Socio-Economic Planning Sciences 95 (2024) 102030

Contents lists available at ScienceDirect

Socio-Economic Planning Sciences

journal homepage: www.elsevier.com/locate/seps

Robust learning of staged tree models: A case study in evaluating transport services

Manuele Leonelli [a,*], Gherardo Varando [b]

[a] School of Science and Technology, IE University, Madrid, Spain
[b] Image Processing Laboratory, Universitat de Valencia, Valencia, Spain

ARTICLE INFO

Keywords:
Bayesian networks
Conditional independence
Service evaluation
Staged trees
What-if analysis

ABSTRACT

Staged trees are a relatively recent class of probabilistic graphical models that extend Bayesian networks to formally and graphically account for non-symmetric patterns of dependence. Machine learning algorithms to learn them from data have been implemented in various pieces of software. However, to date, methods to assess the robustness and validity of the learned, non-symmetric relationships are not available. Here, we introduce validation techniques tailored to staged tree models based on non-parametric bootstrap resampling methods and investigate their use in practical applications. In particular, we focus on the evaluation of transport services using large-scale survey data. In these types of applications, data from heterogeneous sources must be collated together. Staged trees provide a natural framework for this integration of data and its analysis. For the thorough evaluation of transport services, we further implement novel what-if sensitivity analyses for staged trees and their visualization using software.

1. Introduction

Probabilistic graphical models (PGMs) represent the interactions between variables of interest using a graph. They decompose a joint probability mass function (pmf) into the product of local, smaller dimensional conditional pmfs, which are determined by the underlying graph. Because of this decomposition, they are tailored for the integration of heterogeneous data and expert sources, since each source can coherently and independently inform different local distributions [1–3]. Bayesian networks (BNs) are undoubtedly the most commonly used PGM, describing the underlying dependence structure through a directed acyclic graph (DAG) [4].

One of the limitations of BNs is that they can formally encode only symmetric conditional independences between variables, determined by the famous d-separation criterion (e.g. [5]). However, several studies have now shown that more complex patterns of dependence are required to faithfully describe the information stored in collected data [6–9]. The most classical non-symmetric independence is context-specific [10], where independence holds only for a specific subset of values of the conditioning variables. More generic types of non-symmetric independence have been defined and studied in the context of PGMs [11].

Although it was recognized early on the need for PGMs embedding non-symmetric independence [12,13], the development of such models has been limited. Staged trees [14,15] are a class of non-symmetric PGMs that embed flexible types of non-symmetric independence by coloring the vertices of a probability tree. There is now an extensive toolkit of methodologies for analyzing and learning staged trees from data, as well as user-friendly software for their use, including multiple structural learning algorithms for the identification of non-symmetric independence from data [16,17].

Much attention has been devoted to the development of robust learning algorithms for BNs, often based on the use of sampling techniques. Friedman et al. [18] and Caravagna and Ramazzotti [19] investigated the use of data bootstrapping to learn the underlying DAG: a graph is learned for each bootstrap replication and edges in the final BN are chosen depending on how often they appeared [20]. Friedman and Koller [21] introduced Bayesian MCMC approaches of learning, whose use is becoming increasingly popular [22–26]. Cugnata et al. [27] discussed the averaging of BNs learned with different structural learning algorithms: a technique used in several practical applications (e.g. [28–30]). Cross-validation is often further used to select the best BN structure with the aim of avoiding overfitting (e.g. [31]).

Despite the widespread use of such techniques for BNs, their use has not been discussed for staged trees. Structural learning algorithms are usually evaluated over the complete dataset, thus risking overfitting and identifying relationships that do not actually represent the underlying dependence structure. In this direction, Strong and Smith [32]

* Corresponding author.
E-mail address: manuele.leonelli@ie.edu (M. Leonelli).

https://doi.org/10.1016/j.seps.2024.102030
Received 31 December 2023; Received in revised form 3 July 2024; Accepted 22 July 2024
Available online 31 July 2024
0038-0121/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

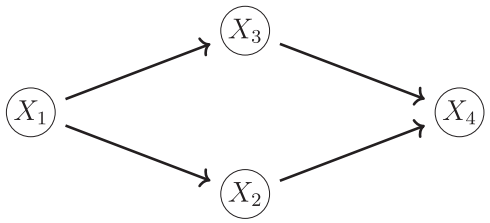

Fig. 1. An example of a DAG over four random variables X_1, X_2, X_3, X_4.

introduced a Bayesian model averaging approach to jointly consider high-scoring staged trees for more robust modeling.

This paper introduces robust routines for learning non-symmetric conditional independence via staged trees, discussing their methodological and software implementation. Methodologically, we first discuss the robust choice of a variables' ordering using bootstrap. Once such an order is identified, bootstrap and cross-validation are jointly used to identify the best staged tree model from data. Software-wise, we take advantage of the capabilities of the `stagedtrees` R package [16].

Our novel, robust learning methods are illustrated to untangle complex dependence patterns in survey data for the evaluation of transport services. BNs have been used for this task in metro [30,33–36], railways [37], airlines [27], airport check-in [29], and air and high-speed rail intermodal [38] services. The comprehensive evaluation of transport services requires the integration of heterogeneous data sources, often collected at different spatial or temporal resolutions. This integration is exemplified in our second data application below about the quality of railway services in the European Union.

A thorough assessment of the factors related to the evaluation of transport services requires the use of sensitivity techniques (e.g. [39]). In the context of BNs, what-if analyses are widely used, where specific vertices are assumed to be observed and the effect of these observations on output of interest is measured (e.g. [27]). In our applications below, we showcase a novel visual way to perform what-if analyses in staged trees.

The paper is structured as follows. Section 2 deals with PGMs, reviewing BNs and staged trees. Section 3, after reviewing structural learning algorithms for staged trees, introduces robust routines based on the bootstrap. Section 4 showcases these methods over two data applications: the first in airline passengers' satisfaction, the second based on a large-scale European survey on railways satisfaction. The paper concludes with a discussion.

2. Non-symmetric probabilistic graphical models

PGMs [40] are a popular class of statistical models that use various graphical representations to visually depict the dependence structure between variables of interest. In this section, we review two instances of PGMs: the most common BN model and the staged tree model.

2.1. Bayesian networks and conditional independence

First introduced by Pearl [4], BNs are a class of PGMs that use DAGs to formally encode independence information. They are now the gold standard for representing causal information and discovering it from observational data [41]. The variables of interest are the vertices of the DAG. The edges denote probabilistic direct dependence and, in some cases, causal relationships. Although BNs can be used with continuous and mixed variables, henceforth, and as common in practice, we focus on the categorical case. Let $X = (X_1, \dots, X_p)$ be a categorical random vector and $x = (x_1, \dots, x_p)$ an instantiation of this vector.

One of the main features of BNs is that they decompose the joint pmf of X, $P(x)$, into a product of smaller-dimensional conditional pmfs. The form of these conditional probabilities is defined by the DAG G. Let X_{Π_i} be the parents of X_i in G. The joint pmf of a BN with DAG G can be written as

$$P(x) = \prod_{i=1}^{p} P(x_i | x_{\Pi_i}), \tag{1}$$

where Π_i could be empty (by construction, there must be at least one vertex with no parents in a DAG). For instance, the DAG in Fig. 1 induces the factorization

$$P(x) = P(x_4|x_3, x_2)P(x_3|x_2)P(x_2|x_1)P(x_1). \tag{2}$$

The great benefit associated with the decomposition in Eq. (1) is that the number of parameters defining the model can decrease dramatically. In our example, if variables are assumed to be binary, nine parameters are required to fully characterize the BN, in contrast to fifteen generally required to define the pmf of four binary variables ($2^4 - 1$, since probabilities must sum up to one).

The lack of edges in the DAG of a BN formally encodes symmetric *conditional independence* [42]. We say that X_i is conditionally independent of X_j given X_k, and write $X_i \perp\!\!\!\perp X_j | X_k$, if and only if

$$p(x_i|x_j, x_k) = p(x_i|x_k), \tag{3}$$

for all possible values x_i, x_j, x_k. The factorization of the pmf in Eq. (1) is associated with a set of conditional independences, usually referred to as *local Markov property* (e.g. [40]): each variable is conditionally independent of its non-descendants given its parents. For our example DAG in Fig. 1 the lack of the edge (X_1, X_4) is associated with the independence $X_4 \perp\!\!\!\perp X_1 | X_2, X_3$, while the missing (X_2, X_3) represents $X_3 \perp\!\!\!\perp X_2 | X_1$. However, the conditional independences from the local Markov property are not the only ones associated with a BN. The DAG can be investigated to check if generic conditional independences hold in the model via the so-called *d-separation* criterion (see e.g. [5] for details).

To better illustrate BNs, consider the following simplified scenario of the much more complex application on railway travel satisfaction we develop in Section 4.2. We are interested in assessing how the length of national railways, average national income, and the country in which the travel took place affect the satisfaction of railway travelers. For simplicity, we grouped European countries into four regions: southern Europe (SE), western Europe (WE), eastern Europe (EE), and northern Europe (NE). We make the assumption that conditionally on the length of the railway and the national income, knowing the region of the traveler is irrelevant to predicting satisfaction. Furthermore, we assume that conditionally on the region, the length of the railway does not provide any information to predict the average income. This situation can be depicted by the DAG in Fig. 1 with X_1 = Country, X_2 = Length, X_3 = Income, and X_4 = Satisfaction. The definition of the BN is completed by the specification of the conditional probabilities of the model. In the categorical case, these are most commonly referred to as *conditional probability tables (CPTs)*. These tables store the conditional probability of a variable X_i for every possible combination of its parents X_{Π_i}. The CPTs for our railway travel satisfaction example are reported in Table 1, assuming Length and Income are categorized into Low and High, while Satisfaction into Low, Medium, and High. Notice that the CPTs automatically embed the conditional independences of the model: for instance, the CPT of Satisfaction states that, conditionally on each combination of Length and Income, its probability distribution is the same for every European region. The probability of any event can then be computed from the CPTs: the probability of a southern European traveler from a high-income country with a low railway track length being highly satisfied can be computed using Eq. (1) as $0.15 \cdot 0.5 \cdot 0.6 \cdot 0.2 = 0.009$.

As apparent from the previous discussion, the definition of a BN model consists of two steps: the definition of a DAG G establishing the dependence structure between the variables; and the CPTs storing the conditional probabilities of variables given their parents in G. Although these two steps can be performed via expert elicitation [43–46], our

Table 1
CPTs associated to the BN in Fig. 1.

Country									
SE			WE			EE			NE
0.15			0.25			0.35			0.25
Country	Length		Country	Income		Length	Income	Satisfaction	
	Low	High		Low	High			Low	Medium	High
SE	0.6	0.4	SE	0.5	0.5	High	High	0.1	0.4	0.5
WE	0.3	0.7	WE	0.2	0.8	High	Low	0.3	0.4	0.3
EE	0.6	0.4	EE	0.7	0.3	Low	High	0.5	0.3	0.2
NE	0.3	0.7	NE	0.2	0.8	Low	Low	0.5	0.3	0.2

focus here is in the case both are learned from observational data using machine learning algorithms. Data-driven algorithms to learn the DAG G are usually referred to as *structural learning algorithms* (see [47–49] for an overview). Three classes of algorithms are common: *constraint-based* algorithms (e.g. the Peter-Clark (PC) algorithm [50]), which use conditional independence tests; *score-based* algorithms (e.g. the tabu algorithm [51]), which use goodness-of-fit scores as objective functions to maximize; and *hybrid* algorithms (e.g. the max–min hill climbing (MMHC) algorithm [52]) that combine both approaches. Given a DAG G, maximum likelihood or Bayesian approaches can be used to learn the CPT tables.

BNs provide an efficient platform to answer *inferential queries*, meaning computing (conditional) probabilities of interest from the model. Although this is, in general, an NP-hard problem, algorithms that take advantage of the underlying DAG structure have been defined (e.g. [40]). In applied analyses, these types of query are usually called *what-if analysis*, which are used to identify the most important factors affecting an output of interest.

2.2. Non-symmetric conditional independence

A BN model can only formally encode symmetric types of conditional independence corresponding to equalities between probabilities as in Eq. (3). Those associated with the local Markov condition are by construction imposed by the CPTs of the BN since probabilities are defined conditionally on the parent variables only. However, often, the CPTs of a BN include additional equalities between its entries, which cannot be inferred by simply looking at the underlying DAG.

As a first example, consider the CPT for Satisfaction in Table 1. It can be seen that the probability distribution of Satisfaction is the same in the cases Length = Low, Income = High and Length = Low, Income = Low. In other words, conditionally, on a low railway length, income is irrelevant to predicting satisfaction. So this is a conditional independence that holds for only a specific value of the conditioning variable, Length = Low, and not for the other (Length = High). This is usually referred to as a *context-specific* independence [10]. More formally, we say that X_i is context-specific independent of X_j in the context x_k if

$$p(x_i|x_j, x_k) = p(x_i|x_k) \tag{4}$$

for all possible values x_i, x_j. Notice that for another value of the variable X_k, say x'_k, we would have that $p(x_i|x_j, x'_k) \neq p(x_i|x'_k)$.

The CPTs in Table 1 include further equalities between their rows. For instance, the probability distribution of Length is the same for Country = SE,EE, as well as for Country = WE,NE. This means e.g. that the probability of Length = High is the same in Northern and Western Europe. Similarly, there are equalities in the CPT of Income. These types of equalities have been referred to as *partial conditional independence* [11]. More formally, we say that X_i is partially conditionally independent of X_j in the domain $\{x_j^1, \dots, x_j^l\}$ given context x_k if

$$P(x_i|x_j^a, x_k) = P(x_i|x_j^b, x_k), \tag{5}$$

Table 2
Example of a CPT including a local independence.

Length	Income	Satisfaction		
		Low	Medium	High
High	High	0.5	0.3	0.2
High	Low	0.3	0.4	0.3
Low	High	0.7	0.2	0.1
Low	Low	0.5	0.3	0.2

for all values x_i and every pair x_j^a, x_j^b in the domain $\{x_j^1, \dots, x_j^l\}$. Notice that in our example there was no conditioning context x_k: this case is sometimes called marginal partial independence. If the domain $\{x_j^1, \dots, x_j^l\}$ includes all possible values x_j, then Eqs. (4) and (5) coincide. Notice that variables must take more than two values for a non-trivial partial conditional independence to hold.

A final type of independence is the so-called *local independence* [12], which simply states that some conditional probability distributions are identical but no discernable patterns as in Eqs. (4) and (5) can be detected. Let X_A be a vector not including X_i and x_A and x'_A two instantiations of X_A. A local independence is an equality of the form

$$P(x_i|x_A) = P(x_i|x'_A), \tag{6}$$

for all values x_i. An illustration of local independence is given in Table 2. The conditional probability distribution of Satisfaction is the same for the case Length = Low, Income = Low and Length = High, Income = High.

These more generic constraints between probabilities are usually referred to as *non-symmetric conditional independence*. As noticed, BNs are not able to graphically visualize this additional information often included in their CPTs. For this reason, extensions of BNs that can graphically visualize non-symmetric patterns of dependence have been proposed (e.g. [6–9]). In this paper, we focus on one specific extension of BNs called *staged tree model*.

2.3. Staged trees

Differently to BNs, whose graphical representation is a DAG, staged trees [14,15] are PGMs that visualize conditional independence by means of a colored tree. A *probability tree* is associated with the vector of interest $X = (X_1, \dots, X_p)$, denoting its sample space and probabilities. A probability tree is a rooted directed tree where each edge reports a (conditional) probability. The sum of the probabilities of edges emanating from the same non-leaf vertex must sum up to one and the product of the probabilities of edges along a root-to-leaf path is equal to the probability of the associated atomic event.

As an illustration, consider the tree in Fig. 2 (the coloring of the vertices is, for now, irrelevant) representing the sample space of our running example in railway service satisfaction. Recall that the variables of interest are Country, Length, Income, and Satisfaction; and suppose we fix this ordering of the variables. The root of the tree v_0 is associated with the first variable Country, and the edges emanating from v_0 represent the possible values Country can take,

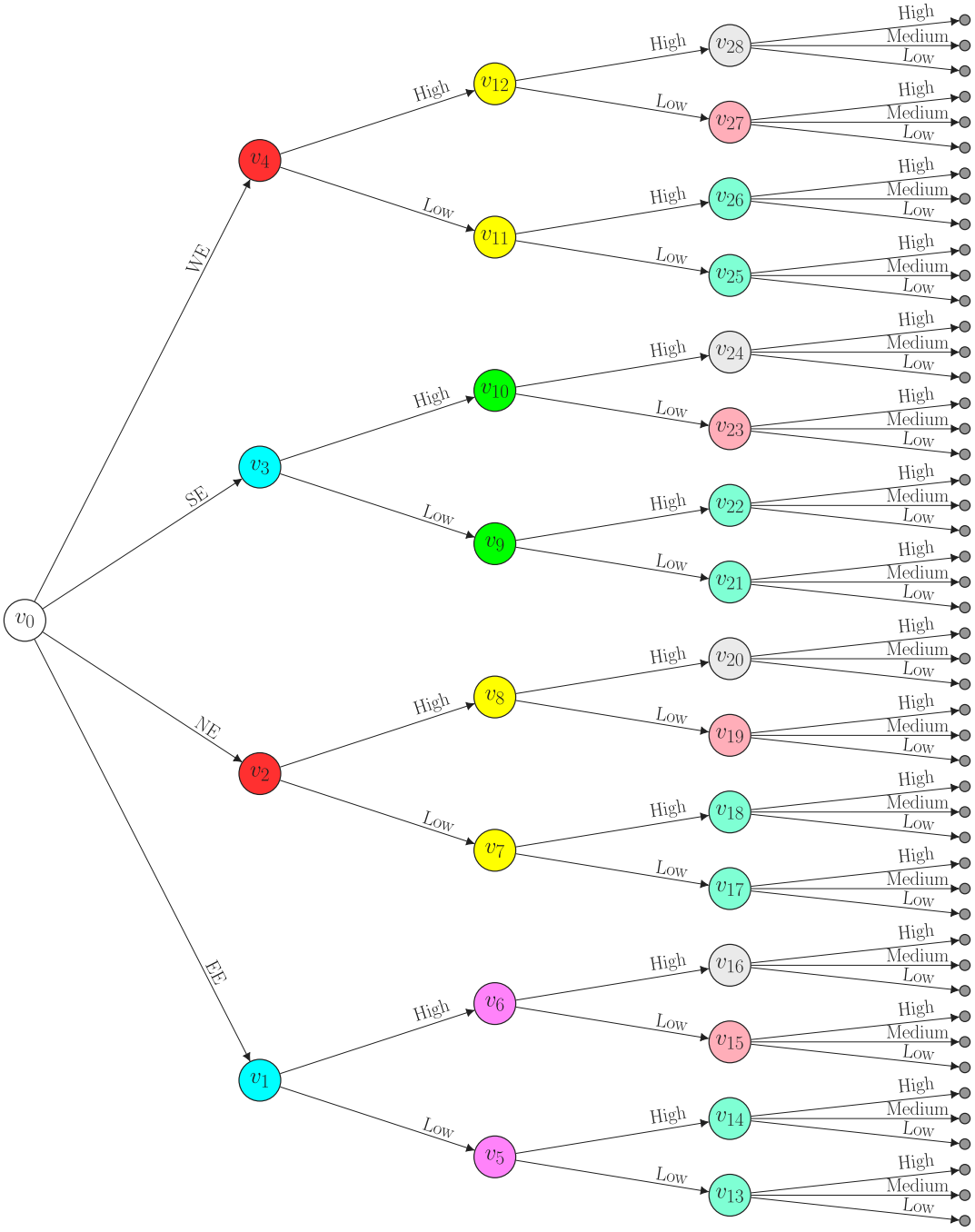

Fig. 2. Staged tree for the railway service satisfaction example whose staging represents all probabilities equalities in Fig. 1 and Table 1.

namely EE, NE, SE, WE. The probability associated with these four edges (not reported here) must sum up to one. Now consider the vertex v_1. The edges emanating from v_1 denote the conditional probability of Length given Country = EE, and again, their probabilities must add up to one. All non-leaf vertices at the same depth in the tree are associated with conditional distributions from the same variable. Because of the standard chain rule of probabilities, the product of the edge probabilities in a root-to-leaf path is the probability of an atomic event. Consider the lowest path: this represents the event Country = EE, Length = Low, Income = Low, Satisfaction = Low.

Conditional independences are visualized in the tree by a coloring of the non-leaf vertices. Two vertices that are assigned the same color are such that their edge probabilities are the same. Furthermore, two equally-colored vertices are said to be in the same *stage*. Considering again Fig. 2, it can be seen that v_1 and v_3 are given the same color. This means that $P(\text{Length} = \text{High}|\text{Country} = \text{EE}) = P(\text{Length} = \text{High}|\text{Country} = \text{SE})$ (and similarly for Length = Low): a marginal partial independence. Considering vertices at depth two, it can be noticed that the stages are $\{v_5, v_6\}, \{v_7, v_8\}, \{v_9, v_{10}\}$, and $\{v_{11}, v_{12}\}$. This implies, for instance (vertices $\{v_5, v_6\}$), that the conditional distribution of Income given Country = EE and Length = Low is the same as the one given Country = EE and Length = High. However, it can be seen that this happens for every value of Country (EE, NE, SE, WE). Altogether, this means that Income and Length are conditionally independent given Country: a symmetric, traditional conditional independence. By investigating further the coloring of the staged tree in Fig. 2 additional symmetric and non-symmetric independences can be identified.

This example clearly illustrates that staged trees can embed both symmetric and non-symmetric types of independence via their coloring. BNs can be seen as a subclass of staged trees [15,53] since all BNs can be represented as staged trees, but not vice-versa since staged trees can graphically represent non-symmetric independences. By a thorough investigation of the staged tree in Fig. 2, it can be noticed that its staging is such that the only equalities between probabilities it enforces are the two symmetric conditional independences from the DAG in Fig. 1 and the constraints in the CPTs in Table 1. Therefore, the staged tree in Fig. 2 gives a complete graphical representation of the BN, including the equalities enforced in its CPTs.

Although staged trees could be expert-elicited, our focus here is on data-driven structural learning algorithms. In the context of staged trees, these require three steps: (i) learning an optimal ordering of the variables; (ii) learning the staging of the non-leaf vertices; (iii) learning the edge probabilities. Step (iii) is similarly conducted as in BNs using either frequentist or Bayesian approaches. Step (i) is computationally challenging due to the factorial explosion of the number of orderings. For a small number of variables, the ordering can be found using a dynamic programming approach [54,55]. In more complex scenarios, a fraction of the space of orderings could be explored [56]. Step (ii) is discussed in detail in Section 3.1.

One important consideration is that structural learning of generic staged trees is hard, due to the explosion of the model search space as the number of variables increases (see e.g. [57]). For this reason, recent research has focused on sub-classes of staged tree models: Carli et al. [58] defined naive staged trees that have the same number of parameters of a naive BN over the same variables; Leonelli and Varando [59] considered simple staged trees which have a constrained type of partitioning of the vertices; Duarte and Solus [57] defined CStrees which only embed symmetric and context-specific types of independence; Leonelli and Varando [60] introduced *k-parents staged trees*, which limit the number of variables that can have a direct influence on another.

A wide array of methods to efficiently investigate real-world applications are now available for staged trees, including user-friendly software [16,17], inferential and sensitivity routines [61–63], dealing with missing data [64], causal reasoning [65] and identification of equivalence classes [66], to name a few. Such techniques are, in general, not available for other graphical models embedding non-symmetric independences, thus making staged trees a viable as well as efficient option for applied analyses.

2.4. Asymmetry-labeled DAGs

An additional challenge in modeling categorical data with staged trees is that the sample space, and therefore the size of the tree to be plotted, grows super-exponentially with the number of variables. In our simple example with one quaternary, one ternary, and two binary variables, the tree becomes already relatively big. In practice, it becomes almost impossible to visualize it and, therefore, graphically assess the learned dependence structure with more than seven binary variables.

To address this visualization challenge, Varando et al. [53] introduced a compression of a staged tree into a DAG, called *asymmetry-labeled DAG (ALDAG)* having the following properties: (i) $X_i \perp\!\!\!\perp X_j | X_k$ is implied by the staging of the tree if and if only X_i and X_j are d-separated by X_k in the ALDAG; (ii) the ALDAG is minimal, in the sense that there are no other DAGs with a smaller number of edges respecting property (i); (iii) the edges of the ALDAG are labeled/colored to denote the type of non-symmetric independence existing between the relevant variables.

Another interpretation of ALDAGs is as an embellishment of a standard BN, letting edges denote the existence of additional probability equalities in its CPTs. To illustrate this, consider the running railway service satisfaction example with BN in Fig. 1 and CPTs in Table 1.

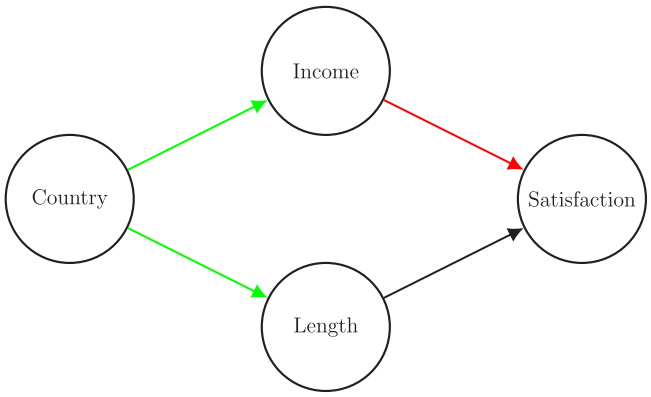

Fig. 3. ALDAG associated with the staged tree in Fig. 2, or, equivalently, with the DAG in Fig. 1 with CPTs in Table 1. Labels: black - symmetric dependence; red - context-specific; green - partial.

As already noticed, the CPTs include additional probability constraints that cannot be visualized from the DAG. The ALDAG representing this extra information is reported in Fig. 3, which, by construction, has the same edge set as the original DAG. The CPT of Length includes the partial independences $P(\text{Length} = \text{High}|\text{Country} = \text{EE}) = P(\text{Length} = \text{High}|\text{Country} = \text{SE})$ and $P(\text{Length} = \text{High}|\text{Country} = \text{WE}) = P(\text{Length} = \text{High}|\text{Country} = \text{NE})$: thus the edge is labeled partial. A similar observation can be made for the CPT of Income. Concerning the CPT of Satisfaction, we already noticed that the probability distribution of Satisfaction does not depend on Income when Length = Low. For this reason, the edge from Income to Satisfaction is given the label context-specific. On the other hand, if the value of Income is kept fixed to either High or Low, changing Length from Low to High has an effect on the probability of Satisfaction. For this reason, the edge from Length to Satisfaction is given a symmetric label since this is the type of relationship that graphically can be described by the DAG and symmetric conditional independence.

The ALDAG in Fig. 3 would also be obtained by applying the compression algorithm of Varando et al. [53] to the staged tree in Fig. 2, since it specifically embeds the probability equalities defined by the DAG and CPTs considered before. The conversion of a staged tree into its associated ALDAG has the advantage that the many routines available for inference over BNs, for instance, the already-mentioned fast inferential algorithms and what-if analyses, can be straightforwardly used for staged trees as well. In our applications below, we showcase the use of what-if analyses over the associated ALDAG of a learned staged tree.

Leonelli and Varando [56] noticed that for staged trees learned from observational data, their associated ALDAGs are usually complete unless some sparsity is imposed during the structural learning algorithm. The interpretational advantages of compressing the tree would basically be lost with a complete ALDAG. For this reason, Leonelli and Varando [60] introduced k-parents staged trees that are staged trees such that the maximum in-degree in the associated ALDAG is k. Restricting the number of parents makes sense from an applied point of view since, most often, only a limited number of variables can be expected to have a direct influence on another. The option of setting a maximum number of parents is also available in the standard `bnlearn` software [67]. Furthermore, by considering k-parents staged trees for a small k, the model search space is drastically reduced, and computationally efficient structural learning algorithms are available.

Limiting the number of parents has the advantage that, by applying the local Markov condition over the ALDAG, irrelevant variables can be removed from the visualization of the staging of a variable of interest. Varando et al. [53] introduced the so-called *dependence subtree*, which, for each variable, shows its staging using only the relevant

parent variables. By visualizing the ALDAG and the dependence subtrees, the whole, complex, non-symmetric dependence structure can be investigated even in situations with a large number of variables.

3. Robust learning algorithms for staged trees

It is now common practice in applied structural learning of BNs to assess the strength of the learned relationships, i.e. *validate* the learned network. The most common approach is to resample with replacement the data via non-parametric bootstrap and to estimate a BN for each of the resampled datasets [18,19]. Edges that appear a frequency of times above some threshold are then retained in the final DAG [68]. Although there are now theoretical methods to choose such a threshold [20], most often it is manually fixed at some pre-specified level (e.g. [69]). Furthermore, the validity of the model is often assessed by evaluating out-of-sample predictions using a k-fold cross-validation [31]. The above-mentioned resampling strategy is then applied within each iteration of the cross-validation. Both resampling and cross-validation can be easily implemented in the learning of BNs using the `bn.boot` and `bn.cv` functions of the `bnlearn` R package.

Despite their widespread use for learning BNs, such techniques have not been applied for learning staged trees from data. The quality of a staged tree model is most often evaluated over the training dataset without an out-of-sample assessment of the validity of the model. Before discussing how this could be implemented in practice, we review standard structural learning algorithms for staged trees.

3.1. Structural learning for staged trees

In Section 2.3 above we started discussing the learning of staged trees from data. The first step is choosing an ordering of the variables in the tree. For a small number of variables, all possible orderings can be evaluated by taking advantage of a dynamic programming approach (implemented in the `search_best` function in the `stagedtrees` R package) [54,55]. Otherwise, the space of possible orderings can be initially pruned, for instance by only selecting orders compatible with a learned BN, so that the optimal ordering selection can also be performed when more variables are studied [56]. However, most often, the order is selected using common knowledge, expert opinion, and, if available, the natural causal ordering of the variables.

Once an ordering has been selected, an optimal staging of the vertices has to be learned. By construction, only vertices at the same depth of the tree, i.e. vertices representing the conditional distributions of the same variable, can be merged in the same stage. Practically, this comes down to finding an optimal clustering of the vertices by exploring the space of vertices' partitions. Although distance-based [58] and k-means [70] methods have been proposed, staged trees are most commonly learned using greedy hill-climbing techniques which optimize a model score (most often the BIC as discussed in [71]). In its most general form (implemented in the `stages_hc` function of the `stagedtrees` R package) every possible merging and splitting of stages is considered at each iteration.

The most common greedy search routine is, on the other hand, the *backward-hill climbing* (or agglomerative hierarchical clustering) algorithm [72] which, starting from the tree where every vertex is in its own stage, at each iteration merges the pair of stages leading to the best score increase until no improvement is found. This routine is implemented in the `stages_bhc` function of the `stagedtrees` R package and is henceforth referred to as BHC. Notice that the learning of the staging can be performed independently and in parallel for vertices at different depths in the tree because the likelihood (and hence most model's scores as BIC) separates across the variables.

Because of the size of the data applications we study in Section 4, we will also use the techniques to learn k-parents staged trees introduced in [56] and reported in the Appendix. For each variable, these first select k parents that maximize the conditional mutual information and then run the BHC to select an optimal staging. Notice that the resulting tree is guaranteed to be k-parent since no additional parents can be added during the learning of the staging.

3.2. Robust learning of a variables' ordering

We now describe how non-parametric bootstrap resampling techniques can be used in the context of learning staged trees from data, starting by deciding on a variables' ordering. Suppose a dataset $\mathcal{D}$ of N observations of the variables of interest $X_1, \dots, X_p$ is available. We construct M synthetic versions of $\mathcal{D}$ each of size N, called $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(M)}$, using non-parametric bootstrap.

Suppose we chose a specific learning algorithm for the staging of the tree (e.g. BHC). For $i = 1, \dots, M$, we apply the dynamic programming approach using $\mathcal{D}^{(i)}$ to learn an optimal ordering of the variables $X_\sigma^{(i)}$. This gives us M variables' orderings.

For every pair (r, s), $r, s = 1, \dots, p$, we compute the frequency that X_r preceded X_s in the order, i.e.

$$N_{rs} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{X_{\sigma(r)}^{(i)} \prec X_{\sigma(s)}^{(i)}}$$

The final ordering of the variables X_σ is then uniquely defined by the $N_{rs} \geq 0.5$.[1]

Of course, this routine has a factorial complexity and can be implemented only for a small number of variables (see [55,56] for a discussion of the time required to learn a variable ordering). However, we showcase in Section 4.2 that the order selection can also be performed in complex scenarios by using expert information to group the variables.

3.3. Robust learning with a fixed ordering

A non-parametric bootstrap is then used to estimate the staging of the tree. Again, assume that a staging learning algorithm has been chosen, $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(M)}$ synthetic copies of $\mathcal{D}$ have been generated, and that an optimal ordering X_σ has been fixed. For each $\mathcal{D}^{(i)}$, we estimate and optimal staged tree $\mathcal{T}^{(i)}$ with an optimal variable ordering X_σ. Call $U_{\sigma(j)}^{(i)}$ the staging of the vertices at depth j in the ith bootstrap replicate, $j = 1, \dots, p-1$, $i = 1, \dots, M$.

Given the M learned stagings at a chosen depth of the tree j, $U_{\sigma(j)}^{(i)}$, $i = 1, \dots, M$, a method to combine them into a unique one must be adopted. This comes down to finding an optimal way to summarize M different partitions of the same set, a problem that has been addressed in Bayesian clustering (e.g. [73]). In this work, we apply a novel methodology that resembles the one to choose a final BN after bootstrapping:

- Create a matrix Z_j whose columns are $U_{\sigma(j)}^{(1)}, \dots, U_{\sigma(j)}^{(M)}$;
- Compute the pairwise dissimilarity matrix D_j, giving for each pair of elements the frequency of times they are not in the same subset (the `psm` function from the `salso` R package is used [74]);
- A standard agglomerative hierarchical clustering is run over D_j using the `hclust` R function;
- The associated dendrogram is cut at a pre-specified height: we draw a horizontal line across the dendrogram at the chosen height and the number of vertical lines this horizontal line intersects with corresponds to the number of stages. Each intersection represents a cluster so that all vertices below that intersection point belong to the same cluster, giving the final staging $U_{\sigma(j)}^{*}$.

The averaged staged tree model is then $\mathcal{T}^*$ with staging $U_{\sigma(j)}^{*}$, $j = 1, \dots, p-1$. Notice that if the above method is used in conjunction with algorithms to learn k-parents staged trees, the resulting tree $\mathcal{T}^*$ is not necessarily within the same model class, and its associated ALDAG can have a maximum in-degree larger than k. However, as illustrated by

[1] In the rare event of ties when $N_{rs} = 0.5$ the ordering of the associated variables can be randomly chosen.

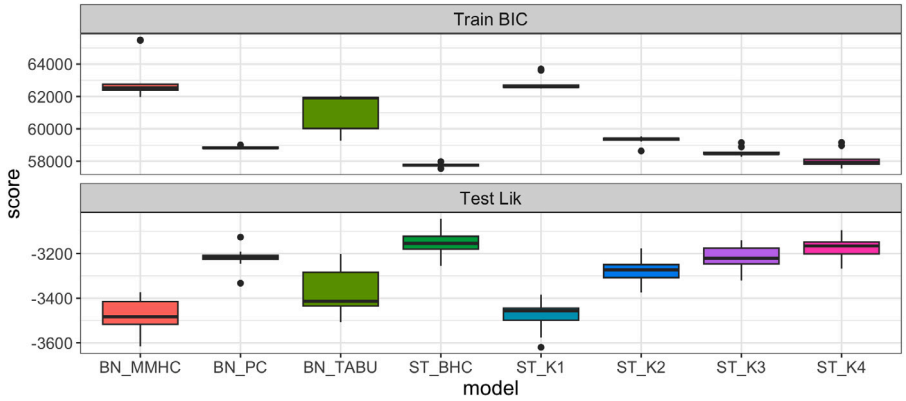

Fig. 4. Boxplots of BIC scores (train dataset) and log-likelihood (test dataset) from a 10-fold cross-validation over the airline dataset.

Table 3
Ordering of the variables for different staged tree learning algorithms using the bootstrap approach over the airline dataset.

Algorithm	BHC	1-parent	2-parents	3-parents	4-parents
1st	crew	crew	crew	crew	crew
2nd	cabin	cabin	cabin	cabin	cabin
3rd	meal	meal	meal	meal	meal
4th	booking	departure	departure	departure	departure
5th	departure	check-in	booking	check-in	booking
6th	check-in	booking	check-in	booking	check-in

our applications below, the resulting staged tree and ALDAG are still sparse.

As our routine computes M different staged trees, they can each be converted into their ALDAG representation $G^{(i)}$. As in BNs, these can be used to assess the strength of the relationship between every pair of variables. Furthermore, as they are ALDAGs, the strength of symmetric and non-symmetric relationships can be assessed by computing the frequency of each possible type of edge.

The complexity of learning staged trees given a variables' ordering has been extensively studies and investigated in simulation studies (e.g. [56]). The bootstrap approach presented here requires learning a staged tree M times and thus the complexity of our routines increases linearly with the number of bootstrap replications. Thanks to already implemented functions in R, the agglomerative hierarchical clustering step and the dendrogram construction are performed almost instantaneously and do not add any significant computational time. A simulation study, reported in Appendix A, illustrates the time required by our routines and their quality in model estimation.

4. Applications

We now showcase the use of our newly defined methods to analyze customer satisfaction surveys. BNs have been used extensively for this task (e.g. [75,76]). Here, we demonstrate the additional insights staged trees and ALDAGs can provide to untangle complex dependence patterns between the factors that affect customer satisfaction.

4.1. Airline passengers' satisfaction

We first analyze a simpler dataset considered in [27] about the satisfaction of airlines' passengers. The questionnaire contains questions on the passengers' satisfaction with their overall experience and six specific service dimensions (departure, booking, check-in, cabin environment, cabin crew, and meal). The evaluation of each item is based on a four-point scale (from extremely dissatisfied to extremely satisfied), which, for simplicity, is merged into two levels (low/high). A total of 9720 responses are available. The aim of the analysis is to evaluate the importance of these six service dimensions on the overall experience. PGMs provide an intuitive platform for this type of analysis since these service dimensions cannot be assumed independent.

We consider five different structural learning algorithms for staged trees: the BHC and the k-parent staged trees learning algorithm for $k = 1, 2, 3, 4$. We start by choosing the best ordering for each algorithm among the six specific service dimensions, while the overall experience is fixed as the last variable in the ordering since the aim is to understand the effect the dimensions have on it. Table 3 reports the learned order for each of the used algorithms using the procedure outlined in Section 3.2. It can be seen that the first three variables are equally ordered for all methods, while the last three may have a different ordering depending on the algorithm used.

Having selected the orderings, we then run a 10-fold cross-validation for the five staged tree algorithms of above and three BN structural learning algorithms for BNs: MMHC, PC and tabu implemented in the `bnlearn` R package. For each of the 10 folds, we run a non-parametric bootstrap of 200 iterations and the thresholds for both BNs and staged trees were set at 0.5. For each fold, we then computed the BIC of the models over the training data and the predictive log-likelihood over the test data. Fig. 4 reports the results. Staged trees learned with the BHC (ST_BHC) and the algorithm for 4-parents staged trees (ST_K4) outperform all BN approaches in both fitting and predictive tasks. Of course, since it is considered the most general class of models, ST_BHC outperforms ST_K4, but the difference appears to be marginal. Thus, the presence of asymmetric dependence is critical to fully understand the relationship between the service dimensions and the overall experience.

As common in BN applied structural learning (e.g. [31]), we then applied the non-parametric bootstrap approach over the full dataset for the ST_BHC and ST_K4 models. Fig. 5 reports the edge strengths for the ST_BHC algorithm. As already noticed, unless sparsity is imposed in the learning of staged trees, the learned ALDAGs are fully connected, as evidenced by all weights being equal to one. In parenthesis, the proportion of times edges are of symmetric, context-specific, or local types is reported (since all variables are binary no partial dependence can exist). The data clearly shows the presence of non-symmetric dependence that BNs cannot model because of their assumption of symmetric conditional independence.

Fig. 6 reports the edge strengths for ALDAGs learned from 4-parents staged trees. The edges' width and color are proportional to their strength. It can be clearly seen that now the learned ALDAGs were not necessarily fully complete. Again, context-specific and local edges are more common than symmetric ones.

The ALDAG associated with the averaged staged tree T^* learned with the algorithm for 4-parents staged trees is reported in Fig. 7. It is not fully connected; it has two missing edges (Check-in, Overall) and (Meal, Check-in), three edges with symmetric labels, six with context-specific labels, and ten with local labels. The ALDAG intuitively shows that the check-in has no effect on the overall experience conditionally

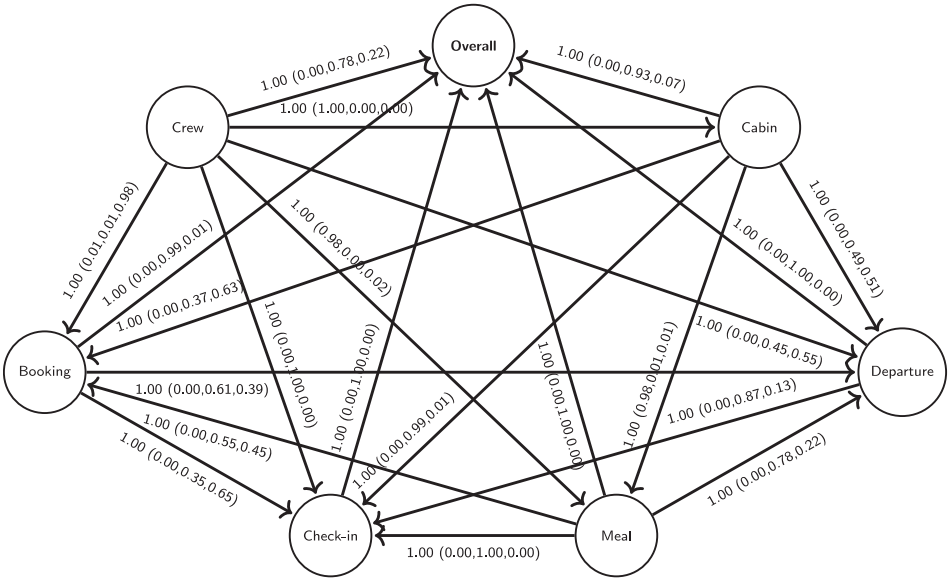

Fig. 5. Edge strengths of the ALDAGs learned with the ST_BHC algorithm over the airline dataset. In parenthesis, the proportion of times edges are given labels symmetric, context-specific, and local.

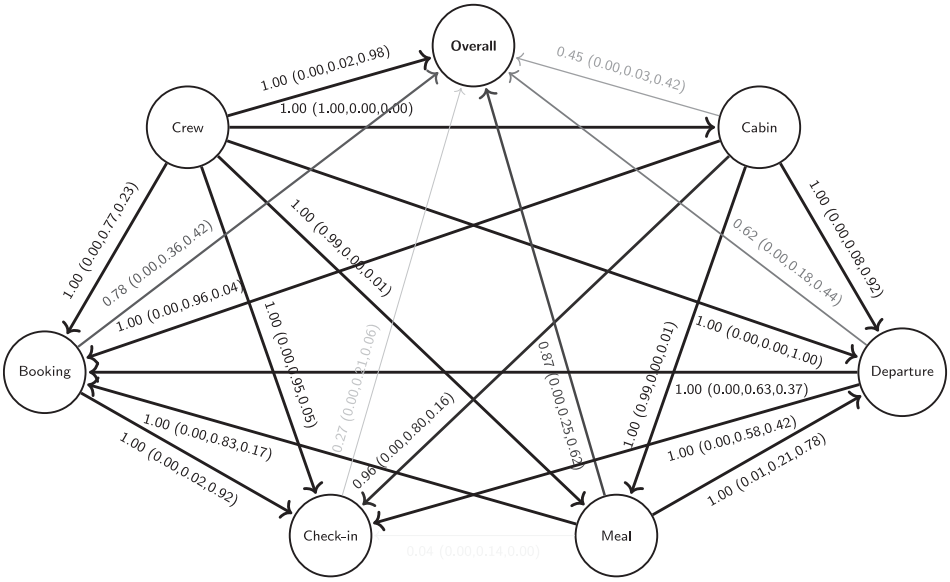

Fig. 6. Edge strengths of the ALDAGs learned with the ST_K4 algorithm over the airline dataset. In parenthesis, the proportion of times edges are given labels symmetric, context-specific, and local.

on all other service dimensions. It also shows that there are complex dependence patterns beyond symmetric ones.

Because the number of variables in the data is still limited, the averaged staged tree T^* can still be visualized and is reported in Fig. 8. It shows a complex staging embedding non-symmetric dependence patterns, which can be identified by investigating the coloring of the non-leaf vertices. A complete interpretation of this tree is beyond the scope of this illustrative example and will, therefore, not be pursued. The staging of the averaged tree was chosen using the a priori fixed threshold of 0.5. However, the bootstrap approach allows for the investigation of a user's preferred staging using the heatmap of the pairwise dissimilarity matrix. As an illustration, Fig. 9 shows the heatmap for the staging of the variable Check-in quite strongly confirming the presence of seven stages as reported in the staged tree in Fig. 8.

Through the ALDAG, we can also perform what-if analyses using the fast propagation routines available for BNs and visualize the results concisely. The model estimates the marginal probability of a passenger being satisfied (high) as 70%. As an illustration, suppose we are interested in assessing how this probability changes for passengers who are dissatisfied with the departure service. This is showcased in Fig. 10 reporting the ALDAG together with introduced evidence (gray node) and the updated marginal probabilities. The probability of a satisfied passenger dramatically decreases to around 37%. This type of what-if analysis is often said to be based on *hard evidence* [27].

As an additional illustration, the learned ALDAG states that the two service dimensions passengers are less satisfied with are the meal and the cabin (probability of 57% and 61% respectively). Consider passengers who are known to be slightly more likely to be satisfied with these two dimensions and that these probabilities are assumed to be equal to 70%. Fig. 11 reports this scenario and demonstrates that the probability of an overall satisfied passenger increased from 70% to 84%. In this case, the what-if analysis is said to be based on *soft evidence*.

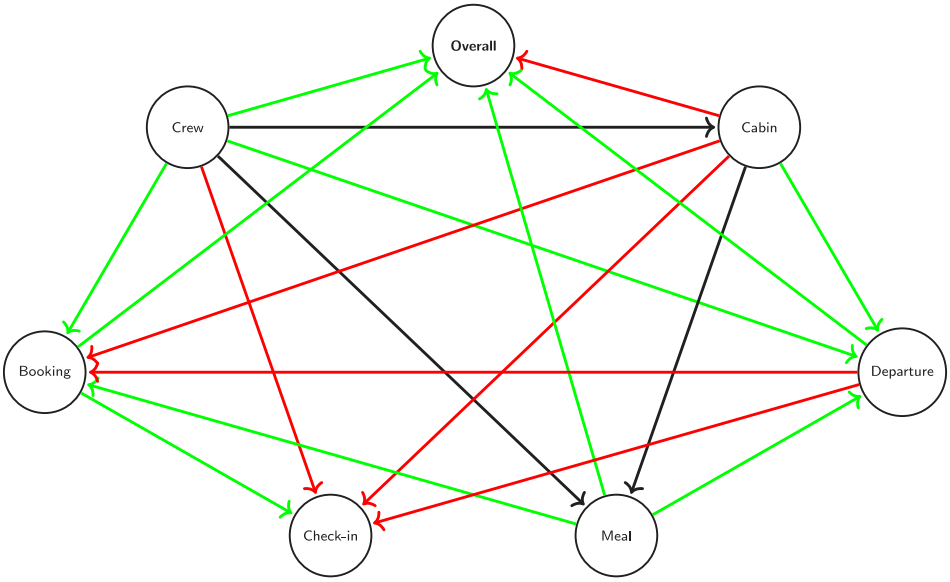

Fig. 7. Averaged ALDAG learned from 4-parents staged trees over the airline dataset.

4.2. Railway travelers' satisfaction

Our second application analyzes the satisfaction of railway travelers in the European Union. A similar analysis using BNs was developed in [37] using data collected between 2010 and 2014.

4.2.1. The data

The Flash Eurobarometer 463 [77] collected information about Europeans' satisfaction with passenger rail service at the beginning of 2018 (this is the latest survey available about rail services). The Flash Eurobarometer were launched by the European Commission in the late eighties and are "small scale" surveys conducted in all EU member states at times, occasionally reducing or enlarging the scope of countries as a function of specific topics. The overall satisfaction of around 21 thousands railway travelers across the European Union is recorded on a three-level scale (low/medium/high). Individual information about the surveyed travelers is also recorded: gender (male/female), age (dichotomized to 15–54/more than 54), education (dichotomized to university/no university), community (rural/small/large), occupation (self-employed/employee/unemployed), and country. Additional information about the location of the travelers was available for some countries at different spatial resolutions: national (e.g. Estonia and Croatia), NUTS1 (Nomenclature of territorial units for statistics) (e.g. Germany and Italy), or NUTS2 (e.g. Spain and Poland).[2] This different geographical resolution was important when integrating additional data sources as discussed next.

As in [37], we included three macroeconomic indicators, namely disposable income of households expressed in PPP (purchasing power parity), population density, and unemployment rates. Data was retrieved from the Eurostat Regional Database [78] and was available at the NUTS2 geographical resolution. Therefore, for countries whose data was available with less resolution, data aggregation and averaging were performed. These variables were included since the economic and social environment is known to have an effect on individuals' satisfaction [79]. The variables were then dichotomized using the quantile method into low/high.

We further included three variables characterizing the rail infrastructure in which the travel took place, as in [37]. The length of the railway (proportionally to the size of the region) was retrieved from the Eurostat Regional Database and measured at NUTS2 resolution. The demand for rail transport was measured by passenger-per-capita-kilometre and retrieved from the Independent Regulators' Group - Rail [80]. Rail fares were measured as passenger revenue per passenger-kilometer, expressed in euros and converted to a common currency using purchasing parity exchange rates [81]. This variable was also retrieved from the Independent Regulators' Group - Rail and measured at the national level. Geographical resolution aggregations were also carried out for these variables, which were ultimately dichotomized into low/high.

Observations with missing values were dropped from the data. Ireland and the UK were not included in the study since some of the rail infrastructure and macroeconomic variables were not available. The final dataset includes 20 995 observations and 13 variables.

Fig. 12 shows the satisfaction responses per country, highlighting quite strong differences between countries. For instance, travelers in the Baltic countries are strongly satisfied with the service. On the other hand, travelers from Bulgaria and Romania are the most dissatisfied with the service. Because of these similarities, and to simplify the analysis with staged trees, which heavily depends on the sample space size, we categorized countries into four regions (Eastern, Western, Southern, and Northern Europe)[3]

Instead of visualizing the relationship between each predictor and satisfaction via barplots as in Fig. 12, we use PGMs to integrate all data sources and study the joint effect of all predictors, as well as their interdependencies, on the travelers' satisfaction.

4.2.2. Model selection

Model selection was performed in a similar way as in Section 4.1. However, the large number of variables considered made impossible the investigation of all possible orders. For this reason, we grouped variables into three groups: demographics, rail-related, and macroeconomic. An optimal order among the variables in each group was identified using resampling. Once these were identified, an optimal order of the three groups was found using resampling again.

All algorithms considered in Section 4.1 were investigated, with the exception of BHC which would not be feasible with this larger number of variables. The results of the cross-validation are reported

[2] See https://ec.europa.eu/eurostat/web/nuts/overview for details about these geographic divisions.

[3] Using the division from https://en.wikipedia.org/wiki/Regions_of_Europe.

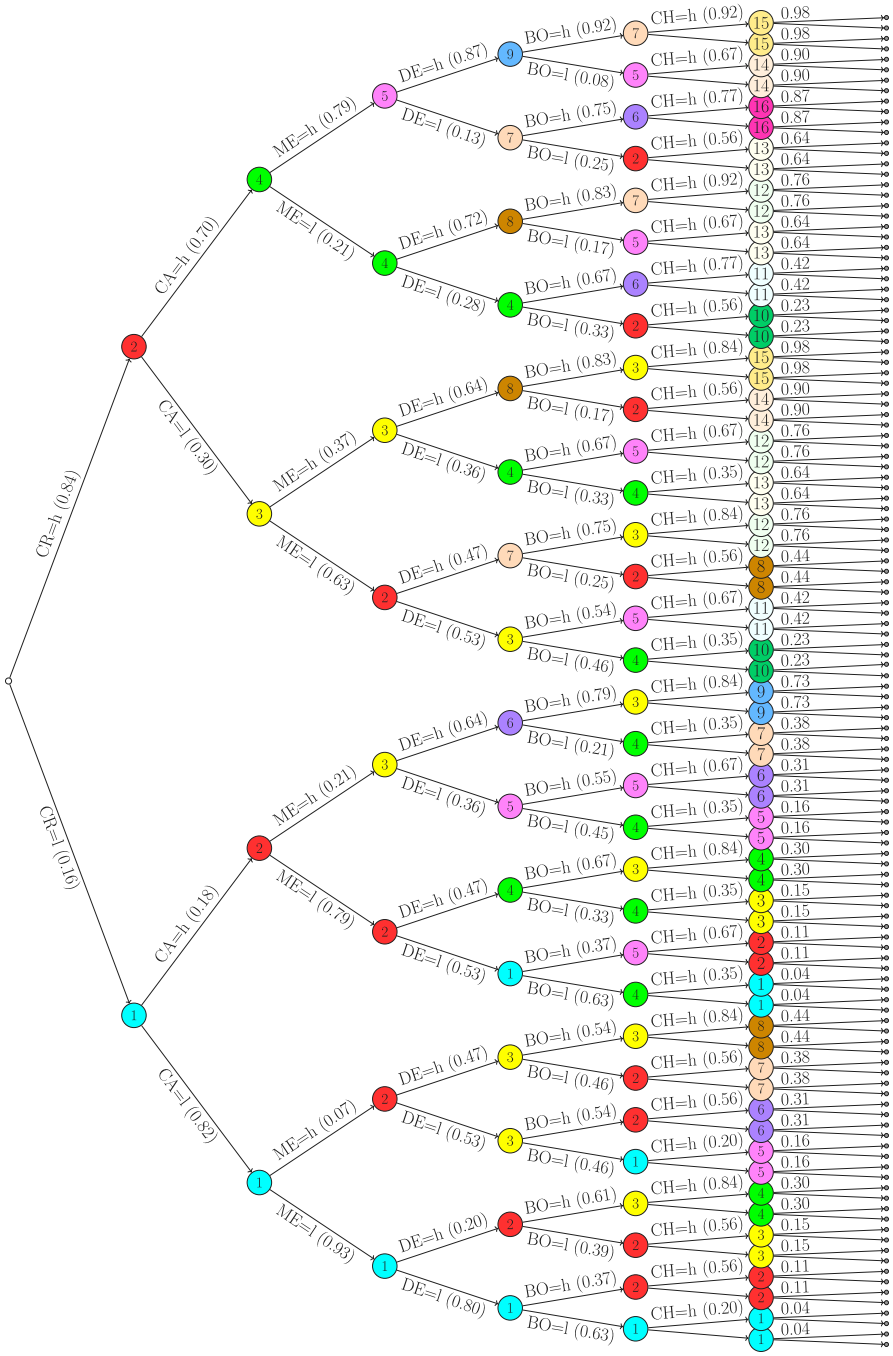

Fig. 8. Averaged staged tree using the 4-parents algorithms over the airline dataset. Though colors are repeated, only vertices at the same depth can be in the same stage.

in Fig. 13. Compared to the airline application, BNs learned with the tabu algorithm are the best scoring model. Staged trees learned with the algorithm for 4-parents staged trees are the second-best scoring approach. Including an additional parent ($k = 5$) might have provided the best scoring model, but this avenue was not pursued to avoid having an overly complex ALDAG. We have two reasons to believe this. First, the average number of parameters of the learned BNs with tabu is 324, while for staged trees it is only 232. Second, the BN learned with the tabu algorithm over the full dataset has three vertices with five parents. In our experience, it is usually the case that learned BNs have way smaller in-degrees than staged trees that comparably well fit the data. For illustrative purposes, we select the staged tree obtained with the 4-parents routine as our favorite one. Of course, in applied modeling with staged trees the choice of the preferred number of parents can depend on additional considerations. For instance, if the model is to be mostly used for predictive or classification purposes, a higher k could be chosen, for instance based on the results of a cross-validation study as in Section 4.1. On the other hand, if the model is to be used for decision support by local authorities, then a lower value of k could be preferred to enhance ease of interpretation and results' communication, at the cost of a perhaps lower predictive accuracy.

Because of the larger number of variables the averaged staged tree cannot be fully visualized: it would include more than 50 thousands root-to-leaf paths. Fig. 14 reports the ALDAG associated with the averaged staged tree. There are 45 edges, of which one has label symmetric (black), 26 have label context-specific (red), 13 have label partial (blue), and 5 have label local (green). The maximum in-degree is

Table 4
Maximum change in satisfaction probabilities computed by what-if analyses using each predictor. The color represents the direction of the change. The last column reports the mutual information between each predictor and satisfaction.

Predictor	Satisfaction			Mutual info
	Low	Medium	High	
Country (SE/EE/WE/NE)	0.046	0.160	0.202	0.015996
Length (Low/High)	0.014	0.054	0.041	0.001788
Density (Low/High)	0.006	0.054	0.048	0.001474
Income (Low/High)	0.015	0.042	0.027	0.001289
Passengers (Low/High)	0.016	0.038	0.022	0.001259
Fares (Low/High)	0.012	0.035	0.024	0.000873
Unemployment (Low/High)	0.007	0.005	0.002	0.000156
Community (rural/small/large)	0.001	0.018	0.018	0.000127
Education (no university/university)	0.004	0.010	0.014	0.000116
Occupation (not working/self-employed/employee)	0.002	0.009	0.011	0.000060
Age (15–54/more than 54)	0.002	0.006	0.004	0.000029
Gender (Male/Female)	0.001	0.001	0.000	0.000002

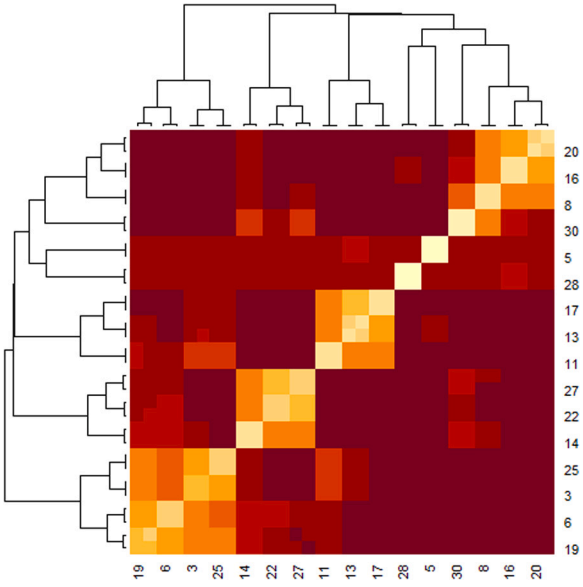

Fig. 9. Heatmap of the pairwise dissimilarity matrix of the staging of the variable Check-in.

five, so that sparsity is still retained even when using resampling techniques. The ALDAG shows a great level of interdependence between the predictors of satisfaction, which would be, in general, overlooked by "traditional" methods such as logistic regression.

4.3. Model interpretation

The ALDAG in Fig. 14 shows that railway length, country, income of households, and the demand/proportion of passengers have a direct effect on service satisfaction; all other predictors are conditionally independent of satisfaction. This confirms the conclusions of Perucca and Salini [37], who observed that country has a direct effect on satisfaction, while all other demographic information is only relatively informative.

The actual relationship between satisfaction and its parents can be visualized using the dependence subtree, reported in Fig. 15. Again, a complete interpretation of this tree is beyond the scope of this paper, but a few interesting patterns can be mentioned. The stage with the highest probability of low satisfaction is number five, corresponding to passengers coming from regions with a high demand/passenger proportion, low track length, and low income from Southern and Eastern Europe. On the other hand, passengers with the highest probability of high satisfaction come from regions of Northern Europe with low demand and length, irrespective of household income (Stage 9).

We then performed an extensive what-if analysis to assess the effect of each predictor on satisfaction. Table 4 reports the maximum absolute change in the probability that satisfaction is equal to a specific value when each predictor is fixed to any of its levels. The color describes the direction of the change: red-decrease, blue-increase, black-no uniform pattern. For instance, for the predictor Length and Satisfaction Low, the value of 0.014 in red means that the probability of Satisfaction equal to Low decreases by 0.014 when Length is changed from Low to High. It can be seen that changes are small, with the exception of the variable Country. By changing country from Southern Europe to Northern Europe the probability of a highly satisfied traveler increases by 0.202. By changing the value of Fares, Passengers, Length, Density, and Income from Low to High, the probability of Low or High Satisfaction decreases, while the probability of a Medium Satisfaction increases.

Table 4 further reports the mutual information between the output variable Satisfaction and each of the predictors computed from the model. Mutual information is a standard sensitivity measure to assess the strength of relationship in PGMs (e.g. [82]). Again, Country has clearly the strongest effect on Satisfaction, followed by Length and Density.

5. Discussion

Staged trees have proven to be a powerful PGM to describe complex patterns of dependence in tabular data. Furthermore, the associated ALDAG and dependence subtrees provide an intuitive graphical representation to visualize these complex patterns for larger applications such as the one in railways evaluation investigated here. As a PGM, staged trees are naturally suited for the integration of heterogeneous data sources, since each variable of the associated ALDAG can be informed by its own data source.

Because of the complexity and size of the model search space, it has been noticed that staged trees tend to overfit the training data and have lower performance over test sets. In this paper, we have provided a solution to this problem by introducing robust modeling approaches based on data resampling and cross-validation, which can be applied to learn both the variables' ordering in the tree and the staging of the vertices once the order is fixed. These methods were implemented using the `stagedtrees` R package and we plan to include them in the next release of the package on CRAN.

The two data analyses showcased the applied use of these routines and the insights staged trees, coupled with their ALDAG representation, can provide in practice. Sensitivity methods and what-if analyses that are standard in BNs have been used for the first time in staged tree models by taking advantage of the underlying ALDAG. We plan to also include such capabilities in the next release of `stagedtrees`.

Just as for BNs, an alternative approach for robust learning of staged trees would be to take a fully Bayesian approach and use MCMC algorithms to create a posterior sample of staged trees. Bayesian clustering methods could be almost directly applied to the Bayesian structural

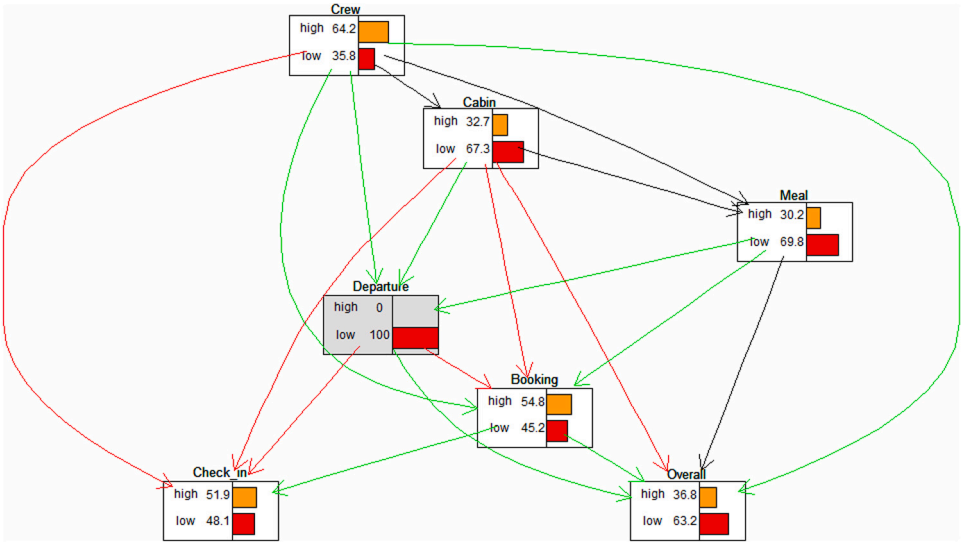

Fig. 10. What-if sensitivity analysis using hard evidence for the airline passengers' satisfaction.

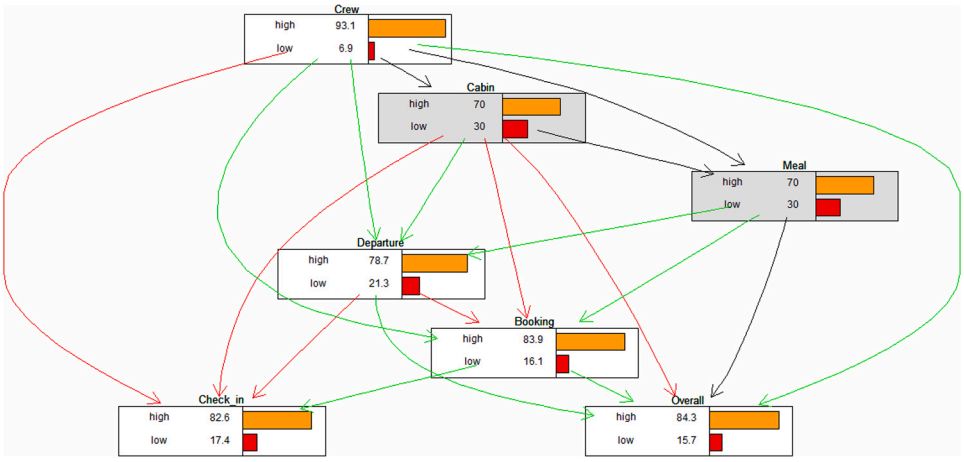

Fig. 11. What-if sensitivity analysis using soft evidence for the airline passengers' satisfaction.

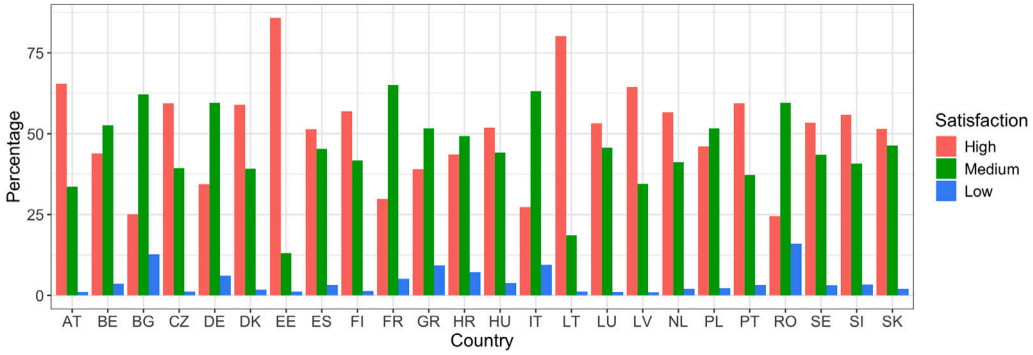

Fig. 12. Barplot of the percentage of satisfied rail passengers by European country. ISO 3166-1 alpha-2 codes are used, see https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes.

learning of staged trees, coupled with standard clustering averaging methods to identify a stage tree point estimate from the sample [73].

One drawback of the approach proposed here is that in order to impose sparsity we used algorithms to learn k-parents staged trees, but their averaged estimate does not necessarily fall within the same class of models. This issue could be avoided using the above-mentioned Bayesian approach by defining appropriate prior distributions and sampling schemes that would only explore models within the required class. The development of this approach is the focus of current research and will be reported in the near future.

An alternative approach to analyze the interdependence between the factors affecting customer satisfaction is the use of multivariate Item

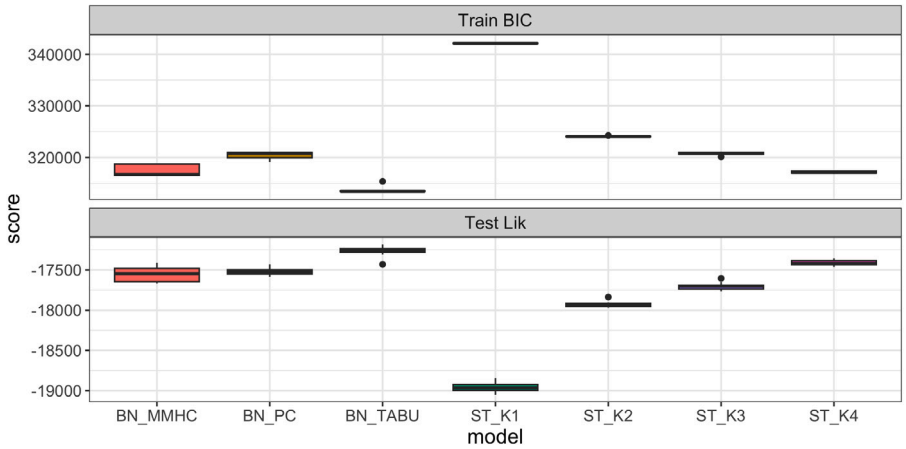

Fig. 13. Boxplots of BIC scores (train dataset) and log-likelihood (test dataset) from a 10-fold cross-validation over the railway dataset.

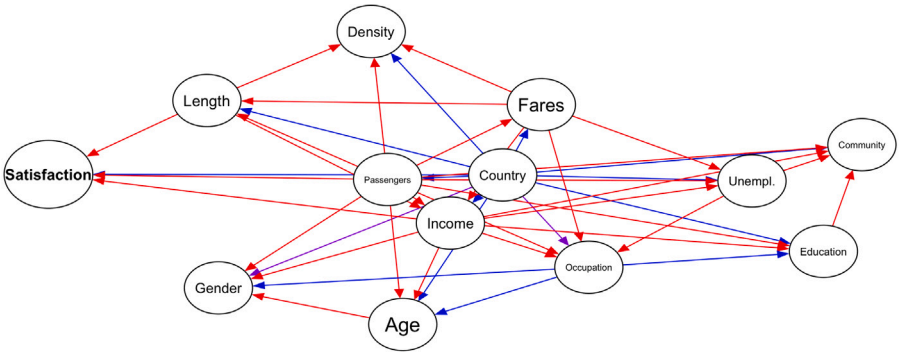

Fig. 14. ALDAG associated to the averaged staged tree learned using 4-parents staged tree algorithm over the railway dataset. Unempl. is an abbreviation for "Unemployement".

Response Theory (IRT) [83]. Furthermore, the imposition of sparsity as in k-parents staged trees could be similarly obtained by using LASSO penalizations or Spike and Slab prior distributions (e.g. [84]). A comparison of the conclusions that could be drawn from staged trees and IRT approaches was beyond the scope of this paper, but we plan to carry it out in future work.

CRediT authorship contribution statement

Manuele Leonelli: Writing – review & editing, Writing – original draft, Visualization, Formal analysis, Data curation, Conceptualization. **Gherardo Varando:** Visualization, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge the help from Prof. Silvia Salini who provided the data of the airline application in Section 4.1 and the code used to produce Figs. 10 and 11.

Appendix A. Simulation study

The performance of the proposed routines was evaluated through a simulation study. Staged trees with binary variables were randomly generated for $p = 4, 5, 6$ variables and $k = 2, 3, 4, 5$ parents in the associated ALDAG representation. Samples of sizes $N = 100, 250, 500, 750, 1000$ were generated from the tree to estimate various models, while 1000 observations were generated to assess the predicted performance of the estimated models. We replicate each combination of inputs 25 times. We consider the BHC algorithm without bootstrap (BHC), the BHC with bootstrap (BOOT), and the k-parents staged trees learning algorithm with bootstrap (BOOT_K). For each model we compute the BIC over the training data, the likelihood over the test data (TestLik), the Kullback–Leibler divergence between the true and estimated probabilities (KL), the Hamming distance between the true and estimated staged trees (Hamming), and the time required to estimate the model. The average among the 25 replications is finally computed.

Fig. A.16 reports the result of the experiment for staged trees with $p = 6$ variables. In terms of performance, it can be noticed that k-parents staged trees have overall a competitive performance with BHC algorithms (both with and without bootstrap) even when the true generating model has actually more than k parents. In terms of Hamming distance, there is small difference between the approaches, but the k-parent staged tree with the correct k tends to outperform the others. The BOOT approach is considerably slower than the others requiring around one minute, while all others are much faster and comparable to the BHC (especially in the case of $k = 1, 2, 3$).

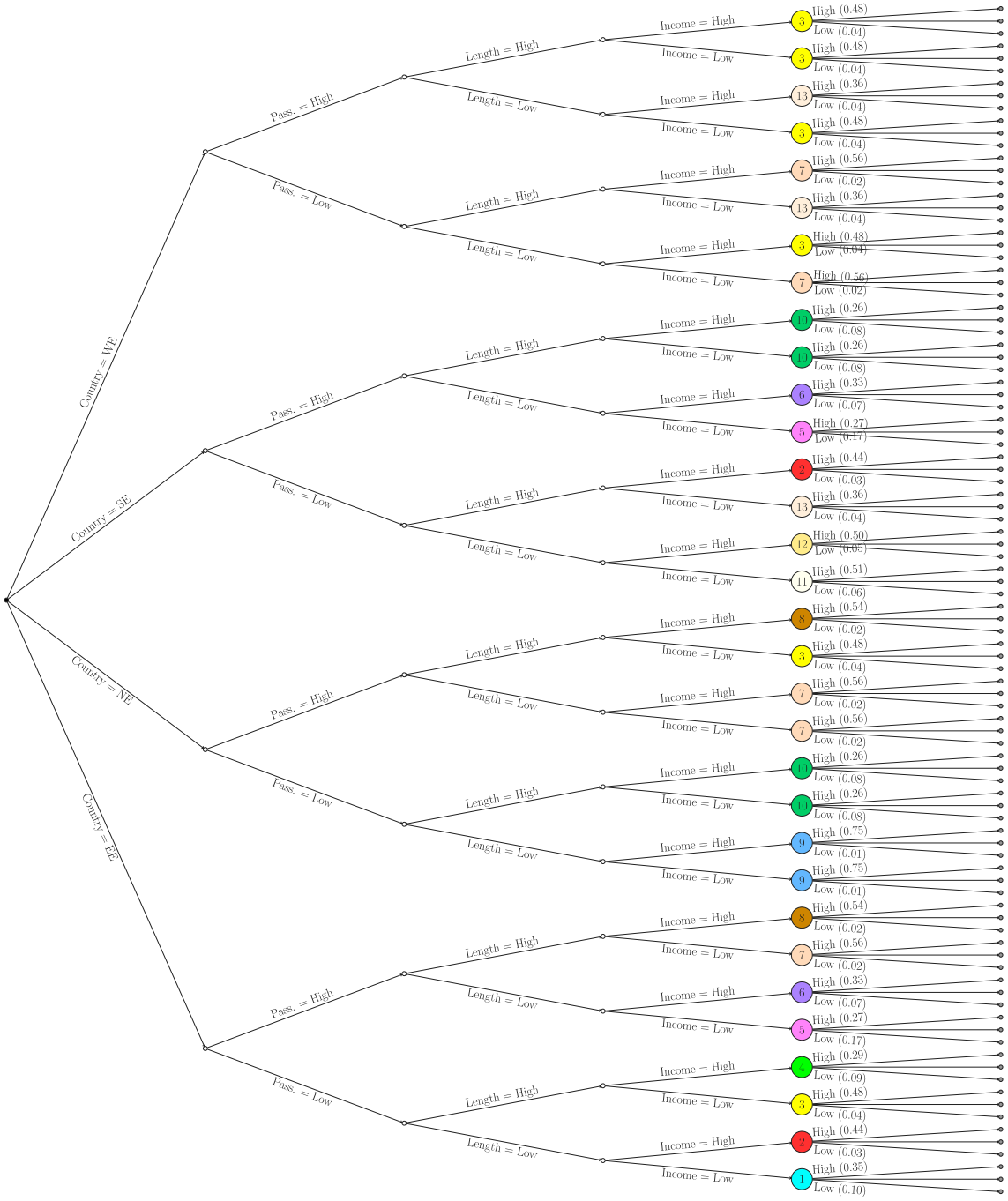

Fig. 15. Dependence subtree associated to the variable satisfaction in the ALDAG in Fig. 14.

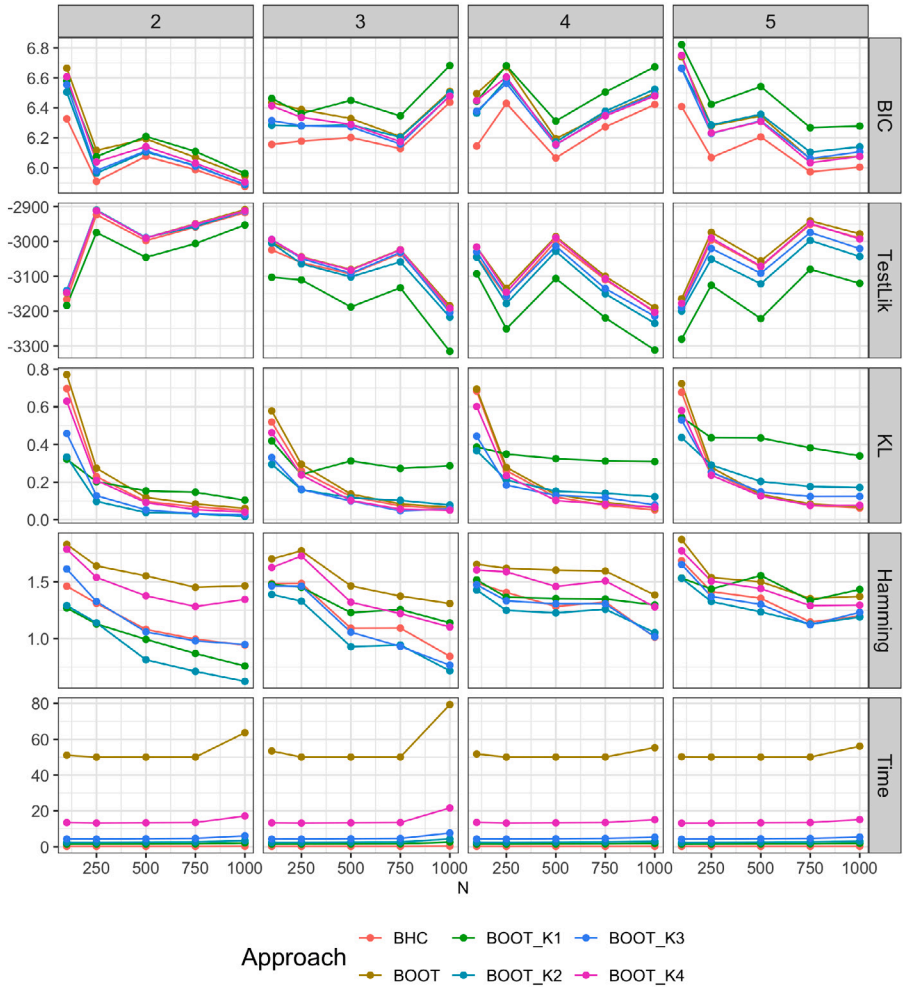

Fig. A.16. Results of the simulation study in the case of $p = 6$ variables. Rows are the number of parents in the underlying staged tree k, columns are different metrics, and the x-axis is the sample size N used to estimate the model.

Appendix B. Algorithm to learn k-parents staged trees

Algorithm 1: Learning algorithm for k-parent staged trees using CMI

Input : A dataset D over categorical variables $X_1, \dots, X_p$ and $k \in \mathbb{Z}_+$

Output: A staged tree T

for $i \leftarrow 1$ **to** p **do**
  $\Pi_i \leftarrow \emptyset$;
  if $i \leq k+1$ **then**
    $\Pi_i \leftarrow [i-1]$;
  else
    for $j \leftarrow 1$ **to** k **do**
      $max \leftarrow -\infty$;
      for $s \in [i-1] \setminus \Pi_i$ **do**
        if $I(X_i, X_j | X_{\Pi_i}) > max$ **then**
          $new \leftarrow s$;
      $\Pi_i \leftarrow \Pi_i \cup \{s\}$;

Construct G using $[p]$ and $\Pi_1, \dots, \Pi_p$;
Transform G to its equivalent staged tree T with staging $U_1, \dots, U_{p-1}$;
$score \leftarrow BIC(T)$; $T^* \leftarrow T$;
for $i \leftarrow 1$ **to** $p-1$ **do**
  $indicator \leftarrow 1$;
  while $indicator \neq 0$ **do**
    for *every pair of stages* $u_j, u_s \in U_i$ **do**
      construct T' by merging u_i and u_j;
      if $BIC(T') < BIC(T^*)$ **then**
        $score \leftarrow BIC(T')$; $T^* \leftarrow T'$;
    if $T = T^*$ **then**
      $indicator \leftarrow 0$
    else
      $T \leftarrow T^*$
return T

References

[1] Johnson S, Mengersen K. Integrated Bayesian network framework for modeling complex ecological issues. Integr Environ Assess Manage 2012;8:480–90.
[2] Leonelli M, Riccomagno E, Smith JQ. Coherent combination of probabilistic outputs for group decision making: An algebraic approach. OR Spectrum 2020;42:499–528.
[3] Marcot BG, Penman TD. Advances in Bayesian network modelling: Integration of modelling technologies. Environ Model Softw 2019;111:386–93.
[4] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann; 1988.
[5] Pearl J. Causality. Cambridge University Press; 2009.
[6] Eggeling R, Grosse I, Koivisto M. Algorithms for learning parsimonious context trees. Mach Learn 2019;108:879–911.
[7] Jaeger M, Nielsen JD, Silander T. Learning probabilistic decision graphs. Internat J Approx Reason 2006;42:84–100.
[8] Pensar J, Nyman H, Koski T, Corander J. Labeled directed acyclic graphs: A generalization of context-specific independence in directed graphical models. Data Min Knowl Discov 2015;29:503–33.
[9] Talvitie T, Eggeling R, Koivisto M. Learning Bayesian networks with local structure, mixed variables, and exact algorithms. Internat J Approx Reason 2019;115:69–95.
[10] Boutilier C, Friedman N, Goldszmidt M, Koller D. Context-specific independence in Bayesian networks. In: Proceedings of the 12th conference on uncertainty in artificial intelligence. 1996, p. 115–23.
[11] Pensar J, Nyman H, Lintusaari J, Corander J. The role of local partial independence in learning of Bayesian networks. Internat J Approx Reason 2016;69:91–105.
[12] Chickering DM, Heckerman D, Meek C. A Bayesian approach to learning Bayesian networks with local structure. In: Proceedings of the 13th conference on uncertainty in artificial intelligence. 1997, p. 80–9.
[13] Friedman N, Goldszmidt M. Learning Bayesian networks with local structure. In: Proceedings of the 12th conference on uncertainty in artificial intelligence. 1996, p. 252–62.
[14] Collazo RA, Görgen C, Smith JQ. Chain event graphs. CRC Press; 2018.
[15] Smith JQ, Anderson PE. Conditional independence and chain event graphs. Artificial Intelligence 2008;172:42–68.
[16] Carli F, Leonelli M, Riccomagno E, Varando G. The R package stagedtrees for structural learning of stratified staged trees. J Stat Softw 2022;102:1–30.
[17] Walley G, Shenvi A, Strong P, Kobalczyk K. Cegpy: Modelling with chain event graphs in Python. Knowl-Based Syst 2023;274:110615.
[18] Friedman N, Goldszmidt M, Wyner A. Data analysis with Bayesian networks: A bootstrap approach. In: Proceedings of the 15th conference on uncertainty in artificial intelligence. 1999, p. 196–205.
[19] Caravagna G, Ramazzotti D. Learning the structure of Bayesian networks via the bootstrap. Neurocomputing 2021;448:48–59.
[20] Scutari M, Nagarajan R. On identifying significant edges in graphical models of molecular networks. Artif Intell Med 2013;57:207–17.
[21] Friedman N, Koller D. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. Mach Learn 2003;50:95–125.
[22] Castelletti F, Peluso S. Equivalence class selection of categorical graphical models. Comput Statist Data Anal 2021;164:107304.
[23] Goudie RJ, Mukherjee S. A Gibbs sampler for learning DAGs. J Mach Learn Res 2016;17:1032–70.
[24] Kuipers J, Moffa G. Partition MCMC for inference on acyclic digraphs. J Amer Statist Assoc 2017;112:282–99.
[25] Kuipers J, Suter P, Moffa G. Efficient sampling and structure learning of Bayesian networks. J Comput Graph Statist 2022;31:639–50.
[26] Viinikka J, Koivisto M. Layering-MCMC for structure learning in Bayesian networks. In: Proceedings of the 36th conference on uncertainty in artificial intelligence. PMLR; 2020, p. 839–48.
[27] Cugnata F, Kenett RS, Salini S. Bayesian networks in survey data: Robustness and sensitivity issues. J Qual Technol 2016;48:253–64.
[28] Ceriani L, Gigliarano C. Multidimensional well-being: A Bayesian networks approach. Soc Indic Res 2020;152:237–63.
[29] Di Pietro L, Mugion RG, Musella F, Renzi MF, Vicard P. Monitoring an airport check-in process by using Bayesian networks. Transp Res A 2017;106:235–47.
[30] Mandhani J, Nayak JK, Parida M. Establishing service quality interrelations for Metro rail transit: Does gender really matter? Transp Res D 2021;97:102888.
[31] Liew BX, de-la Llave-Rincón AI, Scutari M, Arias-Buría JL, Cook CE, Cleland J, Fernández-de Las-Peñas C. Do short-term effects predict long-term improvements in women who receive manual therapy or surgery for carpal tunnel syndrome? A Bayesian network analysis of a randomized clinical trial. Phys Ther 2022;102:pzac015.
[32] Strong P, Smith JQ. Bayesian model averaging of chain event graphs for robust explanatory modelling. In: International conference on probabilistic graphical models. PMLR; 2022, p. 61–72.
[33] Díez-Mesa F, de Oña R, de Oña J. Bayesian networks and structural equation modelling to develop service quality models: Metro of Seville case study. Transp Res A 2018;118:1–13.
[34] Hua W, Feng X, Ding C, Ruan Z. Bayesian network modeling analyzes of perceived urban rail transfer time. Transp Lett 2021;13:514–21.
[35] Mandhani J, Nayak JK, Parida M. Interrelationships among service quality factors of Metro Rail Transit System: An integrated Bayesian networks and PLS-SEM approach. Transp Res A 2020;140:320–36.
[36] Xu X, Lu Y, Wang Y, Li J, Zhang H. Improving service quality of metro systems—A case study in the Beijing metro. IEEE Access 2020;8:12573–91.
[37] Perucca G, Salini S. Travellers' satisfaction with railway transport: A Bayesian network approach. Qual Technol Quant Manage 2014;11:71–84.
[38] Yang M, Wang Z, Cheng L, Chen E. Exploring satisfaction with air-HSR intermodal services: A Bayesian network analysis. Transp Res A 2022;156:69–89.
[39] Borgonovo E. Sensitivity analysis. Tutor Oper Res: Adv Front OR/MS: Methodol Appl 2023;52–81.
[40] Koller D, Friedman N. Probabilistic graphical models: principles and techniques. MIT Press; 2009.
[41] Glymour C, Zhang K, Spirtes P. Review of causal discovery methods based on graphical models. Front Genet 2019;10:524.
[42] Dawid AP. Conditional independence in statistical theory. J R Stat Soc Ser B 1979;41:1–15.
[43] Renooij S. Probability elicitation for belief networks: Issues to consider. Knowl Eng Rev 2001;16:255–69.
[44] Werner C, Bedford T, Cooke RM, Hanea AM, Morales-Napoles O. Expert judgement for dependence in probabilistic modelling: A systematic literature review and future research directions. European J Oper Res 2017;258:801–19.
[45] Wilkerson RL, Smith JQ. Customized structural elicitation. Expert Judgement in Risk and Decision Analysis 2021;83–113.
[46] Zhang G, Thai VV. Expert elicitation and Bayesian network modeling for shipping accidents: A literature review. Saf Sci 2016;87:53–62.
[47] Kitson NK, Constantinou AC, Guo Z, Liu Y, Chobtham K. A survey of Bayesian network structure learning. Artif Intell Rev 2023;1–94.

[48] Scanagatta M, Salmerón A, Stella F. A survey on Bayesian network structure learning from data. Prog Artif Intell 2019;8:425–39.
[49] Scutari M, Graafland CE, Gutiérrez JM. Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms. Internat J Approx Reason 2019;115:235–53.
[50] Spirtes P, Glymour C, Scheines R. Causation, prediction, and search. MIT Press; 2001.
[51] Russell S, Norvig P. Artificial intelligence: a modern approach. Prentice Hall; 2009.
[52] Tsamardinos I, Brown LE, Aliferis CF. The max–min hill-climbing Bayesian network structure learning algorithm. Mach Learn 2006;65:31–78.
[53] Varando G, Carli F, Leonelli M. Staged trees and asymmetry-labeled dags. Metrika 2024;1–28.
[54] Cowell R, Smith J. Causal discovery through MAP selection of stratified chain event graphs. Electron J Stat 2014;8:965–97.
[55] Leonelli M, Varando G. Context-specific causal discovery for categorical data using staged trees. In: International conference on artificial intelligence and statistics. 2023, p. 8871–88.
[56] Leonelli M, Varando G. Learning and interpreting asymmetry-labeled dags: a case study on covid-19 fear. Appl Intell 2024;54:1734–50.
[57] Duarte E, Solus L. Representation of context-specific causal models with observational and interventional data. 2021, arXiv:2101.09271.
[58] Carli F, Leonelli M, Varando G. A new class of generative classifiers based on staged tree models. Knowl-Based Syst 2023;110488.
[59] Leonelli M, Varando G. Structural learning of simple staged trees. Data Min Knowl Discov 2024;38:1520–44.
[60] Leonelli M, Varando G. Highly efficient structural learning of sparse staged trees. In: International conference on probabilistic graphical models. PMLR; 2022, p. 193–204.
[61] Görgen C, Leonelli M, Smith JQ. A differential approach for staged trees. In: Symbolic and quantitative approaches to reasoning with uncertainty. 2015, p. 346–55.
[62] Leonelli M. Sensitivity analysis beyond linearity. Internat J Approx Reason 2019;113:106–18.
[63] Thwaites PA, Smith JQ, Cowell RG. Propagation using chain event graphs. In: Proceedings of the 24th conference on uncertainty in artificial intelligence. 2008, p. 546–53.
[64] Barclay LM, Hutton J, Smith J. Chain event graphs for informed missingness. Bayesian Anal 2014;9:53–76.
[65] Thwaites P, Smith JQ, Riccomagno E. Causal analysis with chain event graphs. Artificial Intelligence 2010;174:889–909.
[66] Görgen C, Bigatti A, Riccomagno E, Smith JQ. Discovery of statistical equivalence classes using computer algebra. Internat J Approx Reason 2018;95:167–84.
[67] Scutari M. Learning Bayesian networks with the bnlearn R package. J Stat Softw 2010;35:1–22.
[68] Scutari M, Denis JB. Bayesian networks: with examples in r. CRC Press; 2021.
[69] Briganti G, Decety J, Scutari M, McNally RJ, Linkowski P. Using Bayesian networks to investigate psychological constructs: The case of empathy. Psychol Rep 2022;00332941221146711.
[70] Silander T, Leong TY. A dynamic programming algorithm for learning chain event graphs. In: Proceedings of the 16th international conference in discovery science. 2013, p. 201–16.
[71] Görgen C, Leonelli M, Marigliano O. The curved exponential family of a staged tree. Electron J Stat 2022;16:2607–20.
[72] Freeman G, Smith JQ. Bayesian MAP model selection of chain event graphs. J Multivariate Anal 2011;102:1152–65.
[73] Wade S. Bayesian cluster analysis. Philos Trans R Soc A 2023;381:20220149.
[74] Dahl DB, Johnson DJ, Müller P. Search algorithms and loss functions for Bayesian clustering. J Comput Graph Statist 2022;31:1189–201.
[75] Cugnata F, Kenett R, Salini S. Bayesian network applications to customer surveys and infoq. Procedia Econ Financ 2014;17:3–9.
[76] Salini S, Kenett RS. Bayesian networks of customer satisfaction survey data. J Appl Stat 2009;36:1177–89.
[77] European Commission, Brussels. Flash Eurobarometer 463 (Europeans' satisfaction with passenger rail services). GESIS Datenarchiv, Köln; 2018, http://dx.doi.org/10.4232/1.13149, ZA6933 Datenfile Version 1.0.0, [Accessed 27 November 2023].
[78] European Commission, Brussels. Eurostat regional database. 2023, https://ec.europa.eu/eurostat/databrowser/explore/all/all_themes. [Accessed 27 November 2023].
[79] Fiorio CV, Florio M, Salini S, Ferrari P. Consumers' attitudes on services of general interest in the EU: accessibility, price and quality 2000-2004. 2007.
[80] Independent Regulators' Group - Rail. Eighth annual IRG-Rail market monitoring report - Working document. 2020, https://irg-rail.eu/irg/documents/market-monitoring/260,2020.html. [Accessed 27 November 2023].
[81] Organization for Economic Co-operation and Development (OECD). PPPs and exchange rates. 2023, https://stats.oecd.org/index.aspx?DataSetCode=SNA_Table4. [Accessed 27 November 2023].
[82] Kjaerulff UB, Madsen AL. Bayesian networks and influence diagrams, Vol. 200, Springer Science+ Business Media; 2008, p. 114.
[83] Chalmers RP. Mirt: A multidimensional item response theory package for the R environment. J Stat Softw 2012;48:1–29.
[84] Chang YW, Yang CX. Bayesian inference with spike-and-slab priors for differential item functioning detection in a multiple-group irt tree model. J Stat Comput Simul 2024;94:1416–44.

Manuele Leonelli is an Assistant Professor in Statistics in the School of Science and Technology at IE University, Madrid, Spain. He received a Ph.D. in Statistics in 2015 from the University of Warwick. After spending one year as a Postdoctoral Researcher at the Federal University of Rio de Jainero, he joined the School of Mathematics and Statistics at the University of Glasgow as a Lecturer. His research is in probabilistic graphical models, sensitivity analysis and applications.

Gherardo Varando received the Ph.D. degree in artificial intelligence in 2018 from the Universidad Politétnita de Madrid (Spain). Afterward, he was a Postdoctoral Researcher at the University of Copenhagen. He is currently working as a postdoctoral researcher in the Image and Signal Processing Group at Universitat de Valencia. His research is in machine learning, statistics and causal analysis methods.