Discover Data

Research

Evaluation of Bayesian network scoring functions in polychotomous data analysis

Xuejia Ke[1,2] · **Katherine Keenan**[2] · **V. Anne Smith**[1]

Received: 19 December 2024 / Accepted: 11 April 2025
Published online: 19 May 2025

Abstract
Bayesian networks (BNs) are probabilistic graphical models used to represent dependencies and independencies among variables. They have been applied widely to many areas in, e.g., biology and medicine, to untangle complex interrelationships, and are now finding wider use in areas such as social science with differing data features, for example highly polychotomous (multi-category) data. To construct BNs, scoring functions guide selection of the most appropriate model. Among these, the BDe scoring function requires specifying hyperparameters that influence the priors on the network parameters. This study evaluates the performance of four scoring functions—AIC, BIC, BDe, and log-likelihood—particularly with highly polychotomous data. We assessed the overall performance of the scoring function, and for BDe, we varied its hyperparameter to evaluate its impact. Performance of the scoring functions was significantly influenced by the number of nodes, network complexity, and sample size. BIC and BDe (with default hyperparameters) generally offered higher precision, especially with larger sample sizes, while log-likelihood tended to overfit, showing high recall but low precision. AIC and BDe required careful tuning based on discrete levels and sample sizes. Optimizing the hyperparameters in BDe was crucial for balancing model complexity and fit. We propose a simulation method for identifying the optimum hyperparameters for using BDe scoring function in real-world data applications. The study provides insights to enhance BN models' robustness and accuracy, emphasizing the importance of considering sample size and the number of discrete levels when selecting and tuning scoring functions for BN structure learning.

Keywords Bayesian networks · Structure learning · Scoring functions · Polychotomous data

1 Introduction

Bayesian networks (BNs) are a class of probabilistic graphical model that are composed of a directed acyclic graph (DAG) that encodes the independence relationships between variables, with parameters describing their distributions. They were firstly proposed by Pearl [1]. They have been used across a wide variety of fields, from gene expression to social science to unravel relationships in complex, interconnected systems [2–5]. Newer applications in areas such as social science face challenges in dealing with high numbers of discrete levels, for example, many geographical sampling locations [6] or high number of ethnicity categories [7, 8]. Here we explore in particular BN structure learning in highly polychotomous (multi-category data).

In BN structure learning, there are three main approaches: constraint-based structure learning, score-based structure learning and Bayesian model averaging, with the score-based and the constraint-based approaches being widely recognized

✉ V. Anne Smith, vas1@st-andrews.ac.uk; Xuejia Ke, xk5@st-andrews.ac.uk; Katherine Keenan, katherine.keenan@st-andrews.ac.uk | [1]School of Biology, University of St Andrews, St Andrews KY16 9TH, UK. [2]School of Geography and Sustainable Development, University of St Andrews, St Andrews KY16 9AL, UK.

[9]. Score-based learning approaches are more often applied in real-data scenarios than others. Unlike constraint-based learning, which relies on testing dependencies and can be biased by a single test failure, and unlike Bayesian model averaging, which is impractical for averaging predictions of all possible structures, score-based approaches are more accurate and versatile [9]. They utilize a scoring function to assess how well a structure fits the observed data, identifying the best network structure as the one with the highest score [10]. Given that the number of possible structures is superexponential and the problem is usually NP-hard, heuristic search techniques are necessary to find the highest-scoring network [11]. This requires high computational resources, especially in large-scale networks, making the choice of scoring function important [9, 11]. In this study, we focus on the scoring functions used in the score-based learning approach.

Several popular scoring functions are used in score-based learning in BNs, including the Bayesian Dirichlet equivalence score (BDe) [12], Akaike's information criterion (AIC) [13], Bayesian information criterion (BIC) [14], K2 [10], and mutual information tests (MIT) [15], among others [16, 17]. BIC is conceptually similar to the minimum description length (MDL) [18]. These scoring functions can be categorized based on their applicability to either discrete or continuous data, with some being versatile enough to accommodate both. Among the most popular, BDe works exclusively for discrete data, while AIC and BIC are applicable to both continuous and discrete data. The multinomial log-likelihood is also used for discrete data [19]. In this study, we focus on four scoring functions: AIC, BIC, BDe and log-likelihood. The first three are widely used and suitable for discrete data, while log-likelihood is included as a basic reference point.

Many scoring functions take the form of penalized log-likelihood functions, which penalize the complexity of network structures to avoid overfitting and enhance model generalization [20]. Such scoring functions are referred to as likelihood scores [9]. Among the four scoring functions we focus on, log-likelihood, AIC, and BIC are all likelihood scores.

Without the penalty term, the log-likelihood is the log probability of the data given the structure $\log P(D \mid G)$, in which D represents data and G represents the structure. Intuitively, one aims to maximize the model's likelihood to make the observed data as likely as possible [9]. A higher log-likelihood indicates a better fit. In the detailed decomposed form of log-likelihood, as shown in Eq. 1 and defined in Table 1, $\ln P(X_i = k \mid Pa(X_i) = j)$ can be interpreted as part of the strength of the dependence between variable X_i and its parents $Pa(X_i)$. Thus, in log-likelihood, the preference is for network structures where the parents of each variables provide substantial information for predicting it. In other words, it favors more complex networks over simpler ones. Consequently, a fully connected network is often the maximum likelihood network, which can lead to overfitting [9].

$$\ln(\hat{L}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \ln P(X_i = k \mid Pa(X_i) = j) \tag{1}$$

The AIC scoring function is derived from information theory and is based on the asymptotic properties of maximum likelihood estimators, assuming a large sample size. This means its effectiveness improves as the sample size grows larger [13, 21]. Unlike the log-likelihood, AIC aims to minimize the score for better fits, as indicated by Eq. 2 and the definitions in Table 1. Its penalty term, $2k$, penalizes the complexity of the network by accounting for the number of parameters, thus helping to prevent overfitting.

Table 1 List of symbol definitions for the four scoring functions: AIC, BIC, BDe, and log-likelihood

Symbol Notions
• $\ln(\hat{L})$: log-likelihood of the model given the data
• $P(X_i = k \mid Pa(X_i) = j)$: Conditional probability of $X_i = k$ given the parent configuration $Pa(X_i) = j$
• k: Number of independent parameters in the model
• N: Number of data points (samples)
• n: Number of nodes in the Bayesian network
• q_i: Number of different configurations of the parent set $Pa(X_i)$ for node X_i
• r_i: Number of states of node X_i
• N_{ijk}: Count of instances where $X_i = k$ and $Pa(X_i) = j$
• N_{ij}: Count of instances where the parents of X_i are in configuration j
• α_{ij}: Hyperparameter for the Dirichlet prior associated with the parent configuration j of X_i
• α_{ijk}: Hyperparameter for the Dirichlet prior associated with the state k of X_i given the parent configuration j
• $\Gamma(\cdot)$: Gamma function, an extension of the factorial function

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{2}$$

The BIC scoring function has a higher penalty term than AIC. The penalty term for BIC is $k\ln(N)$, as shown in Eq. 3 and defined in Table 1, where k is the number of parameters and N is the number of data points. This penalty increases with both the number of parameters and the logarithm of the sample size. In contrast, the penalty term for AIC is $2k$, which increases only with the number of parameters and is independent of the sample size. Because $\ln(N)$ is typically greater than 2 for most sample sizes, BIC imposes a higher penalty for model complexity than AIC does. This makes BIC more conservative, favoring simpler models, especially as the sample size grows larger [20].

$$\text{BIC} = k\ln(N) - 2\ln(\hat{L}) \tag{3}$$

The BDe scoring function does not have a penalty term in the same way that AIC or BIC do, making it more complex, as detailed in Eq. 4 and Table 1. As a Bayesian score, BDe originated from the Bayesian Dirichlet (BD) scoring function, which incorporates prior knowledge into the scoring process through a Dirichlet prior. This prior influences the score based on both the data and the specified priors [10]. The count of instances N_{ij}, where the parents of each variable X_i are in each configuration $Pa(X_i) = j$, is the summation of instances where the variable X_i is at each state k and its parents are in the same configuration, as shown in Eq. 5. The Dirichlet prior is a prior distribution of the data before any actual data is observed, characterized by hyperparameters α_{ij} and α_{ijk}. Similarly, α_{ij} is the summation of α_{ijk} for every state k of variable X_i, as shown in Eq. 6 and defined in Table 1.

$$\text{BDe} = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\log\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} + \sum_{k=1}^{r_i}\log\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{4}$$

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{5}$$

$$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \tag{6}$$

To overcome the limitations of the original BD scoring function, researchers introduced the BDe scoring function, which incorporates a new hyperparameter. This enhancement addresses the need for specifying hyperparameters for all possible combinations of variables and their corresponding parents, as well as resolving inconsistencies in scores for the structures within the same equivalence class in BD. The improved calculation of α_{ij} is detailed in Eq. 7 and Table 1. This α is the sum of the Dirichlet hyperparameters for each parent configuration and each variable state, resulting in equivalent scores for networks within the same equivalence class. Furthermore, under the assumption that all network structures are equally likely, there is a uniform prior distribution over all network structures, as shown in Eq. 8. Thus, the new hyperparameter α is the only input value for BDe and is referred to as the *imaginary sample size (iss)*, *equivalent sample size*, or *effective sample size*, as shown in Eq. 9 and Table 1. This developed version of BDe is also referred to as BDeu [12]. In this study, although we refer to BDe throughout, we are specifically using the BDeu variant.

$$\alpha = \sum_{j=1}^{q_i}\sum_{k=1}^{r_i} \alpha_{ijk} \tag{7}$$

$$\alpha_{ijk} = \frac{\alpha}{q_i \cdot r_i} \tag{8}$$

$$iss = \alpha_{ijk} \cdot q_i r_i \tag{9}$$

In the study of BN structure learning, a crucial distinction is made between discrete and continuous variables. Continuous variables, common in fields such as biology, medical science, economics and other fields are often discretized to simplify analysis. Social science fields often deal exclusively with categorical data. This discretization is significant because discrete scores, which are combinatorial, fundamentally differ from continuous scores, which

are additive [9]. While continuous scores aggregate influences, potentially overlooking extreme values, discrete scores allow for the amalgamation of information from multiple parental levels. This makes discrete scoring methods particularly useful when dealing with multi-category data [9]. However, discretizing continuous variables comes with a trade-off: the potential loss of valuable information [22]. This loss can impact the accuracy and effectiveness of the BN structure learning process. For example, the continuous variable age is a common demographic feature in social science. One often needs to discretize it into certain age groups, such as 18–34 for early career, 35–54 for mid-career, and above 55 for late career in workforce analysis. However, more granular information that characterizes the underlying patterns of the data is lost during this process. Although increasing the number of discretized groups can increase model complexity and lead to problems such as high computational burden, it is necessary to make use the most of the survey data for analysis. Recognizing this challenge, this study focuses on the analysis of polychotomous (multi-category) data. By simulating various scenarios, we aim to understand the performance of four scoring functions—AIC, log-likelihood, BIC, and BDe—in capturing the true network structure.

Some studies have systematically evaluated the performance of common scoring functions [20, 23–25], as summarized in Table 2. However, there is a lack of assessment regarding their effectiveness when applied to polychotomous data. In an empirical evaluation, researchers tested the uniform prior score metric (UPSM), conditional uniform prior score metric (CUPSM), Dirichlet prior score metric (DPSM), BDe and MDL/BIC scoring functions using the three-node, five-node, and the ALARM network with a maximum of four discrete levels. They used data points ranging from 1000 to 10,000 and concluded that DPSM performs better than MDL/BIC, followed by CUPSM, UPSM, and BDe [23]. Another study, involving simulations of 13 gold standard networks, each with fewer than 25 variables and an average of fewer than 3 parents per node, examined 18 different data sizes ranging from 200 to 10,000. The study found that BIC performs better than AIC and BDe when the data is sufficient. AIC performs similarly to BDe but its performance is erratic and unpredictable [20]. Additionally, a simulation study with 10 variables and data points ranging from 200 to 1200, using binary variables, concluded that AIC is better than MDL/BIC at high sample sizes [24]. Furthermore, in a simulation with 20 variables, each having 3 discrete levels, and 2000 data points, BDe performed better than BIC. When the data size was reduced to 100 data points, BDe recovered some information while BIC found no links at all, suggesting that BIC over-penalizes complexity relative to the BDe with small quantities of data [3]. One study of particular interest explored a wide variety of networks ranging from small binary networks to a large polychotomous network (417 nodes, up to 21 discrete levels per node) [25]. They found that the BIC score generally outpeformed AIC, BDe and K2, except for small networks with high data amounts where the BDe score performed best. While not highlighted by the authors, the results show that only for their most polychotomous network, the AIC outperformed all other scores [25]. These studies indicate that the number of nodes, network complexity, and sample sizes all impact the performance of scoring functions in BN structure learning. However, the role of the number of discrete levels in each categorical variable remains under-explored.

The role of the hyperparameter *iss* α in the BDe score remains unclear in BN structure learning, particularly when dealing with polychotomous data. It has been observed that when α approaches zero, it leads to an empty network structure, whereas large α values tend to result in complete network structures [26]. Further empirical simulations confirmed that the performance of BDe is highly sensitive to the hyperparameter α [20, 27, 28]. Researchers have theoretically demonstrated that the ratio of α to sample size is crucial in determining the penalty for adding arcs in BN structure learning. Specifically, as α increases, the number of arcs in the network tends to increase monotonically, while a decrease in α leads to a reduction in the number of arcs [29]. This suggests that α significantly dictates the quantity of data points required to effectively influence the posterior distribution of parameters. Additionally, it has been identified that the skewness (non-uniformity) of the sample distribution is associated with the optimal α values, the more skewed the sample distribution, the smaller the optimal α values [30, 31]. It is recommended to use an α value of 1 as the smallest positive assignment to minimize the influence of α on parameter estimation, thereby better reflecting the data [31, 32]. The α value of 1 is now commonly used as default in many applications [19]. However, these studies primarily focus on the relationship between sample size and optimal α values and their impact on the performance of BDe in BN structure learning. The relationship between optimal *iss* α values and the number of discrete levels in each categorical variable remains under-explored, which is crucial for guiding the application of the BDe scoring function in real data analysis.

In this study, we focus on the following key questions:

1. How do the AIC, BIC, and BDe scoring functions perform in general cases of BN structure learning, particularly when considering highly polychotomous data?
2. Given the research gap regarding the impact of the hyperparameter α (*iss*) on the BDe scoring function:

Table 2 Summary of studies evaluating scoring functions in BN structure learning

Study	Network size	Data sizes	Scoring functions compared	Key findings
Yang et al. [23]	3-node,5-node, ALARM network (up to 4 discrete levels)	1000 to 10,000	UPSM, CUPSM, DPSM, BDe, MDL/BIC	DPSM performs best, followed by CUPSM, UPSM, and BDe
Liu et al. [20]	13 gold standard networks (< 25 variables, < 3 parents/node)	200 to 10,000	BIC, AIC, BDe	BIC performs best with sufficient data; AIC is erratic but similar to BDe
Van Allen et al. [24]	10 variables (binary)	200 to 1200	AIC, MDL/BIC	AIC outperforms MDL/BIC at high sample sizes
Yu et al. [3]	20 variables (3 discrete levels)	100, 2000	BDe, BIC	BDe performs better overall; BIC over-penalizes with small data sizes
Chua and Ong [25]	7 gold standard networks (8–417 variables, most 2–5 discrete levels, one up to 11 and one up to 21 levels)	5000 to 20,000	AIC, BIC, BDe, K2	BDe best with large samples on small networks; BIC performs well most consistently; AIC best on most polychotomous network

a. How does the hyperparameter α (*iss*) affect BDe's performance in BN structure learning with polychotomous data?
b. What is the optimal α (*iss*) value for BDe in BN structure learning with polychotomous data?

Through this investigation, we aim to provide insights that can lead to more robust and accurate Bayesian network models, ultimately benefiting various fields where complex, polychotomous data is prevalent.

2 Methods

These questions were addressed using a simulation approach. The simulation work conducted is categorized into two primary parts: *Global Simulation* and *Local Simulation*. In *Global Simulation*, we evaluated the performance of four scoring functions—log-likelihood, BDe, AIC and BIC—by focusing on the entire network structure. In *Local Simulation*, we assessed their performance by concentrating on a single node, examining the conditions under which false negative arcs or false positive arcs appear in the specified network. The local simulation was based on the fact that these four scoring functions are all decomposable. This means that the overall score of a network can be computed as the sum of local scores for each node and its parents [9]. This property enables us to empirically explore their performance by focusing on a single node, as each node and its parent set can be evaluated independently.

2.1 Global simulation

Figure 1 shows an overview of the global simulation approach. The main steps are as follows:

1. Generate a random graph. This random graph is also referred to as the original structure in the final step for comparison.
2. Sample data points from the random graph to get the simulated data.
3. Learn the Bayesian network structure, either: (a) using scoring function log-likelihood, (b) using BDe with the default *iss* value 1, (c) using AIC, or (d) using BIC.
4. Compare learned Bayesian network structures with the original structure.

In addition to learning the Bayesian network structures using four different scoring functions, we examined the impact of the hyperparameter *iss* values on the performance of the BDe scoring function by varying the *iss* values in *Step 3* of the aforementioned method.

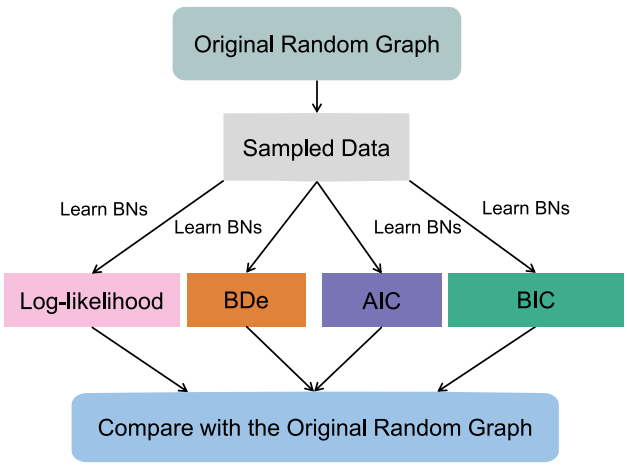

Fig. 1 Flowchart of the global simulation approach

2.1.1 Random network generation and data sampling

We generated random networks and sampled data to evaluate the performance of four scoring functions. First, we generated a randomly connected network structure with the specified number of nodes/variables (ranging from 2 to 8) using Ide's and Cozman's Generating Multi-connected DAGs (ic-dag) algorithm in the function *random.graph* from the R package *bnlearn* [19]. We set the maximum in-degree for any node to 3, with each node having an equal number of specified discrete levels, ranging from 2 to 20. Conditional probability tables (CPTs) for each node were obtained by generating random vectors from the Dirichlet distribution using function *rdirichlet* from the R package *MCMpack* [33]. The hyperparameter α of the Dirichlet distribution was set to 0.5 for nodes with parents and 5 for nodes without parents, providing the random parameterised BN. We then randomly sampled 1000 or 5000 data points from the parameterised BN to get the sampled data using the function *rbn* from the R package *bnlearn* [19]. Each variable number/discrete level/data size was repeated 100 times.

We used the same simulation method to assess the impact of the the hyperparameter α (*iss*) values on the performance of the BDe scoring function. Instead of setting α to its default value of 1, we repeated the aforementioned method using α values of 100, 650, 1200, 4235, and 10,000 for the 5000 data points, nodes/variables (ranging from 2 to 8), with each variable having 10 discrete levels. Each combination was repeated 10 times.

To further explore the impact of *iss*, we focused on two specific models: a 5-node network with each node having a maximum of 2 parents, and a 10-node network with each node having a maximum of 4 parents. We sampled 500, 1000, 5000, or 10,000 data points using logic sampling with the function *rbn* from the R package *bnlearn* [19, 34].

2.1.2 Bayesian network structure learning

Throughout the study, we used a score-and-search algorithm to learn BN structures from the sampled data. As shown in Fig. 1, we used the scoring functions log-likelihood, BDe (with the default *iss* value of 1), AIC and BIC alongside the tabu search algorithm to identify the best network structure and evaluate the performance of these four scoring functions [35]. In the exploration of the hyperparameter *iss* values in the BDe function, we varied the *iss* values from 0 to 1 in intervals of 0.1, and from 2 to 10,000 in intervals of 1, while using the tabu search algorithm in BN structure learning. For all scenarios, we applied the function *tabu* from the R package *bnlearn* [19].

2.1.3 Evaluation of learned network structures

To compare the learned BN structures with the original ones, we compared their skeletons using functions *compare* and *skeleton* from the R package *bnlearn* [19]. We compared skeletons rather than equivalence classes to avoid artefacts generated by single links generating large changes in directionality or not in an equivalence class [36]. Performance was measured via precision and recall (sensitivity). Precision measures ability to recover structure without mistakes (e.g., avoiding false positive arcs), while recall measures how much of a structure is recovered (e.g., avoiding false negative arcs). They are calculated as:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{10}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{11}$$

where True Positive is the learned BN arc present in original structure, False Positive is the learned BN arc not present in original structure, and False Negative is the arc in original structure not present in learned BN.

2.1.4 Evaluation of *iss* values in BDe

We evaluated the performance of the hyperparameter α (*iss*) in BDe in two ways. The first approach was to use the aforementioned evaluation method for α values of 1, 100, 650, 1200, 4235, and 10,000 applied to the 5000 data points

with various number of nodes/variables (ranging from 2 to 8). The second approach was to select the α value that gave the specified model, either the 5-node network or the 10-node network, the highest score within the assessed range of α values from 0.1 to 10,000. These α values were considered the optimum values.

2.2 Local simulation

Figure 2 shows an overview of the local simulation approach. The main steps are as follows:

1. Generate paired graphs: one set as the true graph and the other as the test graph. The test graph differs from the true graph by the presence or absence of a single parent for the focal node, ensuring a controlled variation for comparison.
2. Sample data points from the true graph to create the simulated data.
3. Score the focal node in both the test graph and the true graph using the simulated data with one of the following scoring functions: (a) log-likelihood, (b) BDe with the default *iss* value of 1, (c) AIC, or (d) BIC.
4. Compute the score difference between the test graph and the true graph, using the data sampled from the true graph.

Additionally, we evaluated the impact of the hyperparameter *iss* values on the performance of the BDe scoring function by varying the *iss* values in *Step 3* of the aforementioned method.

2.2.1 Generation of paired graphs and data sampling

Instead of focusing on the entire network structure, we designed paired graphs to investigate the conditions under which a false positive arc or a false negative arc appear. We focused on a single node in the paired graphs, where the only difference between the graphs is the presence or absence of a parent for that focal node. In other words, the only difference between the paired graphs is a single additional or missing arc. This additional or missing arc is treated as the false positive or false negative arc in different scenarios. For example, in Fig. 3a, we have paired two-node structures where the focal node is A. If we set the graph with the arc from node B to A as the true graph and the graph without the arc as the test graph, and if the scoring function favors the test graph, this represents the appearance of a false negative arc. Conversely, if we set the graph without the arc from node B to A as the true graph and the graph without the arc as the test graph, and if the scoring function favors the test graph, this represents the appearance of a false positive arc. We alternated these scenarios to simulate the appearance of false positive and false negative arcs.

To avoid potential bias from using the Dirichlet distribution in the generation of the data (as was above), we generated parameters for a network structure from randomised data. These parameters were then used to generate simulated data from the network structure. We first generated an empty data frame with a specified number of variables. For each variable, values were sampled with replacement from a list of letters, where the number of letters matched the number

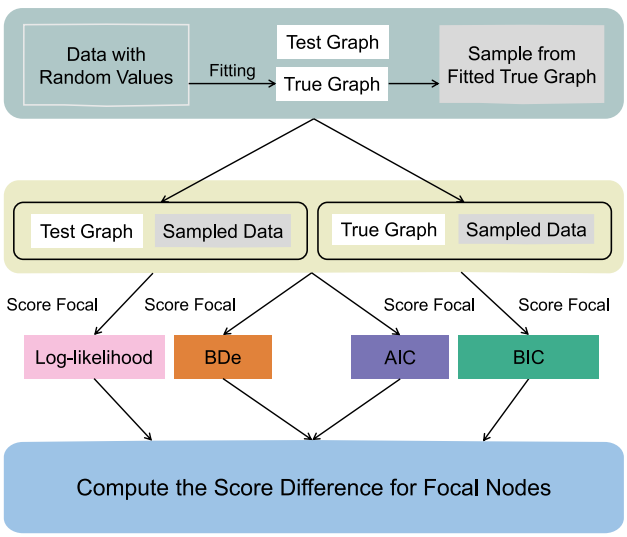

Fig. 2 Flowchart of the local simulation approach

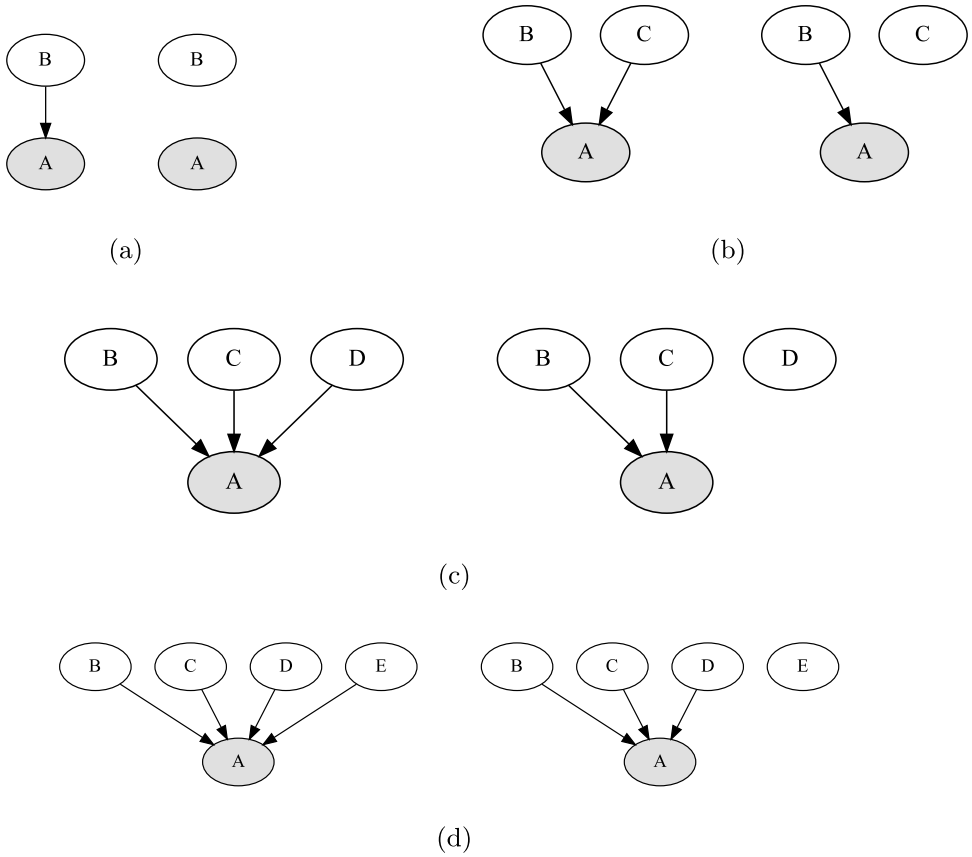

Fig. 3 Paired test and true networks in the local simulation evaluating the performance of the imaginary sample size (iss) values in the BDe scoring function. **a** Two-node network: the focal node, A, has either one parent (left) or no parent (right). If the test network is the left one (with an arc from B to A) and the true network is the right one (without the arc), we examine the appearance of false positive arcs when the score difference of node A (test minus true) is positive. Conversely, if the test network is the right one (without the arc) and the true network is the left one (with the arc), we examine the appearance of false negative arcs when the score difference is positive. **b** Three-node network, **c** Four-node network, and **d** Five-node network: these scenarios are similarly extended to networks with three, four, and five nodes, respectively. In all scenarios, the focal node is A, which is shaded in grey

of discrete levels, and the specified number of data points. This data frame was then fitted to the true graph using the "bayes" method in the *bn.fit* function from the R package *bnlearn* [19]. Afterwards, we sampled data from the obtained parameters using logic sampling [34] with the *rbn* function from the same package. For example, in the case of Fig. 3a, we generated a data frame with two binary variables and a sample size of 5000. We fitted the data frame to the true graph with the arc from node B to A. We then sampled the same number of data points from the fitted parameters to create the dataset for later scoring. Subsequently, we repeated this process by fitting the same data frame to the alternative true graph, which lacks the arc from node B to A, and sampled the same number of data points from the new fitted parameters to create a new dataset.

We compared the four scoring functions using the three-node network shown in Fig. 3b to evaluate their performance when the *iss* value of the BDe scoring function was set to its default value 1. The number of discrete levels ranged from 2 to 20, and the sample sizes were 50, 100, 500, 1000, and 5000. Each combination of discrete level and sample size was repeated 100 times. Additionally, we simulated all scenarios to evaluate the performance of the hyperparameter *iss* in the BDe scoring function. For this evaluation, the number of discrete levels ranged from 2 to 20, and the sample size was 5000. The *iss* values ranged from 0 to 1 in intervals of 0.1, and from 2 to 10,000 in intervals of 1. Each combination of discrete level and *iss* value was repeated 10 times.

We observed different responses of the BDe scoring function's hyperparameter α (*iss*) when dealing with nodes having one or two parents, specifically when α was set to 1. To explore this further, we introduced variation in the number of parents for the focal node. As shown in Fig. 3, we designed four paired graphs to assess the impact of the hyperparameter α (*iss*) on the performance of the BDe scoring function. We focused on the same node *A* in all structures. Figure 3a displays paired two-node network structures, where *A* has either 0 or 1 parent. Figure 3b shows paired three-node network structures, where *A* has 1 or 2 parents. Figure 3c presents paired four-node network structures, where *A* has 2 or 3 parents. Figure 3d illustrates paired five-node network structures, where *A* has 3 or 4 parents.

2.2.2 Comparison of scores

We scored the focal node in both the test graph and the true graph, along with the sampled data, using the *score* function from the R package *bnlearn* [19]. We computed the score difference to indicate the presence of a false positive or false negative arc. By consistently subtracting the score of the focal node in the true graph from the score in the test graph, a positive score difference indicates the appearance of a false positive or false negative arc. This means that the test graph, which contains the incorrect network structure, has a higher score than the true graph, suggesting that a search algorithm might erroneously favor the incorrect structure. Conversely, if the score of the true graph is higher, it indicates no such appearance and suggests the correct network structure is more likely to be identified.

In exploring the performance of four scoring functions, we counted the frequency of positive scores among the 100 repeats for each condition. In evaluating the performance of the hyperparameter *iss* in the BDe scoring function, we averaged the score differences across the 10 repeats and then proceeded with the analysis by extracting the α (*iss*) values where the averaged score difference was positive.

3 Results

3.1 Global simulation

3.1.1 Evaluation of scoring functions in overall network structure

Our simulation results illustrate the precision of learning BN structures using different scoring functions (AIC, BIC, BDe, and log-likelihood) across varying numbers of discrete levels (2 to 20) and two different sample sizes (1000 and 5000 data points). As shown in Fig. 4, precision generally increases with the number of discrete levels for both sample sizes and quickly reaches a plateau at 1. For networks with fewer variables, this plateau is reached more quickly. This trend is consistently observed in AIC and BIC in both sample sizes, and in BDe for the 5000 data point sample size. These findings suggest that more discrete levels provide richer information, aiding in accurately identifying the true network structure and reducing false positives. Interestingly, the plateau in AIC is reached at a slightly higher number of discrete levels for the 5000 data points compared to the 1000 data points. It indicates that larger sample sizes may require more discrete levels to fully capture the underlying structure, potentially due to the variability and sensitivity of AIC to data size.

There are also differences among the four scoring functions. BIC and BDe tend to perform better overall in precision and often overlap, especially at lower number of discrete levels. AIC follows a similar trend but with slightly lower precision compared to BIC and BDe. The log-likelihood generally shows the lowest precision than the other scoring functions, remaining relatively constant across both sample sizes and various number of discrete levels, and it decreases when the number of nodes in the network increases. For certain network scenarios, particularly with 1000 data points and a higher number of nodes, the precision for BDe drops sharply after a certain number of discrete levels (e.g., it drops at level 20 in 5-node networks). Notably, in the 8-node networks, precision decreases at level 15, increases again at level 17, and then drops sharply at level 18. This behavior could be due to overfitting, resulting from the incomplete CPTs because of the insufficient observations for each discrete level [37]. The BIC score generally maintains higher precision, which suggests it balances the fit and complexity of the model effectively across different discrete levels and scenarios. In most cases, especially for AIC and BIC, precision plateaus after reaching a certain number of discrete levels, indicating that additional discrete levels do not significantly improve or decrease the model's ability to identify true dependencies beyond a certain point.

Figure 5 compares the recall of learning Bayesian network structures using different scoring functions (AIC, BIC, BDe, and log-likelihood) across varying numbers of discrete levels (2 to 20) and two different sample sizes (1000 and 5000 data points). The log-likelihood score shows a higher recall (at 1.0) than the other scoring functions and remains relatively constant across both sample sizes and various number of discrete levels. This suggests that there is overfitting. It learns complete networks and captures no false negatives. For the other three scoring functions, recall generally decreases as the number of discrete levels increases for both sample sizes. This trend suggests that more discrete levels might introduce additional complexity or noise, making it harder to capture true dependencies. When there are 1000 data points, recall starts higher but decreases as the number of discrete levels increases, with more noticeable variability, especially for the AIC and BIC scoring functions. With 5000 data points, recall is initially

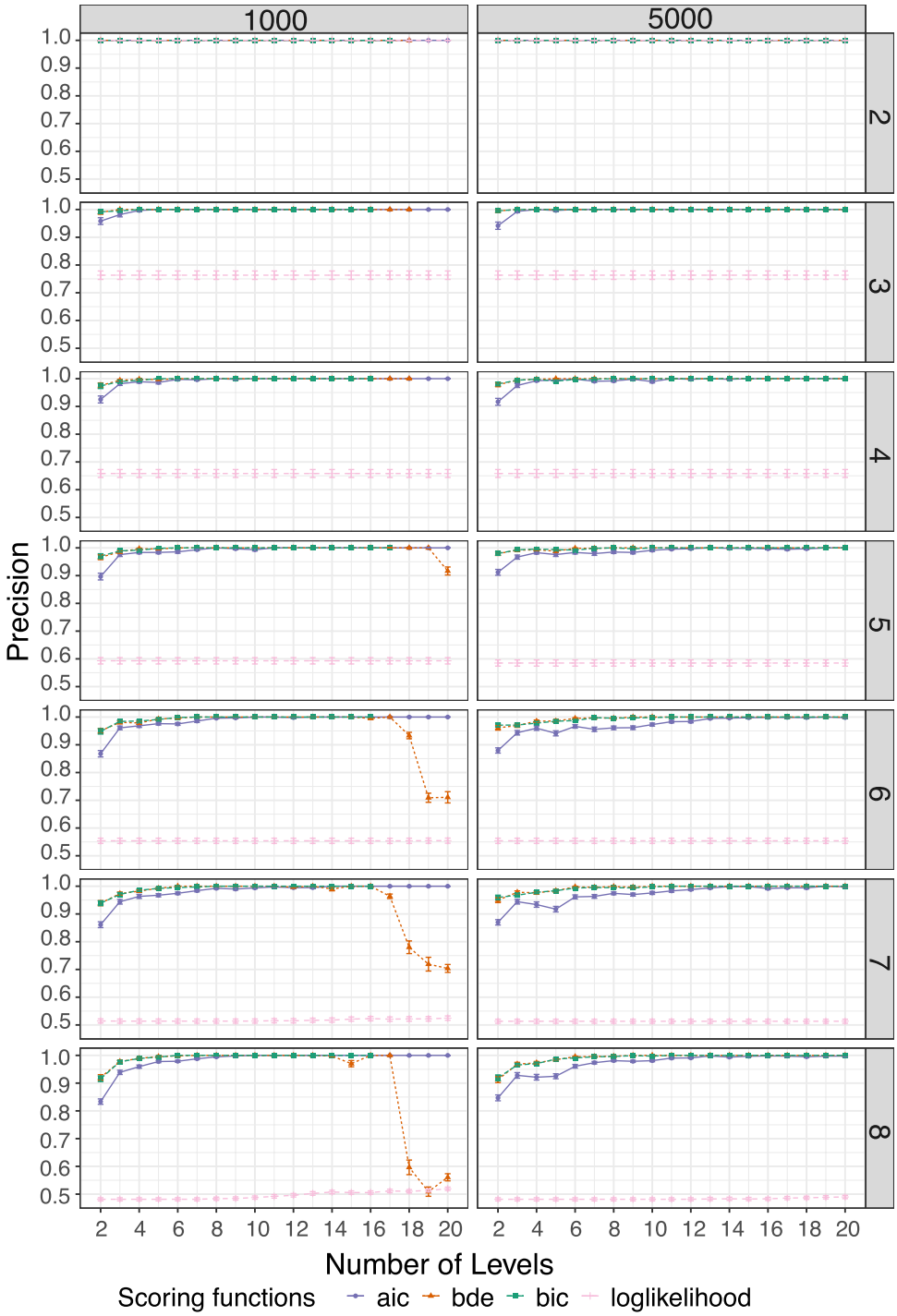

Fig. 4 Comparison of precision of four scoring functions (AIC, BDe, BIC, and log-likelihood) in the global simulation across different data sizes, number of discrete levels and variables. The imaginary sample size (iss) of BDe is 1. The x-axis represents the number of discrete levels, while the y-axis shows the precision. Each column panel corresponds to a different data size (1000 and 5000), and each row panel corresponds to a different number of variables (ranging from 2 to 8). Error bars indicate the standard error of precision across 100 repeats

higher but decreases steadily across discrete levels. Although more data points result in more consistent behavior, recall still decreases as discrete levels increase.

The comparison of the other three scoring functions (AIC, BIC, and BDe) shows distinct differences in their performance, particularly in how they handle varying numbers of discrete levels and sample sizes in learning Bayesian network structures. The AIC scoring function tends to perform better, showing higher recall values than BIC and BDe, but its performance decreases with more discrete levels. BIC and BDe show a similar trend but with slightly lower recall compared to AIC. For certain network scenarios with nodes above 5, particularly with 1000 data points, the recall for BDe increases sharply after a certain number of discrete levels (around level 18). This suggests that a more detailed representation of the variables in the Bayesian network allows the model to capture true dependencies more

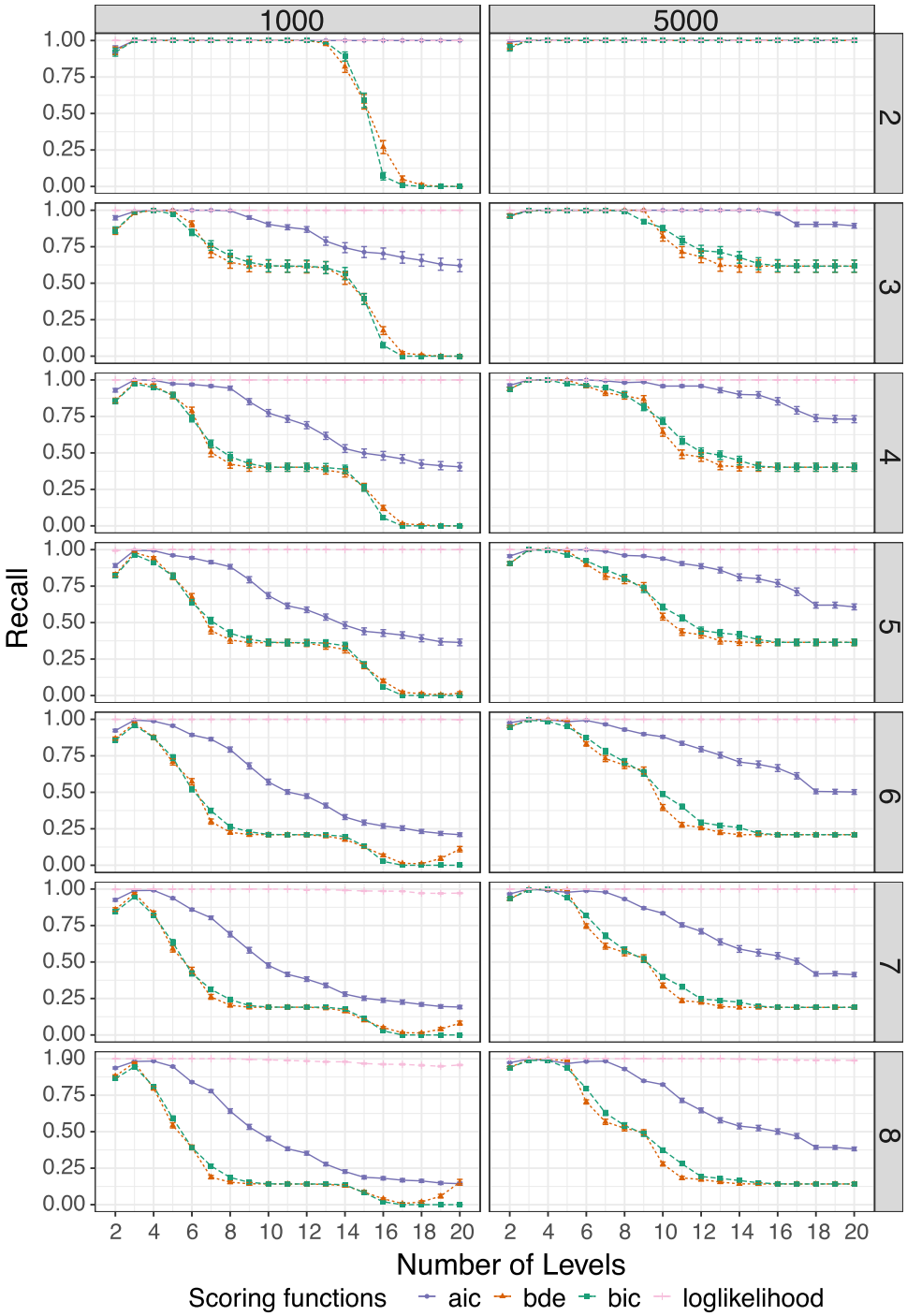

Fig. 5 Comparison of recall of four scoring functions (AIC, BDe, BIC, and log-likelihood) in the global simulation across different data sizes, number of discrete levels and variables. The imaginary sample size (iss) of BDe is 1. The x-axis represents the number of discrete levels, while the y-axis shows the recall. Each column panel corresponds to a different data size (1000 and 5000), and each row panel corresponds to a different number of variables (ranging from 2 to 8). Error bars indicate the standard error of recall across 100 repeats. The BDe score is represented by blue lines, AIC by orange, BIC by green, and log-likelihood by pink

effectively, improving recall by identifying true positives and reducing false negatives. However, it also suggests the importance of prior information, such as the *iss* value, for balancing model complexity with the risk of overfitting, especially with smaller datasets. In both sample sizes, recall decreases significantly for BIC and BDe after a certain number of discrete levels (around level 3 in 1000 data points and level 5 in 5000 data points), indicating difficulty in maintaining true positive rates with increasing complexity. In most cases, recall plateaus or stabilizes at a low level after reaching a high number of discrete levels (e.g., level 12 when the node count is 8 in 5000 data points), indicating that additional discrete levels with granular information do not significantly improve the model's ability to identify true dependencies and may even hinder it.

The trend of increasing precision with more discrete levels and larger sample sizes underscores the importance of detailed data and sufficient sample size for accurate network learning. BIC and BDe generally perform better in maintaining higher precision with larger sample sizes (5000 data points), indicating their effectiveness in balancing model fit and complexity. Conversely, recall decreases with more discrete levels, highlighting the challenge of capturing true dependencies in complex models. AIC maintains higher recall than others, but its performance diminishes as complexity increases. The log-likelihood score consistently shows high recall and low precision, suggesting overfitting and a tendency to learn complete networks. These patterns indicate a trade-off between recall and precision, necessitating careful consideration of each scoring function's strengths and weaknesses in real data applications.

3.1.2 Impact of *iss* on BDe in the overall network structure

Figure 6 illustrates the optimal *iss* values in BDe across different numbers of discrete levels and various sample sizes, highlighting how these optimal values change with increasing complexity and data size. The optimal *iss* value generally increases with the number of discrete levels, a trend more pronounced with larger sample sizes. For the same number of discrete levels, the optimal *iss* values are lower for 500 data points, increasing progressively with 1000, 5000, and 10,000 data points. The complexity of the network structure also impacts the optimum *iss* values. Figure 6A, which displays the 5-node network with a range up to 1000, highlights more gradual changes in *iss* values. Figure 6B, which shows the 10-node network with a range up to 10,000, captures broader trends and shows more substantial increases in *iss* values for higher numbers of discrete levels and larger sample sizes.

It displays how the optimal *iss* values for the BDe scoring function vary with the number of discrete levels, sample sizes, and network complexity. Larger sample sizes, more discrete levels, and more complex networks generally require higher *iss* values, indicating that more detailed data and larger, complex datasets influence the optimal setting of the *iss* hyperparameter in the BDe scoring function for effective network learning. This insight underscores the necessity of

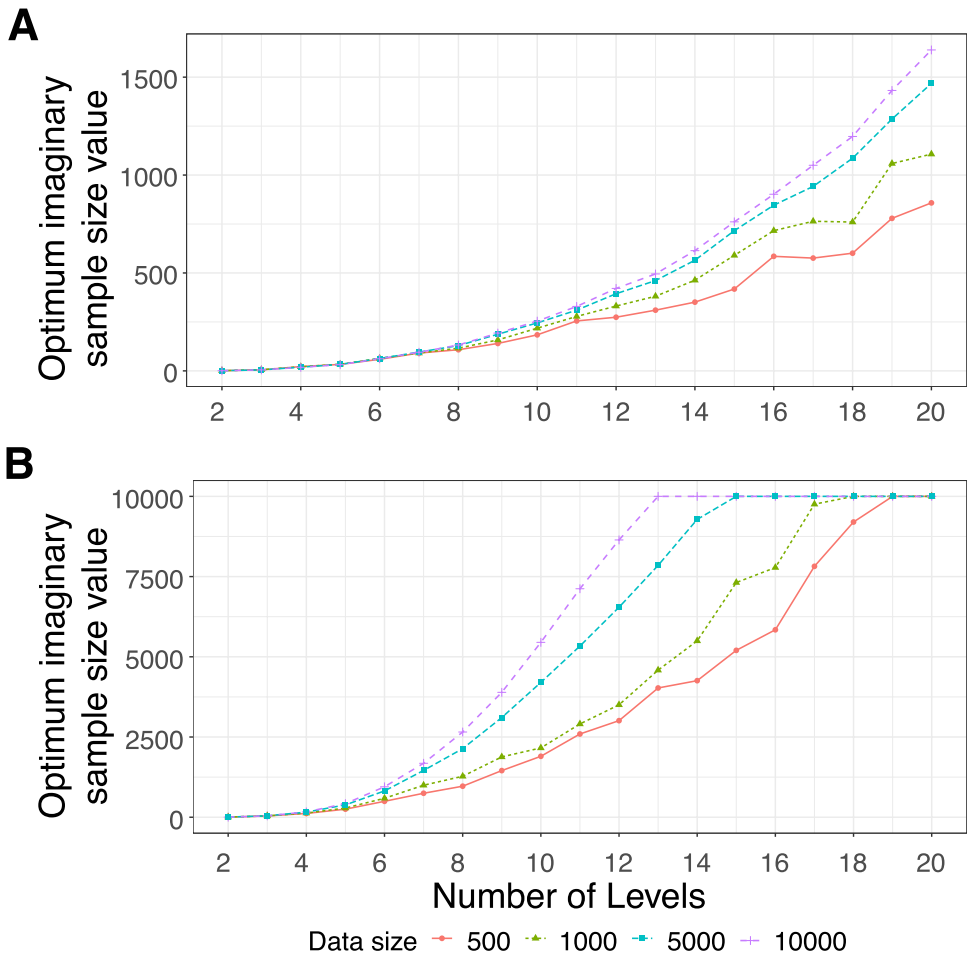

Fig. 6 Optimum imaginary sample size (iss) values required in BDe scoring function to find the true network structure. Panel A shows the simulation based on the 5-node network with each node having a maximum of 2 parents. Panel B is based on the 10-node network with each node having a maximum of 4 parents. The x-axis of both panels show the number of discrete levels in the simulated data set

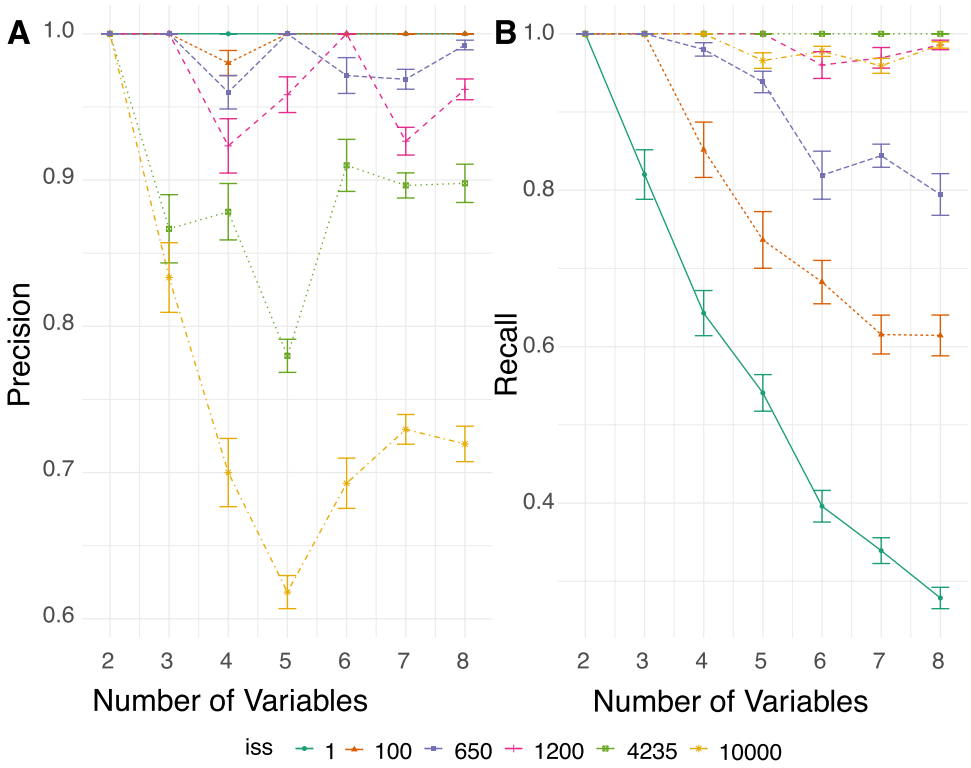

Fig. 7 Performance of the BDe scoring function with various imaginary sample size (iss) values in Bayesian network structural learning. Precision (**A**) and recall (**B**) are plotted against the number of variables in the simulated data. The x-axis represent the number of variables. Error bars indicate the standard error of precision and recall across 10 repeats (100 repeats for *iss* value 1). The simulated data set consists of 5000 data points with each variable having 10 discrete levels. Different lines correspond to different *iss* values (1, 100, 650, 1200, 4235, and 10,000)

selecting optimum *iss* values in BDe scoring function to balance model fit and complexity in Bayesian network structure learning.

We further evaluated the performance of the BDe scoring function with various *iss* values in recovering entire network structures, specifically focusing on how precision and recall vary with the number of variables when each variable has a fixed 10 discrete levels. As shown in Fig. 7, Fig. 7A indicates that precision generally fluctuates as the number of variables increases, with higher *iss* values maintaining lower precision for larger number variables (above 6 variables). The values *iss* value of 1 shows the highest precision, maintaining values close to 1.0 across all numbers of variables. Figure 7B shows that recall generally decreases as the number of variables increases, with lower *iss* values maintaining lower recall. However, an *iss* of 4235, identified as the optimal *iss* values in Fig. 6 for the 10-node network with level 10, shows the highest recall, maintaining values close to 1.0 even as the number of variables increases. These results demonstrate a trade-off between precision and recall. For instance, the precision of *iss* 650 is generally lower than *iss* 100 and *iss* 1, but its recall is higher. Additionally, although *iss* 4235 maintains the highest recall at larger numbers of variables (above 5), its precision is between *iss* 10,000 and 1200. This indicates that appropriately selecting *iss* values is crucial for effective network structure learning, especially in more complex scenarios.

3.2 Local simulation

3.2.1 Evaluation of scoring functions on focal node

We explored the patterns of these four scoring functions by counting the frequency of the positive score differences among 100 repeats in different scenarios. As shown in Fig. 8, the count of false positives for the log-likelihood function is always 100 across all discrete levels and sample sizes, while its count of false negatives is always 0. This indicates a strong overfitting problem with the log-likelihood function. Conversely, BIC consistently shows a very low count of false positives, starting higher initially with smaller sample sizes (50 or 100) and then decreasing to 0, remaining consistent as sample sizes grow. Its count of false negatives tends to start lower initially with smaller sample sizes (50 or 100) and then increases to 100, remaining consistent. This tendency diminishes as the sample sizes grows, with the count number starting higher at initial discrete levels.

The counts for BDe and AIC fluctuate more with the increase of discrete levels and sample sizes. In BDe, the count of false positives varies but generally increases with more discrete levels at small sample sizes (50 and 100 data

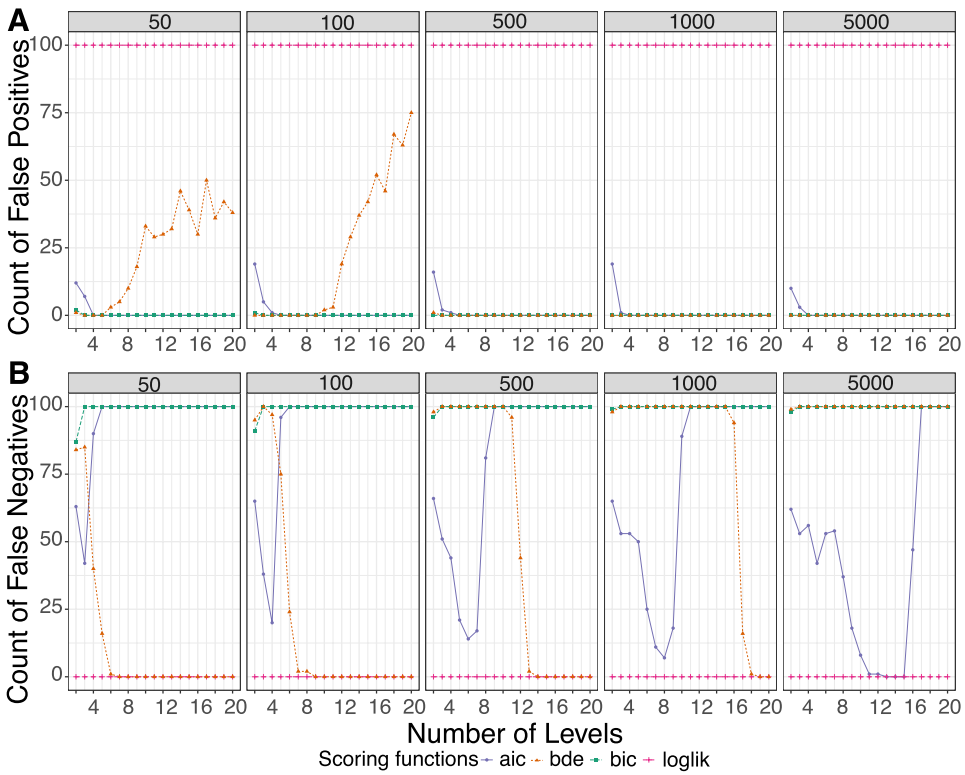

Fig. 8 The counts of positive score differences of four scoring functions (AIC, BDe, BIC, and log-likelihood) for the focal node in the three-node network scenario across 100 repeats. The imaginary sample size (iss) of BDe is set to its default value 1. The counts of positive score differences, indicating the frequency of false positives (**A**) and false negatives (**B**) among the 100 repeats, are plotted against the number of discrete levels (ranging from 2 to 20) for different sample sizes (50, 100, 500, 1000, and 5000). The y-axis represents the count of the positive score differences (Test Network—True Network), indicating the presence of a false positive arc (**A**) or a false negative arc (**B**)

points), remaining low (around 0) at higher sample sizes. Its count of false negatives starts high at initial discrete levels but then drops sharply to 0, with the dropping point increasing as sample sizes grow (e.g., level 6 with 50 data points, level 7 with 100 data points, level 11 with 500 data points, level 16 with 1000 data points, and staying around 100 at 5000 data points). AIC shows a more consistent pattern of false positives, starting high initially and dropping sharply to about 0 around level 4. Its count of false negatives starts at a middle level, drops sharply at level 3, and then increases to a higher point, with the increasing point shifting higher as sample sizes grow (e.g., level 4 with 50 data points, level 5 with 100 data points, level 7 with 500 data points, level 10 with 1000 data points, and level 16 with 5000 data points).

These patterns indicate that the presence of false positives and false negatives is closely associated with discrete discrete levels and sample sizes for scoring functions AIC, BIC, and BDe. The log-likelihood function has a strong tendency to overfit, making it unsuitable for real data applications. BIC performs well in avoiding false positives even with limited sample sizes but tends to miss arcs in the learned network structures, leading to false negatives. In AIC, there is a trade-off between false positives and false negatives for certain ranges of discrete levels, which diminishes with larger sample size. With enough data points, AIC can avoid both false positives and false negatives across a broader range of discrete levels. Similarly, BDe shows a trade-off between false positives and false negatives in low sample size scenarios, which diminishes with larger sample sizes. BDe performs well with enough data points and more granular information (i.e., more discrete levels). This suggests that the performance of BDe with the default *iss* value of 1 is sensitive to the number of discrete levels and sample sizes. This indicates the need to find better alternative *iss* values, particularly when dealing with data with low number of discrete levels and high sample sizes.

3.2.2 Impact of *iss* in BDe on the focal node

We analyzed how the *iss* values (from 0.1 to 10,000) are associated with the presence of false negative and false positive arcs across various discrete levels (from 2 to 20) and four network structures, as shown in Fig. 3. The score differences of each condition were averaged across 10 repeats, and the *iss* values were selected when their corresponding averaged score differences were positive. As displayed in Fig. 9, the presence of false positives or false negatives is influenced by different ranges of *iss* values. Interestingly, all tested *iss* values in the two-node and three-node networks do not lead to false positives across all discrete levels. However, certain ranges of tested *iss* values in all network scenarios cause false

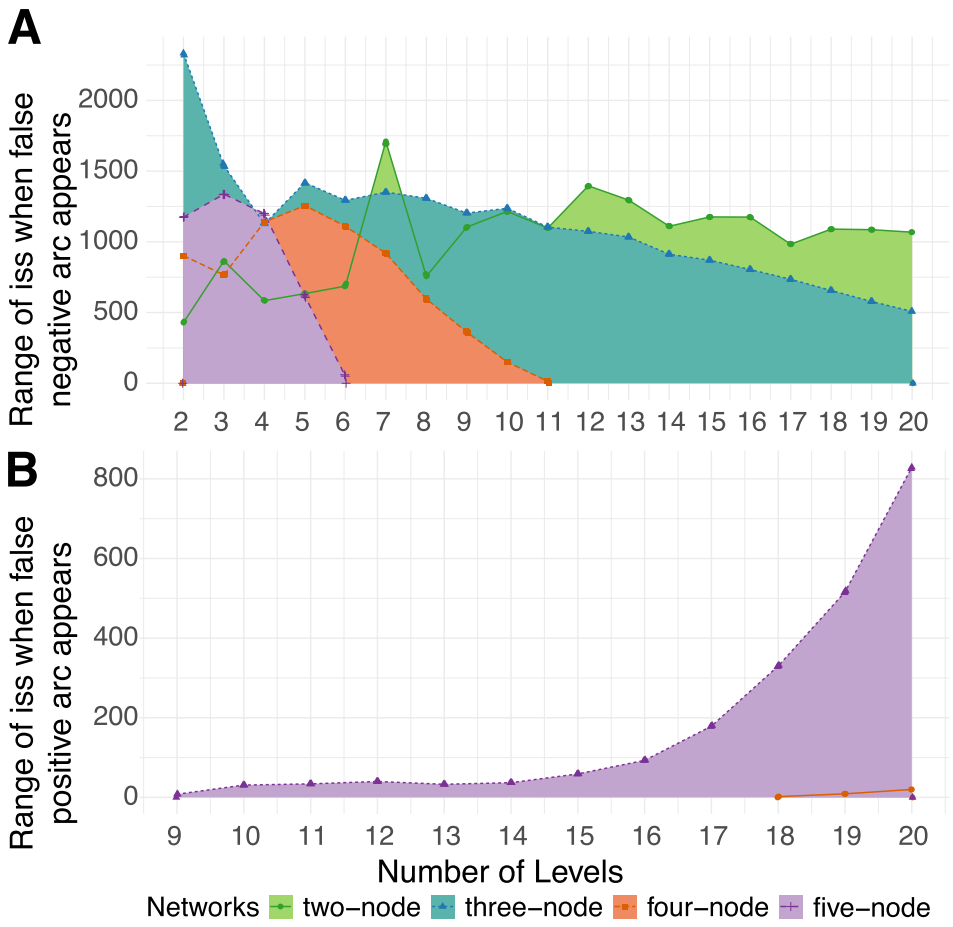

Fig. 9 Analysis of the imaginary sample size (iss) values in the BDe scoring function was performed when false negative and false positive arcs appear in four scenarios across various discrete levels. The sample size of the simulated data is 5000. The x-axis represent the number of discrete levels ranging from 2 to 20. Panel A and B display the range of *iss* values for false negative and false positive arcs, respectively, with a varying number of discrete levels. The scenarios are categorized by the network structure: two-node (green), three-node (orange), four-node (purple), and five-node (teal). Each simulation was repeated 10 times. The *iss* values were selected when their corresponding averaged score differences of the focal node (test—true) across the 10 repeats were positive

negatives. For instance, the range of *iss* values from 0.1 to 1337 leads to false negatives at level 3 in the five-node network scenario. Conversely, the entire range of tested *iss* values (0.1 to 10,000) does not cause false positives at any tested level (2 to 20) in the two-node and three-node network scenarios. This indicates that, in general, the presence of false negative arcs is more sensitive to *iss* values. However, we found that all tested *iss* values allow certain ranges of discrete levels to avoid false negatives, and this range tends to decrease as the complexity of the network structures increases. In other words, while all discrete levels need to be cautious with chosen *iss* values to avoid false negatives in the two-node and three-node network scenarios, discrete levels from 11 to 20 with any tested *iss* value can avoid false negatives in the four-node network scenario, and discrete levels 6 to 20 in the four-node network scenario.

The association pattern between the range of *iss* values and the number of discrete levels in the presence of false negatives is not very straightforward. In the two-node network scenario, the range of *iss* values fluctuates as the number of discrete levels increase. In the three-node network scenario, the range generally decreases as the number of discrete levels increases. In the four-node and five-node network scenarios, the range decreases from its highest point as the number of discrete levels increases. For instance, in the four-node scenario, the maximum *iss* value decreases from 1256 at level 5 until 0 at level 11. However, the pattern is more apparent in the presence of false positives. There are no false positives in the two-node and three-node scenarios. The range of *iss* values causing false positives expands as the number of discrete levels increases in the four-node and five-node network scenarios, while it remains minimal in the four-node scenario. The appearance of false positives in more complex networks could potentially be due to the lack of observations for completing the CPTs when there are also high number of discrete levels [37]. These simulation results indicate that real data with varying number of variables, discrete levels, and complexities require different *iss* values to optimize the performance of the BDe scoring function.

4 Discussion

4.1 Performance of scoring functions and their implications

This study systematically evaluated the performance of four scoring functions—AIC, BIC, BDe, and log-likelihood—in BN structure learning, with a particular focus on polychotomous data. The results from both Global and Local Simulations provide several key insights into the effectiveness of these scoring functions under various conditions. Firstly, BIC and BDe (with default *iss* value of 1) generally offer higher precision (fewer false positives) than AIC and log-likelihood, especially with larger sample sizes and fewer discrete levels. BIC's higher penalty for model complexity, due to its $\ln(N)$ term, makes it more conservative and effective in avoiding overfitting comparing to AIC, particularly as the sample size increases. This finding is consistent with previous studies that indicate BIC's preference for simpler models with sufficient data [14, 20]. On the other hand, AIC generally offers higher recall (fewer false negatives) than BIC and BDe (with default *iss* value of 1), especially with larger sample sizes and higher discrete levels. However, AIC's performance can be erratic, depending on the dataset and conditions, balancing model fit and complexity but sometimes behaving unpredictably, as noted in prior research [13, 20]. Our findings echo another study showing smaller Hamming distance for AIC only for a highly polychotomous network (up to 21 levels), whereas BIC or BDe performed better on other networks with ≤ 11 discrete levels [25]. The log-likelihood scoring function, due to its lack of a complexity penalty, tends to overfit, showing high recall but low precision by favoring more complex networks that capture more data points but also introduce noise [9].

The performance of BDe is highly sensitive to the hyperparameter α (*iss*). Our results indicate that BDe with the default *iss* value of 1 performs inconsistently, with both precision and recall increasing as the number of discrete levels rises in low sample sizes. This suggests that optimizing *iss* is crucial for balancing model complexity and fit to achieve optimal performance. Larger sample sizes and higher discrete levels require larger optimal *iss* values, whereas smaller sample sizes and fewer discrete levels necessitate smaller *iss* values. Networks with more parents per node also demand larger optimal *iss* values. These findings align with theoretical foundations that highlight the influence of *iss* on network complexity [29]. Furthermore, the optimal *iss* value varies depending on the skewness of the sample distribution, with more skewed distributions favoring smaller *iss* values [30, 31].

There is no straightforward analytical solution to determine the relationship between false positive or false negative arcs and optimal optimal *iss* values in the BDe scoring function equation. Although many studies have attempted to understand the impact of *iss* on BDe's performance in recovering true network structures, the complexity remains [29–31, 31, 32]. Consequently, we propose here a simulation method to identify the optimal *iss* for the BDe scoring function when dealing with real-word polychotomous data analysis. The steps are as follows:

1. Generate a random network structure with the same number of nodes/variables as the real-world data set.
2. Parameterize the network with discrete levels identical to those in the real-world data set.
3. Sample data from the parameters with the same sample size as the real-world data set.
4. Score the network structure along with the sampled data using varying *iss values in the BDe scoring function.*
5. Identify the *iss* value that gives the network structure the highest score.

These findings suggests that scoring functions must be used cautiously in real-world data analysis, considering the study's purpose. For instance, in causal inference, it is crucial to avoid false positives to prevent biased causal relationships, making AIC unsuitable for such studies. Although BIC has high precision, its low recall in handling high discrete levels indicates a significant trade-off, making it more suitable for situations where a small number of accurate inferences are desired, such as when leading to expensive biological intervention experiments.

4.2 Contribution of methodology and study limitations

Our study used randomly generated or specifically designed networks for data sampling; however, the simulated data generated may not share the same properties as real-world data. Simulated data allows for controlled testing and ensures a wide range of scenarios, but it lacks the inherent complexity and noise found in real-world datasets. Real-world data includes unique patterns, dependencies, measurement errors, missing values, and non-stationarity,

which can significantly impact the performance of BN algorithms. In contrast, simulated data is generated under idealized conditions that do not account for these imperfections.

In our simulations, we assigned the parameters to the network structures or fitted the network structures to the simulated datasets. The complexities of real-world datasets are often challenging to replicate in simulations, potentially leading to an overestimation of the effectiveness of scoring functions when applied to practical scenarios. The diversity of real-world data, which spans various sample sizes, distributions, and intricate interactions, contrasts with the controlled nature of simulated data typically generated to follow specific distributions and constraints. Future research should incorporate more realistic datasets in simulations (e.g., simulate noise) and address the unique challenges posed by real-world data. By doing so, researchers can better understand how Bayesian network algorithms perform in practical applications, leading to more robust and generalizable findings.

Here, we have not incorporated real-world data scenarios due to the extensive variety and different priorities (e.g., precision vs recall) required for each type of real-world data, which would necessitate several detailed examples and are beyond the scope of this study. Additionally, starting with simulations is essential to establish a baseline understanding of the scoring functions' performance, upon which the complexities and variations of real-world data can later be added.

4.3 Insights and future directions

4.3.1 Potential analytical solutions to optimizing α in BDe

These findings raise a crucial question in the field of BN structure learning: how to determine the optimum hyperparameter for the BDe score. While we propose a simulation-based methodology for identifying this hyperparameter in real-world data analysis, this approach is not a principled solution. A potential avenue for finding a principled solution involves deriving an analytical method for determining the optimum hyperparameter.

Understanding the relationship between the hyperparameter α (*iss*) and the maximum value of the BDe score under conditions constrained by the properties of real-world datasets (e.g., maximum number of states in the categorical variables) would be of interest. However, there is no straightforward analytical solution to obtain such relationships from the equation of the BDe scoring function. Neural networks might offer a way to uncover this relationship, but they come with significant limitations. Firstly, neural networks require thousands of data points for training, which is impractical to obtain from empirical simulations. Secondly, neural networks operate as a black box, meaning that even if they could model the relationship, the exact nature of the relationship between α and the maximum value of BDe would remain unclear. Therefore, further research is necessary to explore these relationships comprehensively and to develop a principled analytical method for optimizing α in the BDe score.

4.3.2 Dirichlet prior in BDe

The Dirichlet prior represents prior beliefs about the probability distributions of variables, specifying prior knowledge about the CPTs of the network. However, its impacts on the performance of the BDe score in BN structure learning is not immediately intuitive. The Dirichlet distribution itself is a distribution over probability distributions, which adds complexity to the analysis by emphasizing the nuanced interplay between prior information and observed data.

Jeffreys' prior, introduced by Harold Jeffreys, is an invariant objective or noninformative prior designed to impart minimal information, maintaining objectivity under monotone transformations [38]. Jeffreys' prior can serve as an alternative to the Dirichlet prior in situations where a noninformative prior is required. It is extensively studied because of its invariance properties and its capacity to provide an objective prior that does not favor any specific outcome. Clarke and Barron elucidated these properties, demonstrating Jeffreys' prior's capability to enhance model performance under certain conditions [39]. Suzuki further examined the asymptotic properties of the BDe score when using Jeffreys' prior, showcasing its practical applications in BN structure learning [40]. Additionally, Natori et al. highlighted Jeffreys' prior's effectiveness in improving the Conditional Independence (CI) test, particularly when integrated with algorithms like RAI [41]. These studies collectively underscore the critical role of Jeffreys' prior in refining BN structure learning, emphasizing both its theoretical importance and practical effectiveness in diverse applications.

Interestingly, while Jeffreys' prior is considered less effective than the BDe score in a score-based approach, it is optimal in the context of Bayes factors in the constraint-based approach [41]. This raises pertinent questions about

its potential application in identifying the optimum hyperparameter α for the BDe score. Jeffreys' prior is constructed using the square root of Fisher Information, which measures the amount of information that an observable variable carries about an unknown parameter. This relationship ensures that Jeffreys' prior is invariant under reparameterization, meaning it does not introduce bias due to the choice of parameterization [38]. Leveraging this property, Jeffreys' prior could potentially provide a principled way to calibrate α in the BDe score, offering insights into optimizing BN structure learning by balancing accuracy and uncertainty quantification. This presents an interesting direction for future research.

4.3.3 Hidden assumptions in simulation studies

In BN structure learning, traditional methods score different structures for a given fixed dataset and use heuristic methods to find the best-fitted structure with the highest score. The underlying assumption is that the given data is fully observed and generated independently and identically distributed (*iid*) from a certain underlying distribution. The objective is to find a network structure that captures the set of independencies in the data, approximating that true underlying distribution. However, in this simulation work, instead of assuming a true underlying distribution defined by the observed data, we assume there is a true joint probability distribution represented by BNs of the same equivalent class sharing the same set of dependencies/independencies. The objective becomes to find the BNs given the data (with varying sizes and number of discrete levels) that are induced by this joint probability distribution. This assumption is reasonable for real-world data analysis. For instance, if social survey data includes two independent variables, such as site and gender, their independence should be identifiable in the resulting network structure, regardless of the number of discrete levels and sample size of the collected data. However, the nuances between these two assumptions are often overlooked in simulations, yet they can potentially impact the results. More research is needed to explore this aspect to enhance the robustness and applicability of BN structure learning methods.

4.3.4 Additional directions for future research

Despite the insights gained, several aspects remain unexplored. The effect of the number of discrete levels on the penalty term in other scoring metrics is still unclear. Determining the optimal scoring function for scenarios involving a high number of discrete levels but sufficient data is crucial. This "sufficiency" likely depends on the number of parents for individual nodes and the number of discrete levels, factoring in network complexity and the study's purpose. Furthermore, the relationship between score changes and the addition or deletion of arcs in BDe requires a more intuitive interpretation. Future research should address these areas by developing new theoretical frameworks or conducting empirical studies to deepen our understanding. By filling these gaps, we can improve the accuracy and applicability of BN structure learning in diverse and complex real-world scenarios.

5 Conclusion

Our study evaluated the performance of four scoring functions—AIC, BIC, BDe, and log-likelihood in Bayesian network structure learning using global and local simulations. We found that BIC and BDe (with default *iss* value 1) generally offered higher precision, particularly with larger sample sizes, while log-likelihood tended to overfit, showing high recall but low precision. The optimization of the *iss* parameter in BDe was essential for balancing model complexity and fit. Local simulations indicated that BIC minimized false positives effectively, and AIC and BDe required careful tuning based on discrete levels and sample sizes. We provide a detailed comparison of scoring functions and a proposed method for optimizing *iss* values, enhancing Bayesian network performance in real-world applications. These findings offer practical guidance for selecting scoring functions in BN structure learning: for scenarios such as causal analysis where prioritizing precision and minimizing false positives are crucial, BDe is recommended, and optimizing the *iss* parameter in BDe can be particularly beneficial. For a balanced approach between precision and recall, AIC is recommended. Ultimately, the choice of scoring function should align with the specific goals and constraints of the application, whether it is precision, recall, or a balance of both. This work highlights the importance of selecting appropriate scoring functions and adjusting their parameters in Bayesian network analysis, providing valuable guidelines for more precise and effective modeling of complex systems.

Acknowledgements The authors acknowledge Research Computing at the James Hutton Institute for providing computational resources and technical support for the "UK's Crop Diversity Bioinformatics HPC"(BBSRC grants BB/S019669/1 and BB/X019683/1), use of which has contributed to the results reported within this paper.

Author contributions XK performed the simulation and analyses on simulated data, designed the figures and wrote the initial draft of the manuscript. VAS assisted XK in analysis of simulation results; VAS and KK assisted XK in interpretation of results. All authors conceptualised general study idea. All authors revised and agreed on the final manuscript.

Funding XK is funded by a St Leonard's College Doctoral Scholarship provided by the University of St Andrews.

Data availibility The code underlying the simulations reported in this study is accessible at https://github.com/thebestecho/Bayesian-network-scoring-functions.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare no competing interests.

References

1. Pearl J. Bayesian networks: A model of self-activated memory for evidential reasoning. In: Proceedings of the 7th conference of the cognitive science society, University of California, Irvine, CA, USA; 1985. p. 15–7.
2. Werhli AV, Husmeier D. Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. Stat App Genet Mol Biol. 2007;6:15.
3. Yu J, Smith VA, Wang PP, Hartemink AJ, Jarvis ED. Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics. 2004;20(18):3594–603.
4. Wu Y, Mascaro S, Bhuiyan M, Fathima P, Mace AO, Nicol MP, Richmond PC, Kirkham L-A, Dymock M, Foley DA. Predicting the causative pathogen among children with pneumonia using a causal Bayesian network. PLoS Comput Biol. 2023;19(3):1010967.
5. McLachlan S, Dube K, Hitman GA, Fenton NE, Kyrimi E. Bayesian networks in healthcare: distribution by medical condition. Artif Intell Med. 2020;107: 101912.
6. Keenan K, Papathomas M, Mshana SE, Asiimwe B, Kiiru J, Lynch AG, Kesby M, Neema S, Mwanga JR, Mushi MF, Jing W, Green DL, Olamijuwon E, Zhang Q, Sippy R, Fredricks KJ, Gillespie SH, Sabiiti W, Bazira J, Sloan DJ, Mmbaga BT, Kibiki G, Aanensen D, Stelling J, Smith VA, Sandeman A, Holden MTG, HATUA Consortium. Intersecting social and environmental determinants of multidrug-resistant urinary tract infections in East Africa beyond antibiotic use. Nat Commun. 2024;15:9418.
7. Cézard G, Gruer L, Steiner M, Douglas A, Davis C, Buchanan D, Katikireddi SV, Millard A, Sheikh A, Bhopal R. Ethnic variations in falls and road traffic injuries resulting in hospitalisation or death in Scotland: the Scottish Health and Ethnicity Linkage Study. Public Health. 2020;182:32–8.
8. Catney G, Lloyd CD, Ellis M, Wright R, Finney N, Jivraj S, Manley D. Ethnic diversification and neighbourhood mixing: a rapid response analysis of the 2021 census of England and Wales. Geogr J. 2023;189:63–77.
9. Koller D, Friedman N. Probabilistic graphical models: principles and techniques. Cambridge: MIT press; 2009.
10. Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn. 1992;9:309–47.
11. Chickering DM. Learning Bayesian networks is NP-complete. In: Learning From data: artificial intelligence and statistics V. New York: Springer; 1996. p. 121–30.
12. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Mach Learn. 1995;20:197–243.
13. Akaike H. Information theory and an extension of the maximum likelihood principle. In: Selected Papers of Hirotugu Akaike. New York: Springer; 1998. p. 199–213.
14. Schwarz G. Estimating the dimension of a model. Ann Stat. 1978;6:461–4.
15. De Campos LM, Friedman N. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. J Mach Learn Res. 2006;7(10):2149–87.

16. Silander T, Roos T, Kontkanen P, Myllymäki P. Factorized normalized maximum likelihood criterion for learning Bayesian network structures. In: Proceedings of the 4th European workshop on probabilistic graphical models (PGM-08); 2008. p. 257–72.
17. Carvalho AM, Roos T, Oliveira AL, Myllymäki P. Discriminative learning of Bayesian networks via factorized conditional log-likelihood. J Mach Learn Res. 2011;12(7):2181–210.
18. Lam W, Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. Comput Intell. 1994;10(3):269–93.
19. Scutari M. Learning Bayesian networks with the bnlearn R package. J Stat Softw. 2010;35:3.
20. Liu Z, Malone B, Yuan C. Empirical evaluation of scoring functions for Bayesian network model selection. BMC Bioinform. 2012;13(Suppl 15):14.
21. Bozdogan H. Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. Psychometrika. 1987;52(3):345–70.
22. Dojer N. Learning Bayesian networks from datasets joining continuous and discrete variables. Int J Approx Reason. 2016;78:116–24.
23. Yang S, Chang K-C. Comparison of score metrics for Bayesian network learning. IEEE Trans Syst Man Cybernetics Part A Syst Humans. 2002;32(3):419–28.
24. Allen TV, Greiner R. Model selection criteria for learning belief nets: an empirical comparison. In: Proceedings of the seventeenth international conference on machine learning. ICML '00. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA; 2000. p. 1047–54.
25. Chua CY, Ong HC. Comparison of scoring functions on greedy search Bayesian network learning algorithms. Pertanika J Sci Technol. 2017;25(3):719–34.
26. Steck H, Jaakkola TS. On the Dirichlet prior and Bayesian regularization. In: Becker S, Thrun S, Obermayer K, editors. Adv Neural Inf Process Syst NIPS. Cambridge, MA: MIT Press; 2002. p. 697–704.
27. Silander T, Kontkanen P, Myllymäki P. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In: Laskey KB, Mahoney SM, Goldsmith J, editors. Proceedings of the 23rd conference on uncertainty in artificial intelligence. Morgan Kaufmann, Vancouver, Canada; 2007. p. 360–7.
28. Ueno M. Learning likelihood-equivalence Bayesian networks using an empirical Bayesian approach. Behaviormetrika. 2008;35(2):115–35.
29. Ueno M. Learning networks determined by the ratio of prior and data. In: Grunwald P, Spirtes P, editors. Proceedings of the twenty-sixth conference on uncertainty in artificial intelligence. AUAI Press, Arlington, VA; 2010. p. 598–605.
30. Steck H. Learning the Bayesian network structure: Dirichlet prior versus data. In: Proceedings of the 24th conference on uncertainty in artificial intelligence (UAI). AUAI Press, Arlington, VA; 2008. p. 511–8.
31. Ueno M. Robust learning Bayesian networks for prior belief; 2012. arXiv preprint arXiv:1202.3766
32. Castillo E, Hadi AS, Solares C. Learning and updating of uncertainty in Dirichlet models. Mach Learn. 1997;26:43–63.
33. Martin AD, Quinn KM, Park JH. MCMCpack: Markov Chain Monte Carlo in R. J Stat Softw. 2011;42(9):22.
34. Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. Mach Intell Pattern Recogn. 1988;5:149–63.
35. Glover F. Tabu search—part I. ORSA J Comput. 1989;1:190–206.
36. Ke X, Keenan K, Smith VA. Treatment of missing data in Bayesian network structure learning: an application to linked biomedical and social survey data. BMC Med Res Methodol. 2022;22(1):326.
37. Yu J. Developing Bayesian network inference algorithms to predict causal functional pathways in biological systems. Phd thesis, Duke University, Durham, NC, USA; 2005.
38. Jeffreys H. Theory of probability. 3rd ed. Oxford: Clarendon Press; 1961.
39. Clarke BS, Barron AR. Jeffreys' prior is asymptotically least favorable under entropy risk. J Stat Plan Inference. 1994;41:37–60.
40. Suzuki J. A theoretical analysis of the BDeu scores in Bayesian network structure learning. Behaviormetrika. 2017;44:97–116.
41. Natori K, Uto M, Nishiyama Y, Kawano S, Ueno M. Constraint-based learning Bayesian networks using Bayes factor. In: Advanced methodologies for Bayesian networks: second international workshop, AMBN 2015, Yokohama, Japan, November 16–18; 2015. Proceedings 2. Cham, Switzerland: Springer; 2015. p. 15–31.