± 0.34	7.73 ± 2.29	8.62 ± 2.47
	5000	740.71 ± 275.59	1.59 ± 0.32	14.81 ± 5.49	15.33 ± 5.24
	10000	657.52 ± 313.70	1.73 ± 0.53	22.49 ± 7.52	22.75 ± 7.20
MNAR	100	154.04 ± 44.86	1.54 ± 0.56	8.80 ± 3.27	8.79 ± 2.67
	500	419.91 ± 145.50	1.62 ± 0.44	8.18 ± 3.58	7.90 ± 2.47
	1000	552.16 ± 181.62	1.58 ± 0.41	7.69 ± 3.05	7.56 ± 2.06
	5000	776.32 ± 359.57	1.87 ± 0.87	13.29 ± 5.93	12.89 ± 7.74
	10000	615.91 ± 271.88	1.86 ± 0.77	20.07 ± 9.38	17.93 ± 8.38

Acknowledgements This research was supported by the EPSRC Fellowship project EP/S001646/1 on *Bayesian Artificial Intelligence for Decision Making under Uncertainty*.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Yang Liu. The first draft of the manuscript was written by Yang Liu and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data used for the simulation results are available upon request to the corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1), 40–49.
- Balov, N., et al. (2013). Consistent model selection of discrete Bayesian networks from incomplete data. *Electronic Journal of Statistics*, 7, 1047–1077.
- Bodewes, T., & Scutari, M. (2021). Learning Bayesian networks from incomplete data with the node-average likelihood. *International Journal of Approximate Reasoning*, 138, 145–160.
- Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov), 507–554.
- Constantinou, A. C., Fenton, N., Marsh, W., & Radlinski, L. (2016). From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. *Artificial Intelligence in Medicine*, 67, 75–93.

- Constantinou, A. C., Liu, Y., Chobtham, K., Guo, Z., & Kitson, N. K. (2021). Large-scale empirical validation of Bayesian Network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, 131, 151–188.
- Cussens, J. (2011). Bayesian network learning with cutting planes. In *Proceedings of the 27th conference on uncertainty in artificial intelligence (UAI 2011)*, AUA Press, pp. 153–160.
- Friedman, N., et al. (1997). Learning belief networks in the presence of missing values and hidden variables. In *ICML, Citeseer*, Vol. 97, pp. 125–133.
- Gain, A., & Shpitser, I. (2018). Structure learning under missing data. In *International conference on probabilistic graphical models*, PMLR, pp. 121–132.
- Gámez, J. A., Mateo, J. L., & Puerta, J. M. (2011). Learning bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1), 106–148.
- Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual Review of Psychology*, 60, 549–576.
- Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3), 197–243.
- Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260), 663–685.
- John, C., Ekpenyong, E. J., & Nworu, C. C. (2019). Imputation of missing values in economic and financial time series data using five principal component analysis approaches. *CBN Journal of Applied Statistics*, 10(1), 51–73.
- Mohan, K., & Pearl, J. (2021). Graphical models for processing missing data. *Journal of the American Statistical Association* pp 1–16.
- Mohan, K., Pearl, J., & Tian, J. (2013). 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*, Curran Associates, Inc., vol 26, <https://proceedings.neurips.cc/paper/2013/file/0ff8033cf9437c213ee13937b1c4c455-Paper.pdf>.
- Pedersen, A. B., Mikkelsen, E. M., Cronin-Fenton, D., Kristensen, N. R., Pham, T. M., Pedersen, L., & Petersen, I. (2017). Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, 9, 157.
- Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3), 581–592.
- Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys* (Vol. 81). New York: Wiley.
- Ruggieri, A., Stranieri, F., Stella, F., & Scutari, M. (2020). Hard and soft EM in Bayesian network learning from incomplete data. *Algorithms*, 13(12), 329.
- Schwarz, G., et al. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2), 461–464.
- Scutari, M. (2010). Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3).
- Silander, T., Leppä-Aho, J., Jääsaari, E., & Roos, T. (2018). Quotient normalized maximum likelihood criterion for learning Bayesian network structures. In *International conference on artificial intelligence and statistics*, PMLR, pp. 948–957.
- Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). *Causation, prediction, and search*. Cambridge: MIT press.
- Strobl, E. V., Visweswaran, S., & Spirtes, P. L. (2018). Fast causal inference with non-random missingness by test-wise deletion. *International Journal of Data Science and Analytics*, 6(1), 47–62.
- Tian, Y., Zhang, K., Li, J., Lin, X., & Yang, B. (2018). LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318, 297–305.
- Tsamardinos, I., Aliferis, C. F., Statnikov, A. R., & Statnikov, E. (2003). Algorithms for large scale Markov blanket discovery. *FLAIRS conference*, 2, 376–380.
- Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1), 31–78.
- Tu, R., Zhang, C., Ackermann, P., Mohan, K., Kjellström, H., & Zhang, K. (2019). Causal discovery in the presence of missing data. In *The 22nd international conference on artificial intelligence and statistics*, PMLR, pp. 1762–1770.
- Zemichael, T., & Dietterich, T.G. (2019). Anomaly detection in the presence of missing values for weather data quality control. In *Proceedings of the 2nd ACM SIGCAS conference on computing and sustainable societies*, pp. 65–73.