



# Staged trees and asymmetry-labeled DAGs

Gherardo Varando<sup>1</sup> · Federico Carli<sup>2</sup> · Manuele Leonelli<sup>3</sup>

Received: 3 March 2023 / Accepted: 12 February 2024 / Published online: 7 March 2024  
© The Author(s) 2024

## Abstract

Bayesian networks are a widely-used class of probabilistic graphical models capable of representing symmetric conditional independence between variables of interest using the topology of the underlying graph. For categorical variables, they can be seen as a special case of the much more general class of models called staged trees, which can represent any non-symmetric conditional independence. Here we formalize the relationship between these two models and introduce a minimal Bayesian network representation of a staged tree, which can be used to read conditional independences intuitively. A new labeled graph termed asymmetry-labeled directed acyclic graph is defined, with edges labeled to denote the type of dependence between any two random variables. We also present a novel algorithm to learn staged trees which only enforces a specific subset of non-symmetric independences. Various datasets illustrate the methodology, highlighting the need to construct models that more flexibly encode and represent non-symmetric structures.

**Keywords** Asymmetric graphical models · Bayesian networks · Context-specific independence · Staged trees · Structural learning

## 1 Introduction

Probabilistic graphical models give an intuitive and efficient representation of the relationships between random variables of interest. Bayesian networks (BNs) (see

---

✉ Gherardo Varando  
gherardo.varando@uv.es

Federico Carli  
carli@dim.unige.it

Manuele Leonelli  
manuele.leonelli@ie.edu

<sup>1</sup> Image Processing Laboratory, Universitat de València, Valencia, Spain

<sup>2</sup> Dipartimento di Matematica, Università degli Studi di Genova, Genoa, Italy

<sup>3</sup> School of Science and Technology, IE University, Madrid, Spain

e.g. Darwiche 2009) are the most commonly used graphical model and have been applied in a variety of real-world applications. One of the main limitations of BNs is that they can only represent symmetric conditional independences, which in practice can be too restrictive.

For this reason, Boutilier et al. (1996) introduced the notion of *context-specific* independence, meaning that independences hold only for specific values, or *contexts*, of the conditioning variables. Extensions of BNs encoding context-specific independences are usually defined by associating a tree representation to each vertex of the network (Cano et al. 2012; Friedman and Goldszmidt 1996; Talvitie et al. 2019), by labeling the edges (Pensar et al. 2015; Hyttinen et al. 2018), or by using some alternative approach (Chickering et al. 1997; Geiger and Heckerman 1996; Poole and Zhang 2003). In recent years there has been a growing interest in formalizing context-specific independence (Corander et al. 2019; Shen et al. 2020) and in generalizing other graphical models with non-symmetric dependencies (Nyman et al. 2016; Pensar et al. 2017).

Except for Jaeger et al. (2006) and Pensar et al. (2015), BNs embellished with context-specific independence lose their intuitiveness since all the model information cannot be succinctly represented in a unique graph. Staged trees (Smith and Anderson 2008; Collazo et al. 2018) are probabilistic graphical models that, starting from an event tree, represent non-symmetric conditional independence statements via a coloring of its vertices. Coloring has recently been found to provide a valuable embellishment to other graphical models (Højsgaard and Lauritzen 2008; Massam et al. 2018).

As demonstrated by Smith and Anderson (2008) and Duarte and Solus (2023), every BN can be represented as a staged tree. However, the class of staged tree models is much more general and can represent both symmetric and context-specific, partial and local independences (Pensar et al. 2016). Furthermore, a wide array of methods to efficiently investigate real-world applications have been introduced for staged trees, including user-friendly software (Carli et al. 2022), inferential routines (Görge et al. 2015), structural learning (Freeman and Smith 2011), dealing with missing data (Barclay et al. 2014), causal reasoning (Thwaites et al. 2010) and identification of equivalence classes (Görge et al. 2018), to name a few. Such techniques are generally not available for other graphical models embedding non-symmetric independences, thus making staged trees a viable and efficient option for applied analyses.

Our first contribution is a deeper study of the relationship between BNs and staged trees. We introduce a minimal BN representation of a staged tree that embeds all its symmetric conditional independences. Importantly, this allows us to introduce a criterion to identify all symmetric conditional independences implied by the model, which has proven to be a very challenging task (Thwaites and Smith 2015).

Reading non-symmetric independences directly from the staged tree is even more challenging. Our second contribution is a novel definition of classes of dependence among variables and the introduction of methods to identify the appropriate class from the staged tree. The presence or absence of edges in a BN encodes either (conditionally) full dependence or independence between two variables. However, the flexibility of the staged tree enables us to model and consequently identify intermediate relationships between variables, namely context-specific, partial, or local (Pensar et al. 2016).

As a result, our third contribution is the definition of a new class of directed acyclic graphs (DAGs), termed *asymmetry-labeled DAGs* (ALDAGs), by coloring

edges according to the type of relationship existing between the corresponding variables. Learning algorithms for ALDAGs, which use any structural learning algorithm for staged trees (see e.g. those included in the R package *stagedtrees*, Carli et al. 2022), are discussed below and applied to various datasets. Our fourth contribution is the definition of a new visualization of dependence, called the *dependence subtree*, which shows how a variable is related to only those directly affecting it, namely its parents in the associated ALDAG. The use of such a tool is showcased in our data applications below.

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. Duarte and Solus 2021). For this reason, recent research has focused on sub-classes of staged tree models: Carli et al. (2023) defined naive staged trees which have the same number of parameters of a naive BN over the same variables; Leonelli and Varando (2024b) considered simple staged trees which have a constrained type of partitioning of the vertices; Leonelli and Varando (2022) introduced  $k$ -parents staged trees which limit the number of variables that can have a direct influence on another; Duarte and Solus (2021) defined CStrees which only embed symmetric and context-specific types of independence. Our last contribution is the introduction of a novel algorithm, called *context-specific backward hill-climbing (CSBHC)*, to learn a new class of staged trees whose staging is restricted. In particular, the proposed algorithm learns staged trees whose corresponding ALDAGs have a restricted subset of labels, those associated to the search of context-specific independences only: see Sect. 5.3.1 for more details.

The code to replicate our analyses is available at [https://github.com/stagedtrees/stagedtrees\\_aldag](https://github.com/stagedtrees/stagedtrees_aldag). Datasets are freely available from the associated references.

## 2 Bayesian networks and conditional independence

Let  $G = ([p], F)$  be a directed acyclic graph (DAG) with vertex set  $[p] = \{1, \dots, p\}$  and edge set  $F$ . Let  $\mathbf{X} = (X_i)_{i \in [p]}$  be categorical random variables with joint mass function  $P$  and sample space  $\mathbb{X} = \times_{i \in [p]} \mathbb{X}_i$ . For  $A \subset [p]$ , we let  $\mathbf{X}_A = (X_i)_{i \in A}$  and  $\mathbf{x}_A = (x_i)_{i \in A}$  where  $\mathbf{x}_A \in \mathbb{X}_A = \times_{i \in A} \mathbb{X}_i$ . We say that  $P$  is Markov to  $G$  if, for  $\mathbf{x} \in \mathbb{X}$ ,

$$P(\mathbf{x}) = \prod_{k \in [p]} P(x_k | \mathbf{x}_{\Pi_k}),$$

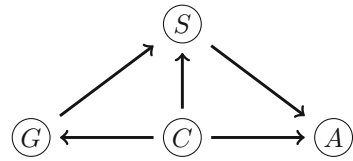
where  $\Pi_k$  is the parent set of  $k$  in  $G$  and  $P(x_k | \mathbf{x}_{\Pi_k})$  is a shorthand for  $P(X_k = x_k | \mathbf{X}_{\Pi_k} = \mathbf{x}_{\Pi_k})$ . Assuming  $\mathbf{X}$  is topologically ordered according to  $G$  (i.e. a linear ordering of  $[p]$  for which only pairs  $(i, j)$  where  $i$  appears before  $j$  in the order can be in the edge set), the ordered Markov condition implies conditional independences of the form

$$X_i \perp\!\!\!\perp \mathbf{X}_{[i-1]} | \mathbf{X}_{\Pi_i}, \tag{1}$$

which are equivalent to

$$P(x_i | \mathbf{x}_{[i-1] \setminus \Pi_i}, \mathbf{x}_{\Pi_i}) = P(x_i | \mathbf{x}_{\Pi_i}), \quad \text{for all } \mathbf{x}_{[i]} \in \mathbb{X}_{[i]}. \tag{2}$$

**Fig. 1** Learned BN for the Titanic dataset



**Definition 1** The *Bayesian network* model (associated to  $G$ ) is

$$\mathcal{M}_G = \{P \in \Delta_{|\mathbb{X}|-1} \mid P \text{ is Markov to } G\},$$

where  $\Delta_{|\mathbb{X}|-1}$  is the  $(|\mathbb{X}| - 1)$ -dimensional probability simplex.

Henceforth, we assume that the natural ordering of the positive integers  $[p]$  respects the topological order of the BN. Furthermore, we assume that  $P$  is strictly positive as common when probabilities are learned using smoothed maximum likelihood or Bayesian estimators.

To illustrate our methodology, we use throughout the paper the Titanic dataset (Dawson 1995), which provides information on the fate of the Titanic passengers, available from the `datasets` package bundled in R. Titanic includes four categorical variables: `Class` ( $C$ ) has four levels while `Gender` ( $G$ ), `Survived` ( $S$ ) and `Age` ( $A$ ) are binary. The BN learned using the hill-climbing algorithm implemented in the R package `bnlearn` (Scutari 2010) is reported in Fig. 1 and embeds the conditional independence  $\text{Age} \perp\!\!\!\perp \text{Gender} \mid \text{Class}, \text{Survived}$  only. Only one topological order of the variables exists and is henceforth used:  $C, G, S, A$ .

### 2.1 Non-symmetric conditional independence

BNs have the capability of expressing only symmetric conditional independences of the form in (1) and (2). The most common non-symmetric extension of conditional independence is the so-called context-specific independence which is often represented by associating a tree to each vertex of a BN (Boutilier et al. 1996). Let  $A, B$  and  $C$  be three disjoint subsets of  $[p]$ . We say that  $X_A$  is context-specific independent of  $X_B$  given context  $x_C \in \mathbb{X}_C$  if

$$P(x_A|x_B, x_C) = P(x_A|x_C) \tag{3}$$

holds for all  $(x_A, x_B) \in \mathbb{X}_{A \cup B}$  and write  $X_A \perp\!\!\!\perp X_B | x_C$ . The condition in (3) reduces to standard conditional independence in (1) if it holds for all  $x_C \in \mathbb{X}_C$ .

Pensar et al. (2016) introduced a more general definition of non-symmetric conditional independence called *partial conditional independence*. We say that  $X_A$  is partially conditionally independent of  $X_B$  in the domain  $\mathcal{D}_B \subseteq \mathbb{X}_B$  given context  $X_C = x_C$  if

$$P(x_A|x_B, x_C) = P(x_A|\tilde{x}_B, x_C) \tag{4}$$

holds for all  $(\mathbf{x}_A, \mathbf{x}_B), (\mathbf{x}_A, \tilde{\mathbf{x}}_B) \in \mathbb{X}_A \times \mathcal{D}_B$  and write  $X_A \perp\!\!\!\perp X_B | \mathcal{D}_B, \mathbf{x}_C$ . Clearly, (3) and (4) coincide if  $\mathcal{D}_B = \mathbb{X}_B$ . Furthermore, the sample space  $\mathbb{X}_B$  must contain more than two elements for a non-trivial partial conditional independence to hold.

A final condition is the so-called *local conditional independence*, first discussed in Chickering et al. (1997). For  $i \in [p]$  and an  $A \subset [p]$  such that  $A \cap \{i\} = \emptyset$ , local conditional independence expresses equalities of probabilities of the form

$$P(x_i | \mathbf{x}_A) = P(x_i | \tilde{\mathbf{x}}_A) \tag{5}$$

for all  $x_i \in \mathbb{X}_i$  and two  $\mathbf{x}_A, \tilde{\mathbf{x}}_A \in \mathbb{X}_A$ . Notice that in terms of generality, (3)  $\preceq$  (4)  $\preceq$  (5). Condition (5) simply states that some conditional probability distributions are identical, where no discernable patterns as in (3) and (4) can be detected.

Differently to any other probabilistic graphical model, the class of staged trees that we review next is able to graphically represent and formally encode any of the types of conditional independences defined in (2)–(5).

### 3 Staged trees

Differently from BNs, whose graphical representation is a DAG, staged trees visualize conditional independence using a colored tree. Let  $(V, E)$  be a directed, finite, rooted tree with vertex set  $V$ , root node  $v_0$ , and edge set  $E$ . For each  $v \in V$ , let  $E(v) = \{(v, w) \in E\}$  be the set of edges emanating from  $v$  and  $\mathcal{C}$  be a set of labels.

An  $\mathbf{X}$ -compatible staged tree is a triple  $T = (V, E, \theta)$ , where  $(V, E)$  is a rooted directed tree and:

1.  $V = v_0 \cup \bigcup_{i \in [p]} \mathbb{X}_{[i]}$ ;
2. For all  $v, w \in V$ ,  $(v, w) \in E$  if and only if  $w = \mathbf{x}_{[i]} \in \mathbb{X}_{[i]}$  and  $v = \mathbf{x}_{[i-1]}$ , or  $v = v_0$  and  $w = \mathbf{x}_1$  for some  $\mathbf{x}_1 \in \mathbb{X}_1$ ;
3.  $\theta : E \rightarrow \mathcal{L} = \mathcal{C} \times \bigcup_{i \in [p]} \mathbb{X}_i$  is a labelling of the edges such that  $\theta(v, \mathbf{x}_{[i]}) = (\kappa(v), x_i)$  for some function  $\kappa : V \rightarrow \mathcal{C}$ . The function  $\kappa$  is called the *colouring* of the staged tree  $T$ .

If  $\theta(E(v)) = \theta(E(w))$  then  $v$  and  $w$  are said to be in the same *stage*. The equivalence classes induced by  $\theta(E(v))$  form a partition of the internal vertices of the tree in *stages*.

Points 1 and 2 above construct a rooted tree where each root-to-leaf path, or equivalently each leaf, is associated with an element of the sample space  $\mathbb{X}$ . Then, a labeling of the edges of such a tree is defined where labels are pairs with one element from a set  $\mathcal{C}$  and the other from the sample space  $\mathbb{X}_i$  of the corresponding variable  $X_i$  in the tree. By construction,  $\mathbf{X}$ -compatible staged trees are such that two vertices can be in the same stage if and only if they correspond to the same sample space, and therefore, by construction, if they are at the same depth in the tree.

Figure 2 reports an example of an  $\mathbf{X}$ -compatible staged tree model for the Titanic dataset learned with the R package `stagedtrees`. The *coloring* given by the function  $\kappa$  is shown in the vertices and each edge  $(\cdot, (x_1, \dots, x_i))$  is labeled with  $x_i$ . The edge labeling  $\theta$  can be read from the graph combining the text label and the color of the emanating vertex. For example,  $\theta(v_1, v_6) \neq \theta(v_1, v_5)$ , while  $\theta(v_1, v_5) =$

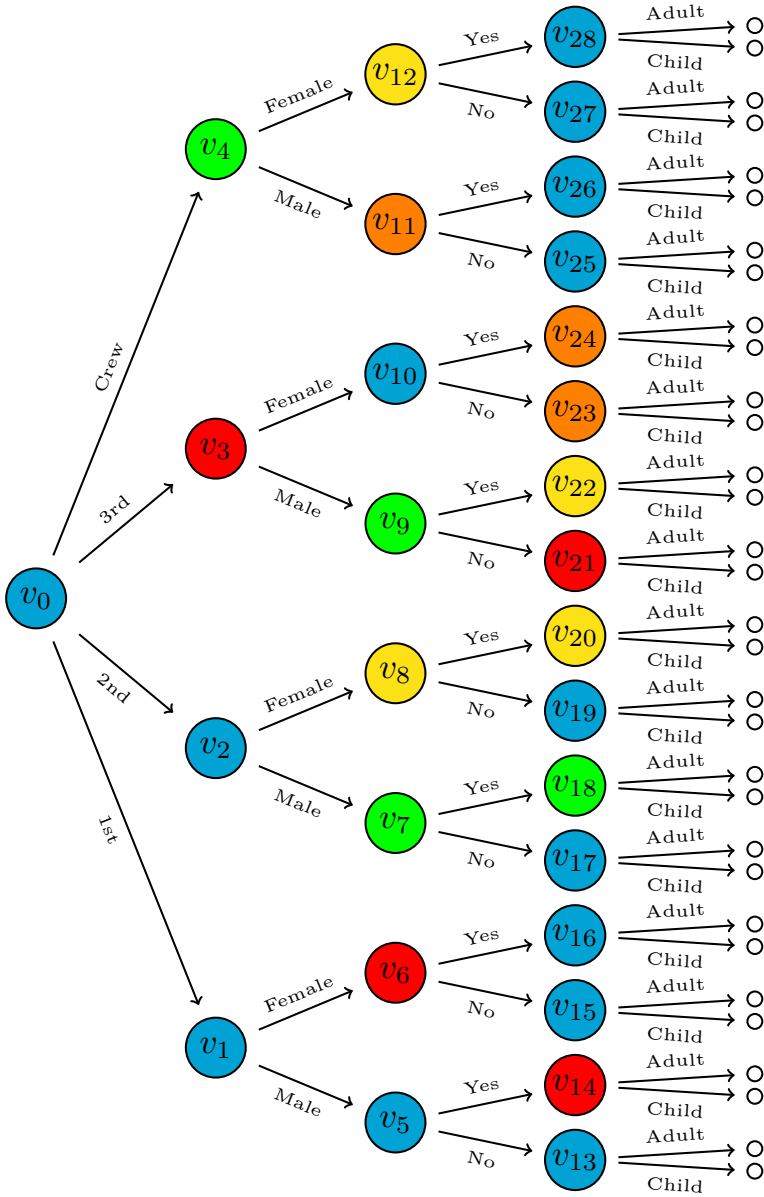


Fig. 2 A staged tree compatible with (Class, Gender, Survived, Age), learned over the Titanic dataset

$\theta(v_3, v_{10})$ . Similarly,  $\theta(v_2, v_7) = \theta(v_3, v_9)$ , while  $\theta(v_2, v_7) \neq \theta(v_4, v_{11})$ . This representation of the labeling  $\theta$  over vertices is equivalent to that over edges, while being more interpretable, and is henceforth used. There are 29 internal vertices and the staging is  $\{v_0\}, \{v_1, v_2\}, \{v_3\}, \{v_4\}, \{v_5, v_{10}\}, \{v_6\}, \{v_7, v_9\}, \{v_8, v_{12}\}, \{v_{11}\}, \{v_{13}, v_{15}, v_{16}, v_{17}, v_{19}, v_{25}, v_{26}, v_{27}, v_{28}\}, \{v_{14}, v_{21}\}, \{v_{18}\}, \{v_{20}, v_{22}\}$  and  $\{v_{23}, v_{24}\}$ .

The parameter space associated to an  $\mathbf{X}$ -compatible staged tree  $T = (V, E, \theta)$  with labeling  $\theta : E \rightarrow \mathcal{L}$  is defined as

$$\Theta_T = \left\{ \mathbf{y} \in \mathbb{R}^{|\theta(E)|} \mid \forall e \in E, y_{\theta(e)} \in (0, 1) \text{ and } \sum_{e \in E(v)} y_{\theta(e)} = 1 \right\}. \tag{6}$$

Equation (6) defines a class of probability mass functions over the edges emanating from any internal vertex coinciding with conditional distributions  $P(x_i | \mathbf{x}_{[i-1]})$ ,  $\mathbf{x} \in \mathbb{X}$  and  $i \in [p]$ .

Let  $I_T$  denote the leaves of a staged tree  $T$ . Given a vertex  $v \in V$ , there is a unique path in  $T$  from the root  $v_0$  to  $v$ , denoted as  $\lambda(v)$ . The *depth* of a vertex  $v \in V$  equals the number of edges in  $\lambda(v)$ . For any path  $\lambda$  in  $T$ , let  $E(\lambda) = \{e \in E : e \in \lambda\}$  denote the set of edges in  $\lambda$ .

**Definition 2** The *staged tree model*  $\mathcal{M}_T$  associated to the  $\mathbf{X}$ -compatible staged tree  $(V, E, \theta)$  is the image of the map

$$\begin{aligned} \phi_T : \Theta_T &\rightarrow \Delta_{|I_T|-1}^\circ \\ \mathbf{y} &\mapsto \left( \prod_{e \in E(\lambda(l))} y_{\theta(e)} \right)_{l \in I_T} \end{aligned} \tag{7}$$

Therefore, staged tree models are such that atomic probabilities are equal to the product of the edge labels in root-to-leaf paths and coincide with the usual factorization of mass functions via recursive conditioning.

Conditional independence is formally modeled and represented in staged trees via the labeling  $\theta$ . As an illustration, consider the staged tree in Fig. 2 for the Titanic dataset. The fact that  $v_1$  and  $v_2$  are in the same stage represents the partial independence  $\text{Gender} \perp\!\!\!\perp \text{Class} \mid \{1\text{st}, 2\text{nd}\}$ . Considering vertices at depth two, the green and yellow staging again represents partial conditional independences. More interesting is the blue staging of the vertices  $v_5$  and  $v_{10}$  which implies  $P(S = s \mid \text{Female}, 3\text{rd}) = P(S = s \mid \text{Male}, 1\text{st})$ , i.e. the probability of survival for females traveling in third class is the same as that of males traveling in first class. Such a statement is a generic local conditional independence. Considering the last level, we can notice a very non-symmetric staging structure. As an illustration, consider the top four vertices  $v_{25}, v_{26}, v_{27}$  and  $v_{28}$  belonging to the same stage. This implies the context-specific independence  $\text{Age} \perp\!\!\!\perp \text{Survived}, \text{Gender} \mid \text{Class} = \text{Crew}$ . The staged tree in Fig. 2, embedding the above non-symmetric conditional independences, gives a more refined representation of the data than the BN in Fig. 1. Indeed, the BIC of the staged tree can be computed as 10,440.39, while the one of the BN is larger and equal to 10,502.28 (see G3rgen et al. 2022, for a discussion of using the BIC for

staged trees). A more nuanced evaluation of the performance of staged tree vs BNs is carried out below in our computational experiment of Sect. 6.

This example illustrates the capability of staged trees to represent any non-symmetric conditional independence graphically. Although such independences can be read directly from the tree via visual inspection, it becomes challenging to detect them as the size of the tree increases. Below we formalize how to assess the type of conditional independence existing between pairs of random variables.

## 4 Staged trees and Bayesian networks

Although the relationship between BNs and staged trees was already formalized by Smith and Anderson (2008), we introduce an implementable routine to transform a DAG to its equivalent staged tree.

Assume  $X$  is topologically ordered with respect to a DAG and consider an  $X$ -compatible staged tree with vertex set  $V$ , edge set  $E$  and labeling  $\theta$  defined via the coloring  $\kappa(\mathbf{x}_{[i]}) = \mathbf{x}_{\Pi_i}$  of the vertices. The staged tree  $T_G$ , with vertex set  $V$ , edge set  $E$  and labeling  $\theta$  so constructed, is called *the staged tree model of  $G$* . Importantly,  $\mathcal{M}_G = \mathcal{M}_{T_G}$ , i.e. the two models are exactly the same since they entail exactly the same factorization of the joint probability. The staging of  $T_G$  represents the Markov conditions associated with the graph  $G$ .

As an illustration, Fig. 3 reports the tree  $T_G$  associated with the BN in Fig. 1. Since the variables `Class`, `Gender` and `Survived` are fully connected in the BN, the associated staged tree is such that vertices at depth one and two are in their own individual stages. The only symmetric conditional independence embedded in the BN is represented by joining pairs of vertices at depth three (associated with the variable `Age`) in the same stage. Clearly, the staging of the staged tree representing a BN in Fig. 3 exhibits a lot more symmetry than the one in Fig. 2, which can represent a wide array of non-symmetric independences.

Our first contribution is the solution to the following inverse problem: given an  $X$ -compatible staged tree  $T = (V, E, \theta)$  find the corresponding DAG  $G$ . This DAG cannot represent, in general, the same staged tree model, since BNs cannot represent non-symmetric conditional independences. Nevertheless, we prove that we can retrieve a minimal DAG, in a sense that we formalize next. A proof of the result is in the “Appendix”.

**Proposition 1** *Let  $T = (V, E, \theta)$  be an  $X$ -compatible staged tree, with  $\kappa : V \rightarrow \mathcal{C}$  the vertex labeling that defines  $\theta$ . Let  $G_T = ([p], F_T)$  be the DAG with vertex set  $[p]$  and whose edge set  $F_T$  includes the edge  $(k, i)$ ,  $k < i$ , if and only if there exist  $\mathbf{x}_{[i-1]}, \mathbf{x}'_{[i-1]} \in \mathbb{X}_{[i-1]}$  such that  $x_j = x'_j$  for all  $j \neq k$  and*

$$\kappa(\mathbf{x}_{[i-1]}) \neq \kappa(\mathbf{x}'_{[i-1]}). \quad (8)$$

*Then  $G_T = ([p], F_T)$  is the minimal DAG such that  $\mathcal{M}_T \subseteq \mathcal{M}_{G_T}$ , in the sense that for every DAG  $G = ([p], F)$  such that  $1, \dots, p$  is a topological order, if  $\mathcal{M}_T \subseteq \mathcal{M}_G$*

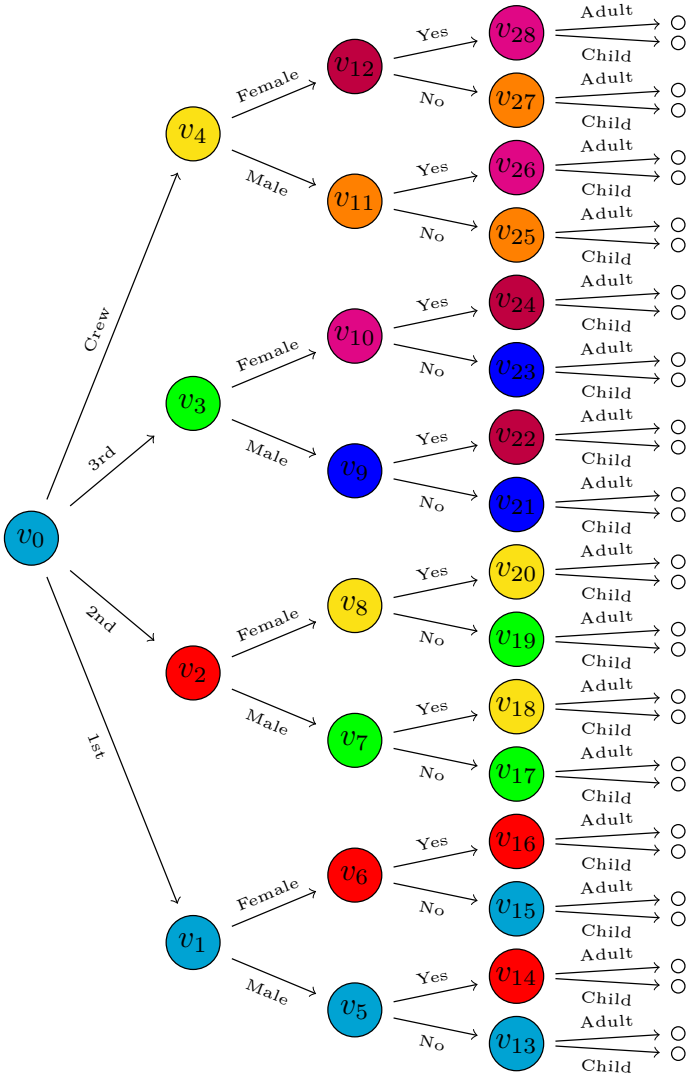


Fig. 3 The staged tree representation of the BN in Fig. 1 for the Titanic dataset

then  $F_T \subseteq F$ . In particular  $X_A \perp\!\!\!\perp X_B | X_C$  holds in  $\mathcal{M}_T$  if and only if  $A$  and  $B$  are  $d$ -separated by  $C$  in  $G_T$ .

**Corollary 1** In the setup of Proposition 1,  $\mathcal{M}_{T_G} = \mathcal{M}_{G_{T_G}}$ .

A staged tree  $T$  is therefore a sub-model of the resulting  $G_T$  which embeds the same set of symmetric conditional independences. The BN  $G_T$  is minimal in the sense that it includes the smallest number of edges among all possible BNs that include  $\mathcal{M}_T$  as a sub-model. The models  $\mathcal{M}_T$  and  $\mathcal{M}_{G_T}$  are equal if and only if  $T$  embeds only symmetric conditional independences. As an illustration consider the staged tree in

Fig. 2. It can be shown using Proposition 1 that the associated BN  $G_T$  is complete and therefore it must be that  $\mathcal{M}_T \subseteq \mathcal{M}_{G_T}$ . Conversely, if the staged tree in Fig. 3 is transformed into a BN, then using Corollary 1 the resulting BN must be the one in Fig. 1.

Importantly, Proposition 1 gives a novel criterion to read symmetric conditional independence statements from a staged tree, by transforming it into a BN whose structure represents the same equalities of the form in (2). Conditional independence statements in the staged tree can then be read from the associated BN using the d-separation criterion (see e.g. Darwiche 2009). For instance, the staged tree in Fig. 2 does not embed any symmetric conditional independence, since the associated BN is complete.

The ‘‘Appendix’’ gives a detailed implementation of both conversion algorithms, from BN to staged tree and vice versa.

## 5 Non-symmetric dependence and DAGs

Proposition 1 identifies if there is a dependence between two random variables in a  $\mathbf{X}$ -compatible staged tree  $T$  and, in such a case, draws an edge in  $G_T$ . However, the staged tree carries much more information about the relationship between the two variables. In this section, we introduce methods to label the edges of  $G_T$  so as to depict some of the information about the non-symmetric independences of  $T$  in  $G_T$ .

### 5.1 Classes of statistical dependence

First, we need to characterize the type of dependence existing between two random variables joined by an edge in a DAG  $G$ .

**Definition 3** Let  $P$  be the joint mass function of  $\mathbf{X}$

and  $P$  be Markov with respect to a DAG  $G = ([p], F)$ . For each  $(j, i) \in F$  we say that the dependence of  $X_i$  from  $X_j$  is of class

- *context*, if  $X_i$  and  $X_j$  are context-specific independent given some context  $\mathbf{x}_C$  with  $C = \Pi_i \setminus \{j\}$ .
- *partial*, if  $X_i$  is partially conditionally independent of  $X_j$  in a domain  $\mathcal{D}_j \subset \mathbb{X}_j$  given a context  $\mathbf{x}_C$  with  $C = \Pi_i \setminus \{j\}$ ; and  $X_i$  and  $X_j$  are not context-specific independent given the same context  $\mathbf{x}_C$ .
- *local*, if none of the above hold and a local independence of the form  $P(x_i | \mathbf{x}_{\Pi_i}) = P(x_i | \tilde{\mathbf{x}}_{\Pi_i})$  is valid where  $x_j \neq \tilde{x}_j$ .
- *total*, if none of the above hold.

Notice that if the class of dependence between  $X_i$  and  $X_j$  is context or partial, then there may also be local independence statements as in (5) involving these two variables. Similarly, the dependence between  $X_i$  and  $X_j$  can be both context and partial with respect to two different contexts. On the other hand, if their class of dependence is local, then, by definition, there are no context-specific or partial equalities.

Proposition 1 paves the way to assess the class of dependence existing between  $X_i$  and  $X_j$ . In particular, one has to check if there are equalities of the form (8) for all or some  $\mathbf{x}_{[i]} \in \mathbb{X}_{[i]}$  and, if so, to which class they correspond. A discussion of the implementation of such checks is in the ‘‘Appendix’’. As illustrated in Sect. 6, these can be performed quickly, although all combinations of ancestral variables have to be considered.

### 5.2 Asymmetry-labeled DAGs

An edge in a BN represents, by construction, a total dependence between two random variables. However, the flexibility of staged trees allows us to assess if such dependence is of any other of the classes introduced in Definition 3. This observation leads us to define a new graphical representation, that we term *asymmetry-labeled DAG* (ALDAG), where edges are colored depending on the type of relationship between variables.

Formally, let  $G$  be a DAG and  $F$  its edge set and

$$\mathcal{L}^A = \{ \text{‘context’}, \text{‘partial’}, \text{‘context/partial’}, \text{‘local’}, \text{‘total’} \}$$

be the set of edge labels marking the type of dependence.

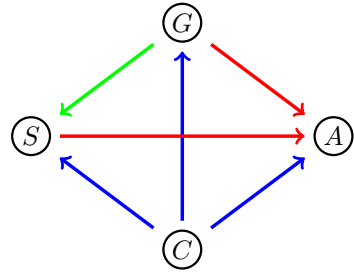
**Definition 4** An ALDAG is a pair  $(G, \psi)$  where  $G = ([p], F)$  is a DAG and  $\psi$  is a function from the edge set of  $G$  to  $\mathcal{L}^A$ , i.e.  $\psi : F \rightarrow \mathcal{L}^A$ . We say that a joint mass function  $P$  is compatible with an ALDAG  $(G, \psi)$  if  $P$  is Markov to  $G$  and additionally  $P$  respects all the edge labels given by  $\psi$ ; that is, for each  $(j, i) \in F$ ,  $X_i$  is  $\psi(i, j)$  dependent from  $X_j$ .

Henceforth, we represent the labeling by coloring the edges of the ALDAG. Standard BNs have an ALDAG representation where all edges have label ‘total’. Notice that standard features of BNs are also valid over ALDAGs: for instance, the already-mentioned d-separation criterion. Furthermore, probability queries can be efficiently answered from the ALDAG using any of the standard propagation algorithms of BNs (e.g. using junction trees, Koller and Friedman 2009). In Leonelli and Varando (2023), we formalized the use of ALDAGs to explore the equivalence class of staged trees and therefore learn complex causal relationships from observational data. Additional potential features of ALDAGs are discussed in the conclusions below.

ALDAGs share features with labeled DAGs of Pensar et al. (2015), but they differ in two critical aspects: first, labeled DAGs can only embed context-specific independence while ALDAGs represent any asymmetric independence; second, labeled DAGs specifically report the contexts over which independences hold, while ALDAGs do not. There are two reasons behind this: on one hand, the specific independences in ALDAGs can be read from the associated staged tree; on the other, for applications with a larger number of variables, the required contexts are often too complex to be reported within the DAG.

As an illustration of ALDAGs consider the staged tree for the Titanic data in Fig. 2 which, using Proposition 1, is transformed into the ALDAG in Fig. 4. Although

**Fig. 4** An ALDAG for the Titanic dataset constructed from the staged tree in Fig. 2. The edge coloring is: red—context; blue—partial; green—local (color figure online)



the ALDAG does not carry all the information stored in the staged tree, which is quite complex to read, it intuitively describes the classes of dependence among the random variables. The blue edges denote a partial dependence between `Class` and any other variable. The red edges denote that `Age` has a context dependence with `Gender` and `Survived`. Notice importantly that in the standard BN, which does not have the flexibility to embed non-symmetric independences, the variables `Age` and `Gender` were considered conditionally independent. Lastly, the green edge between `Gender` and `Survived` implies that there is only a local dependence between these two variables, and therefore there is no specific pattern guiding the equalities of probabilities between these two variables. Such extended forms of dependence better describe the fate of the Titanic passengers since, as already noticed, the BIC of the associated staged tree is smaller than the one of the BN.

### 5.3 Constructing ALDAGs

ALDAGs can be obtained from estimated staged trees, and, in particular, with the following routine: (i) learn a staged tree model  $T$  from data, using, for instance, any of the algorithms in `stagedtrees`; (ii) transform  $T$  into  $G_T$  as in Proposition 1; (iii) assign a label to each edge of  $G_T$  by checking the equalities in (8) that hold in  $T$ . Steps (ii) and (iii) are implemented in the `stagedtrees` R package using the algorithms in the “Appendix”. Illustrations of the checks in step (iii) are given in Sect. 5.3.1 below.

The most critical and computationally expensive step of learning an ALDAG is the staged tree learning step (i). There is a large literature on learning staged trees from data (Carli et al. 2022, 2023; Cowell and Smith 2014; Freeman and Smith 2011; Leonelli and Varando 2024b; Silander and Leong 2013). Here we consider the available algorithms implemented in the `stagedtrees` R package (Carli et al. 2022). In particular, we will use the following algorithms, which work with a fixed order of the variables (see Carli et al. 2022, for more details):

- a hill-climbing (HC) algorithm which at each step either joins or splits vertices of the tree in stages by optimizing a model score (usually the BIC);
- a backward (hill-climbing) (BHC) algorithm which, at each step, can only join stages together by optimizing a model score.
- a novel backward algorithm which iteratively add context-specific independences (CSBHC, see Sect. 5.3.1).

Furthermore, any of the above-mentioned algorithms can be used within the dynamic programming approach of Cowell and Smith (2014) to choose an optimal order of the variables (Leonelli and Varando 2023).

An ALDAG can also be obtained as a refinement of the DAG of a BN by the addition of edge labels indicating the class of dependence. Given a DAG  $G$ , the following steps implement such a refinement: (i) transform  $G$  into the staged tree  $T_G$  using any of the topological orders of variables; (ii) run a backward hill-climbing algorithm using  $T_G$  as starting model and obtain a new tree  $T$ ; (iii) transform  $T$  into  $G_T$  and apply the edge-labeling. The resulting ALDAG has an edge set equal to or a subset of the edge set of  $G$ . Furthermore, the edge set is now labeled and denoting the classes of dependence.

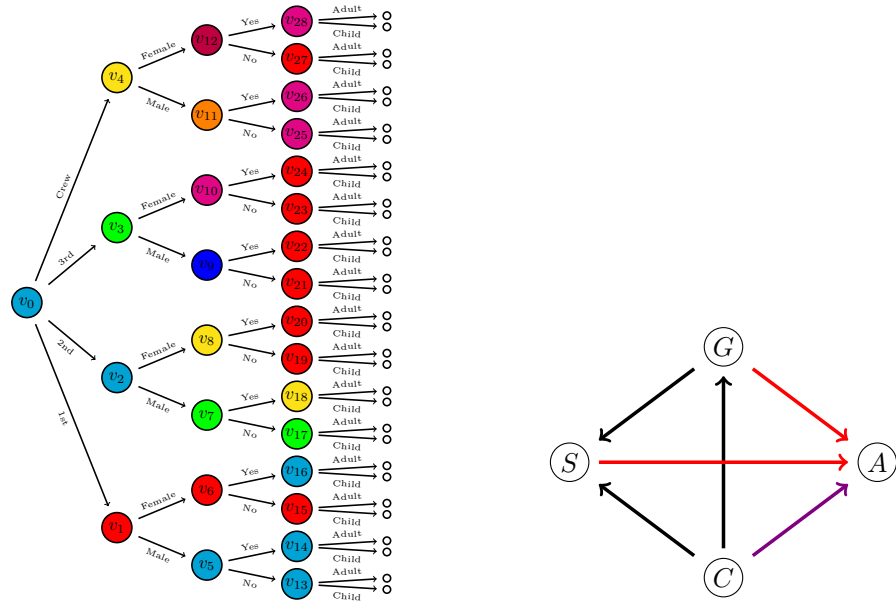
DAGs with tree-parametrized CPTs (as in Hyttinen et al. 2018; Pensar et al. 2016; Talvitie et al. 2019) and staged tree methods are very similar approaches that use trees to represent conditional probabilities. In particular, the statistical models represented by staged tree and DAGs with tree-CPTs are, in principle, equivalent. Even the learning algorithm proposed by Pensar et al. (2016) consists of a heuristic search using splitting and joining operations on each CPT tree, similar to the hill-climbing moves mentioned above. In the staged tree approach, the difference is that we do not assume a sparse DAG between variables by default and do not search both a DAG and sparse CPTs. Of course, restricting to sparse DAGs and searching over refinements only is beneficial from a computational perspective, and it has been proposed and shown to be effective for staged trees (Barclay et al. 2013; Leonelli and Varando 2022). Furthermore, in the context of staged trees, we are able to restrict the model search in meaningful ways, for instance, by only considering context-specific independences as in the novel algorithm introduced in Sect. 5.3.1 below. Unfortunately, we are not aware of any available implementation of these related methods, and we are thus unable to run any empirical comparisons.

For the purpose of this paper, when learning ALDAGs as a refinement of BNs, we select one topological order of the variables at random (the one automatically provided by `bnlearn`). In Leonelli and Varando (2024a), we have developed algorithms that select the best staged tree out of all compatible orders to a DAG or its CPDAG.

### 5.3.1 Searching context-specific independences

The flexibility of staged trees to represent any non-symmetric independence has three major drawbacks: first, reading independences from the tree can become complex; second, learning trees from data can be computationally very expensive; third, representing larger systems can become challenging. To address the first two difficulties, we introduce a new heuristic search for the stage structure, motivated by our new definition of the ALDAG. In particular, we consider a backward hill-climbing algorithms that, for each variable, iteratively adds context-specific independence relationships to optimize a given score (e.g. BIC), called CSBHC. In particular, at each step, the algorithm searches all possible additional context-specific conditional independences of the form,

$$X_i \perp\!\!\!\perp X_j | X_C = x_C \quad j < i$$



**Fig. 5** A staged tree learned with the proposed CSBHC algorithm over the Titanic dataset and its associated ALDAG. The edge coloring is: red—context; purple—context/partial; black—total (color figure online)

where  $C = [i] \setminus \{i, j\}$  and thus  $x_C \in \mathbb{X}_C$  is a context specified by all variables preceding  $X_i$  in the tree except  $X_j$ .

The algorithm is based on the construction of *stage matrices*, summarizing the staging of the vertices at a specific depth with respect to a preceding variable. These are fully defined in the “Appendix”; we just give the intuition behind them here. Figure 5 reports the staged tree learned with the CSBHC algorithm for the Titanic dataset. Consider the variable Age corresponding to vertices  $v_{13} - v_{28}$  and number the stages from bottom to top of the tree (blue—1; red—2; green—3; yellow—4; pink—5). The stage matrix of Age with respect to Survived is

$$\begin{pmatrix} 1 & 2 & 3 & 2 & 2 & 2 & 5 & 2 \\ 1 & 1 & 4 & 2 & 2 & 2 & 5 & 5 \end{pmatrix}, \tag{9}$$

where each row is associated with an element of the sample space of Age (first row—Child; second row—adult), and each column is a context defined by all preceding variables to Age excluding Survived (for instance, the first column represents Class = 1st and Gender = Male). The equality of all elements in a column represents a context-specific independence. For instance, the fourth column of the stage matrix, including only the label 2, represents the independence between Age and Survived when Class = 2nd and Gender = Female. The CSBHC algorithm searches context-specific independences by setting elements of individual columns in stage matrices equal to each other at each iteration.

Figure 5 further reports the ALDAG learned with the CSBHC algorithm, including edges with total, context, and context/partial labels. Notice that the combination of multiple context-specific independences with respect to a variable may lead to a partial independence with another. To see this, consider the stage matrix of Age with respect to Class, equal to

$$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 3 & 4 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 5 & 5 & 2 & 5 \end{pmatrix}.$$

The third column represents the independence between Age and Class in the context Gender = Female and Survived = No. The fourth column has two elements equal to each other (in positions 2, 4 and 3, 4), representing a partial conditional independence (conditionally on Gender = Female and Survived = Yes, the probability distribution of Age is the same for travelers of 2nd and 3rd Class). Notice that this partial conditional independence is equivalent to the combination of the context-specific independences between Age and Survived for Gender = Female and Class = 2nd or 3rd. More formally, the fact that the 4th and 6th columns of the matrix in Eq. (9) have all elements equal implies context-specific independences. However, since the elements are equal in the two columns (coded as 2), this generates a partial conditional independence. For this reason, the CSBHC algorithm can also identify context/partial types of relationships.

The staged tree in Fig. 5 has a BIC of 10,479, which is worse than the one of the generic staged tree (BIC = 10,440), but still better than the BN (BIC = 10,502), again highlighting the need for models embedding non-symmetric independences.

## 6 Applications

We now consider a variety of datasets commonly used in the probabilistic graphical models literature. First, we carry out an experiment to assess the performance of ALDAGs as well as the complexity of our routines. Then we consider two additional real-world applications to illustrate the capabilities of staged trees and ALDAGs.

### 6.1 Computational experiment

Nine datasets, which are either available in R packages or downloaded from the UCI repository, are considered. For each dataset, a DAG is first learned by optimizing the BIC score (using a tabu greedy search, Scutari 2010) and then both the BHC and the proposed CSBHC algorithms are used to refine the DAG to a staged tree. The learned staged trees are then transformed into ALDAGs using Algorithm 2 given in the ‘‘Appendix’’. Additionally, the test log-likelihood for a holdout test set is computed. The above is repeated for all 20 repetitions of a 20-fold cross-validation scheme. The results, summarized in Tables 1, 2 and 3, suggest the following:

**Table 1** Results of the data experiments

	Variables	Edg. ALDAG-BHC	Edg. ALDAG-CSBHC
Abalone	9	(5, 10, 0, 0, 5)	(8.5, 11.5, 0, 0, 0)
Breastcancer	10	(4, 0, 3, 0, 0)	(7, 0, 0, 0, 0)
Creditapproval	14	(11, 2, 0, 0, 1)	(11, 3, 0, 0, 0)
Housevotes	17	(10, 9, 0, 0, 6)	(13, 13, 0, 0, 0)
Indianliver	11	(8.5, 1, 0, 0, 5)	(14, 1, 0, 0, 0)
Nursery	9	(0, 0.5, 5, 3, 0)	(5, 1.5, 0, 2, 0)
Phdarticles	6	(3, 0, 1, 0, 0)	(4, 0, 0, 0, 0)
Tic-tac-toe	10	(1, 0, 11, 0, 0)	(6.5, 6, 0, 0, 0)
Titanic	4	(0, 1, 3, 0, 1)	(4, 1, 0, 0, 0)

Edg. ALDAG-\* columns report the median (across the 20-folds repetition) number of edges by type (total, context, partial, context/partial, local) in the two ALDAGs obtained by refinement of the DAG with BHC and CSBHC algorithms respectively

- Staged trees provide a more refined representation of the datasets considered than the standard BN (lower BICs in Table 1), thus highlighting the need to consider models which embed asymmetric conditional independences to untangle complex dependence structures.
- DAG refined with both BHC and CSBHC obtain comparable log-likelihood on holdout test sets (Table 2), showing that the additional flexibility of the models does not necessarily translate into over-fitting.
- Only a small fraction of the edges learned via BN structural search algorithms are related to a symmetric dependence between variables. All but two ALDAGs have a number of edges with a label that is not total (see Table 1).
- The construction of the ALDAG does not impose a computational burden with computational times comparable to the staged tree model selection step. Furthermore, the experiment shows that the methods are efficiently implemented even for a medium-large number of variables.

## 6.2 Aspects of everyday life

We next illustrate the use of staged trees to uncover dependence structures using data from the 2014 survey “Aspects on everyday life” collected by ISTAT (the Italian National Institute of Statistics) (ISTAT 2014). The survey collects information from the Italian population on a variety of aspects of their daily lives. For the purpose of this analysis we consider five of the many questions asked in the survey: do you practice sports regularly? (S = yes/no); do you have friends you can count on? (F = yes/no/unsure); do you believe in people? (B = yes/no); what’s your gender (G = male/female); what grade would you give to your life? (L = low/medium/high).<sup>1</sup> Instances with a missing answer were dropped, resulting in 38,156 answers to the

<sup>1</sup> The original grade is numeric between zero and ten and it has been aggregated as follows: 0–5/low; 6–8/medium; 9–10/high.

**Table 2** Results of the data experiments

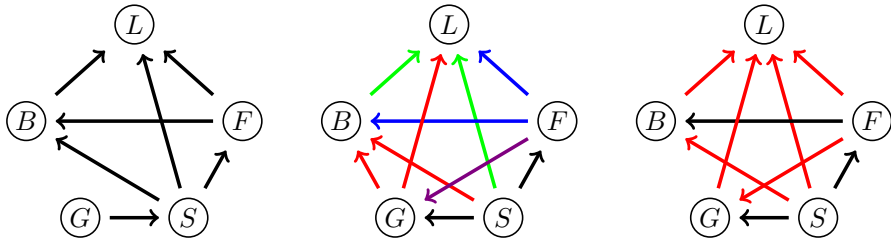
	BIC DAG	BIC BHC	BIC CSBHC	Log-lik test DAG	Log-lik test BHC	Log-lik test CSBHC
Abalone	27,682.79	27,469.04	27,492.40	- 721.88	- 720.54	- 721.06
Breastcancer	4422.89	4372.53	4422.89	- 111.99	- 111.64	- 111.99
Creditapproval	9392.59	9382.55	9384.08	- 247.00	- 247.37	- 247.50
Housevotes	3369.16	3315.08	3321.44	- 93.23	- 92.40	- 92.40
Indianliver	6644.12	6619.34	6634.51	- 173.10	- 173.51	- 173.72
Nursery	239,992.06	239,364.93	239,810.29	- 6285.96	- 6286.04	- 6285.71
Phdarticles	7958.60	7955.42	7956.64	- 206.90	- 206.89	- 206.82
Tic-tac-toe	18,337.07	18,100.26	18,299.50	- 473.77	- 474.32	- 475.26
Titanic	9985.09	9934.70	9970.41	- 258.52	- 259.34	- 259.03

Median BIC scores and test log-likelihood values across the 20-fold cross-validation repetitions

**Table 3** Median elapsed time (in s) for the data experiments across the 20-fold cross-validation repetitions

	DAG	BHC	CSBHC	ALDAG
Abalone	0.0655	0.1140	0.1015	0.00950
Breastcancer	0.0130	0.0190	0.0705	0.28050
Creditapproval	0.0170	0.0135	0.0995	0.42250
Housevotes	0.0250	0.0390	0.3400	0.83925
Indianliver	0.0255	0.0130	0.0220	0.02675
Nursery	0.1085	0.3280	0.1005	0.10000
Phdarticles	0.0185	0.0050	0.0080	0.00325
Tic-tac-toe	0.0170	0.0765	0.0745	0.09400
Titanic	0.0165	0.0325	0.0145	0.00150

The DAG column reports the seconds to estimate the DAG; The columns BHC and CSBHC report the seconds needed to refine the DAG to a staged tree, with the BHC and the CSBHC algorithms respectively; lastly the ALDAG column contains the time to build the ALDAG from the staged trees



**Fig. 6** BN for the aspects of everyday life data (left) as well as ALDAGs over the same data associated to the staged trees learned with BHC (center) and CSBHC (right). The edge coloring is: red—context; blue—partial; violet—context/partial; green—local; black—total (color figure online)

survey. Our aim is to analyze how various factors affect the life grade of the Italian population.

A BN with the variable life grade (L) as downstream variable is learned using the hill-climbing function (optimizing the BIC score) in the `bnlearn` package by blocklisting all outbound edges from L. The learned DAG is reported in Fig. 6 (left). This embeds the symmetric conditional independence  $L, B, F \perp\!\!\!\perp G | S$ : given the level of sports activity, gender has no effect on life grade, trust in people, and the availability of friends. The learned BN has a BIC of 251,781.

A staged tree over the same dataset is learned as follows. First, we learn a staged tree with the hill-climbing algorithm (optimizing BIC score) and considering all possible orders of the variables but life grade (L), which we then fix as the last variable of the tree. The resulting staged tree is plotted in Fig. 7 (left).

The staging of the life grade variable reveals a complex dependence pattern from which interesting conclusions can be drawn. For instance, conditionally on whether individuals believe in people, life grade does not depend on gender for those that practice sports and have friends they can count on (stages  $v_{37} - v_{40}$ ). Similarly, conditionally on whether individuals believe in people, the distribution of life grade is the same for

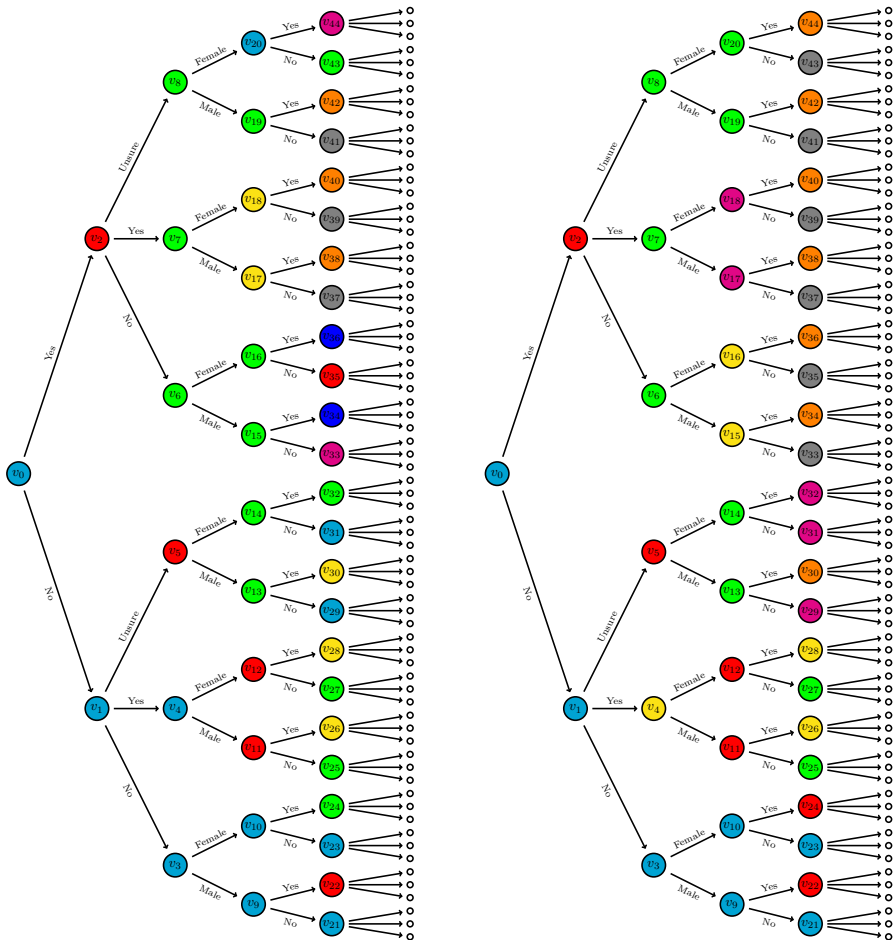


Fig. 7 Staged tree learned using the hill-climbing algorithm (left) and the CSBHC algorithm (right) over the aspects of everyday life data with variables' order S, F, G, B, L

male individuals that practice sports who either have friends or are unsure about it (stages  $v_{37}$ ,  $v_{38}$ ,  $v_{41}$ ,  $v_{42}$ ). It can also be noticed that gender almost has no effect on whether individuals believe in others, the only dependence existing for individuals that practice sports and are unsure about having friends (stages  $v_{19}$ – $v_{20}$ ). These are just a few of the many conclusions that can be drawn from the tree, and a whole explanation is beyond the scope of this analysis.

Although the staged tree in Fig. 7 can still be visually inspected, it is already rather extensive, with 72 leaves and 45 internal vertices. Its associated ALDAG in Fig. 6 (middle) provides a compact summary of the dependence structure and shows that all variables are related to each other according to different types of dependence.

An alternative tree is learned with the CSBHC algorithm over the same dataset and variable ordering and is reported in Fig. 7 (right). The tree has a staging that is

**Table 4** Variables from the 2012 ISTAT enterprise innovation survey

Name	Explanation	Levels
GP	Belongs to an industrial group	Yes/no
LARMAR	Main market	Regional/national/international
INPDDG	Product innovation 2010–2012	Yes/no
INPDSV	Service innovation 2010–2012	Yes/no
INPD	Other innovations 2010–2012	Yes/no
INABA	Abandoned innovation 2008–2010	Yes/no
INONG	Ongoing innovation from 2008–2010	Yes/no
CO	Cooperation agreements for innovation	Yes/no
ORG	New organization practices	Yes/no
MKT	New marketing practices	Yes/no
PUB	Contracts with public institutions	Yes/no
EMP	Number of employees	10–49/50–249/> 250
EMPUD	Employees with degree	0%/1–10%/> 10%
RD	Research and development	Yes/no
GROWTH	Increased revenue 2012/2010	Yes/no

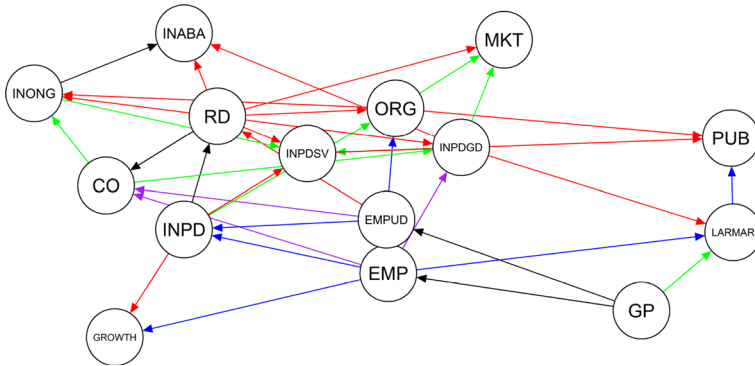
a lot more symmetric than the one of the generic staged tree. For instance, it states that life grade is independent of gender and the availability of friends conditionally on whether you believe in people and on conducting sports activity (stages  $v_{33}$ – $v_{44}$ ). Also, the staging of the vertices  $v_9$ – $v_{20}$  reports that, given a specific level of sports activity and the availability of friends, gender does not affect whether an individual believes in people. The associated ALDAG in Fig. 6 (right) is therefore not complete and has a missing edge from G to B. It can also be noticed that edges are only of type total or context. Compared to the standard BN, both ALDAGs show additional patterns of dependence that can be retrieved because the assumption of symmetric dependence was relaxed.

Notice that both the generic staged tree and the alternative staged tree obtained with CSBHC, provide a better description of the dependence structure of the data since they have lower BICs than the one associated with the BN, 251,648 and 251,673, respectively.

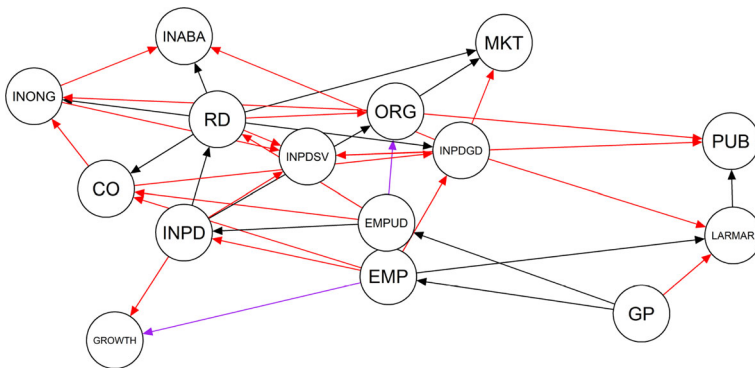
### 6.3 Enterprise innovation

We next consider data from the 2012 Italian enterprise innovation survey, again collected by ISTAT (ISTAT 2015). The survey reports information about medium-sized Italian companies and their involvement with innovation in the 3-year period between 2010 and 2012. The aim of the analysis is to assess which factors related to innovation are connected with changes in the company revenue.

Out of the many questions in the survey, we consider 14 factors that could influence the revenue of an enterprise. The variables considered are summarized in Table 4.



**Fig. 8** BN (without considering edge coloring) and ALDAG for the enterprise innovation survey data. The edge coloring is: red—context; blue—partial; violet—context/partial; green—local; black—total (color figure online)



**Fig. 9** ALDAG for the enterprise innovation survey data learned with the CSBHC algorithm. The edge coloring is: red—context; violet—context/partial; black—total (color figure online)

Instances with a missing answer were dropped, resulting in 8938 answers to the survey. In this case, it is unfeasible to study the staged tree directly since it would have more than 100 k leaves, making it impossible to visualize its staging.

Therefore, we follow the alternative strategy outlined in Sect. 5.3 of creating an ALDAG as a refinement of a BN. Thus, we first learn a BN from data, reported in Fig. 8 (by not considering the edge coloring). Interestingly, the BN suggests that the only two factors that have a direct influence on the change of revenue of a company (GROWTH) are the number of employees (EMP) and whether it carried out innovations of other types in the past 3 years (INPD)—meaning not product or service innovations.

The resulting BN is refined into an ALDAG using the backward hill-climbing algorithm, which can only join stages together and is reported in Fig. 8. Of the original 35 edges, only five are still of type total, embedding symmetric dependences. All other edges are colored, indicating that there are types of dependence in the data that cannot be represented symmetrically. This is confirmed by the BIC of the ALDAG, which is equal to 133,689, much lower than the one of the BN (134,311).

The CSBHC is also used to refine the learned BN, and the associated ALDAG is reported in Fig. 9. This consists of 14 total, 20 context, and 2 context/partial edges, confirming the presence of non-symmetric patterns in the data. This ALDAG also gives a more refined representation of the variables' relationships, having a BIC of 133,872.

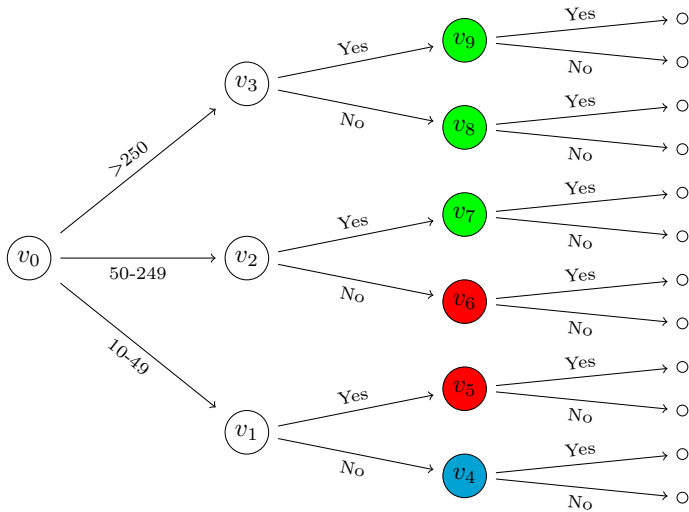
Since GROWTH is independent of all other variables conditionally on EMP and INPD and our interest is in assessing how factors are relevant for the change of revenue, we can construct a tree with these three variables only and delete all those that are conditionally independent. We call such a tree the *dependence subtree*. This is reported in Fig. 10 for GROWTH and the ALDAG in Fig. 8. It shows the staging of the variable GROWTH using only its parents in the associated ALDAG. The staging tells us that for larger companies the probability of revenue change does not depend on other innovations, and it is the same as for medium-sized companies that invested in other innovations (stages  $v_7-v_9$ ). Medium-sized companies that did not invest in other innovations have the same probability of revenue change as small ones that did invest in other innovations (stages  $v_5-v_6$ ). Importantly, larger enterprises and medium-sized ones that invested in other innovations have a larger probability of increasing revenue (0.61). On the other hand, smaller companies that did not invest in other innovations are more likely to decrease their revenue since their probability of increasing is only 0.47.

An algorithm for constructing the dependence subtree is based on a simple variation of those given in the "Appendix". Dependence subtrees are extremely powerful since they allow us to visualize the dependence structure of GROWTH by means of the small tree in Fig. 10, without having to investigate the full staged tree having more than 100k leaves. Dependence subtrees are most useful when the associated ALDAG is sparse. They aim to represent the local dependence structure in the conditional probability of a node given its parents. Thus, although using a different type of representation based on coloring, they are in spirit equivalent to tree-parametrized CPTs that have been extensively studied in the literature (e.g. Hyttinen et al. 2018; Pensar et al. 2016; Talvitie et al. 2019).

## 7 Discussion

Staged trees are a flexible class of models that can represent highly non-symmetric relationships. This richness has the drawback that independences are often difficult to assess and visualize intuitively through its graph. This paper introduces methods that summarize the symmetric and non-symmetric relationships learned from data via structural learning by transforming the tree into a DAG. As a result, we introduced a novel class of graphs extending DAGs by labeling their edges. Our data applications showed the superior fit to data of such models as well as the information they can provide in real domains.

Datasets currently modeled with staged trees consist of at most 15–20 variables, as in the enterprise innovation application of Sect. 6.3. One of the difficulties in dealing with larger numbers of variables is the exponential growth of the size of the underlying event tree. However, the ALDAG paves the way for new methods to learn complex



**Fig. 10** The dependence subtree associated to the ALDAG in Fig. 8 for the variable GROWTH. The variable order is (EMP, INPD, GROWTH)

asymmetric dependencies which do not require the construction of the whole event tree. Such methods would remind already developed approaches that represent asymmetry in the CPTs of BNs using trees or decision graphs (e.g. Pensar et al. 2016). Different from standard methods, though, they would construct the edge set of the underlying DAG by taking into account asymmetric information in the data.

The new DAG edge labeling is based on the identification of the class of dependence. A different possibility would be to define a dependence index between any two variables, which measures how different their relationship is from total dependence/independence. By learning a staged tree from data, we could label the edges of a BN with such indexes. The definition of such models is the focus of current research.

This work provides a first criterion for reading any symmetric conditional independence from a staged tree. Algorithms to assess generic non-symmetric conditional independence statements still need to be developed. Here we have provided an intermediate solution to this problem by characterizing whether a non-symmetric independence exists. In future work, we plan to provide a conclusive solution to non-symmetric independence queries.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Declaration**

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence,

and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A Proof of Proposition 1

For the proof we use the following Lemma.

**Lemma 1** *Let  $T = (V, E, \theta)$  and  $T' = (V, E, \theta')$  be two staged trees with same tree  $(V, E)$ . We have that  $\mathcal{M}_T \subseteq \mathcal{M}_{T'}$  if and only if for each  $e, f \in E$ ,  $\theta'(e) = \theta'(f) \Rightarrow \theta(e) = \theta(f)$ .*

The result follows directly from the definition of  $\mathcal{M}_T$ .

The proof of Proposition 1 starts by showing that  $\mathcal{M}_T \subseteq \mathcal{M}_{G_T}$ . We have that  $\mathcal{M}_{G_T} = \mathcal{M}_{T_{G_T}}$  and let  $\theta'$  be the labeling of  $T_{G_T}$ . Both  $T$  and  $T_{G_T}$  are  $\mathbf{X}$ -compatible staged tree and thus share the same vertex and edge set. Suppose that for  $\mathbf{x}_{[i-1]}, \mathbf{y}_{[i-1]} \in \mathbb{X}_{[i-1]}$ ,  $\kappa(\mathbf{x}_{[i-1]}) \neq \kappa(\mathbf{y}_{[i-1]})$ . We define now a sequence of nodes  $\mathbf{x}_{[i-1]} = \mathbf{x}_{[i-1]}^0, \dots, \mathbf{x}_{[i-1]}^{i-1} = \mathbf{y}_{[i-1]}$  in  $\mathbb{X}_{[i-1]} \subseteq V$  as following,

$$\mathbf{x}_{[i-1]}^0 = \mathbf{x}_{[i-1]}, \quad x_j^h = \begin{cases} x_j^{h-1} & j > h \\ y_j & j \leq h \end{cases}.$$

Define  $h_0 = \min\{h \text{ s.t. } \kappa(\mathbf{x}_{[i-1]}^h) \neq \kappa(\mathbf{x}_{[i-1]}^{h-1})\}$ . We have that  $k(\mathbf{x}_{[i-1]}) = \dots = \kappa(\mathbf{x}_{[i-1]}^{h_0-1}) \neq \kappa(\mathbf{x}_{[i-1]}^{h_0})$  and thus, by construction of  $G_T$ ,  $(h_0, i) \in F_T$  that in turn implies  $\theta'(\mathbf{x}_{[i-1]}, x_i) \neq \theta'(\mathbf{y}_{[i-1]}, x_i)$  for all  $x_i \in \mathbb{X}_i$ .

We thus have  $\mathcal{M}_T \subseteq \mathcal{M}_{T_{G_T}} = \mathcal{M}_{G_T}$  by Lemma 1.

Assume now  $G = ([p], F)$  is a DAG with  $1, \dots, p$  as a topological ordering and such that  $\mathcal{M}_T \subseteq \mathcal{M}_G$ , then  $\mathcal{M}_T \subseteq \mathcal{M}_{T_G}$  and if  $\gamma$  is the labeling of  $T_G$  we have that  $\gamma(e) = \gamma(f) \Rightarrow \theta(e) = \theta(f)$  for each  $e, f \in E$  by Lemma 1. Let  $(k, i) \in F_T$  be an edge of  $G_T$ , then by construction there exist  $\mathbf{x}_{[i-1]}, \mathbf{x}'_{[i-1]} \in \mathbb{X}_{[i-1]}$  with  $x_j = x'_j$ ,  $j \neq k$  such that  $\kappa(\mathbf{x}_{[i-1]}) \neq \kappa(\mathbf{x}'_{[i-1]})$ . Hence,  $\gamma(\mathbf{x}_{[i-1]}, (\mathbf{x}_{[i-1]}, x_i)) \neq \gamma(\mathbf{x}'_{[i-1]}, (\mathbf{x}'_{[i-1]}, x_i))$  and it is easy to see that this implies  $(h, i) \in F$  by construction of  $T_G$ .

## Appendix B Algorithms' implementations

The implemented algorithms work with  $\mathbf{X}$ -compatible staged trees  $T = (V, E, \theta)$ . There are different ways to practically implement staged trees, here we use a representation similar to the R package `stagedtrees` (Carli et al. 2022). In particular, let  $\mathbf{X}$  be a random vector taking values in  $\mathbb{X} = \times_{i \in [p]} \mathbb{X}_i$  and assume there are total orders over each  $\mathbb{X}_i$ . Then  $\mathbb{X}_{[i]}$  has an induced lexicographic total order. The node labeling (or coloring)  $\kappa$  in the  $\mathbf{X}$ -compatible staged tree definition can be represented by a

---

**Algorithm 1** DAG to  $X$ -Compatible Stratified Staged Tree

---

**Require:** A DAG  $G = ([p], F)$  such that  $1, \dots, p$  is a topological order of  $G$ .

**Ensure:** The stage vectors  $s^1, \dots, s^{p-1}$  encoding an  $X$ -compatible stratified staged tree  $T$  such that  $\mathcal{M}_T = \mathcal{M}_G$ .

```

1: for  $i = 1$  to  $p - 1$  do
2:    $s^i = [1]$ 
3:   for  $j = 1$  to  $i$  do
4:     if  $j \in \Pi_{i+1}$  then
5:        $s^i = [s \times \mathbb{X}_j : s \in s^i]$ 
6:     else
7:        $s^i = [s \times [1, \dots, 1] : s \in s^i]$ 
8:     end if
9:   end for
10: end for
    
```

---

sequence of vectors of symbols  $s^1, s^2, \dots, s^{p-1}$  where  $s^j = (\kappa(v))_{v \in \mathbb{X}_{[j]}} \in \mathcal{C}^{|\mathbb{X}_{[j+1]}|}$  is the vector of coloring and  $\mathbb{X}_{[j+1]}$  is the set of nodes of  $T$  with depth  $j$ , ordered with respect to the induced lexicographic order. The vectors  $s^1, \dots, s^{p-1}$  represent the stages of the tree and thus we refer to them as *stage vectors*.

Algorithm 1 describes the pseudo-code for the conversion algorithm that takes as input a DAG over  $[p]$  and outputs an  $X$ -compatible staged tree. We assume that  $1, \dots, p$  is a topological ordering of the DAG nodes, if not a simple permutation is sufficient before applying the algorithm.

Algorithm 2 is an implementation of the conversion from staged trees to DAGs. The algorithm records additional information about the type of dependence between variables, so that it can be used to define the corresponding ALDAG.

Algorithm 3 is the pseudo-code of the context specific backward hill-climbing algorithm described in Sect. 5.3.1.

In the pseudo-code for Algorithms 2 and 3 we use the following notation:  $\text{vec}(A)$  is the column-wise vectorization of a matrix  $A$  and  $\text{mat}^{m,n}(a)$  is the column-wise  $(m, n)$ -matrix-filling such that  $\text{vec}(\text{mat}^{m,n}(a)) = a$  and  $\text{mat}^{m,n}(\text{vec}(A)) = A$ , where  $a$  is a vector of symbols of length  $mn$  and  $A$  is a matrix of symbols of dimensions  $(m, n)$ .

Both Algorithms 2 and 3 are based on the following observation. For every  $i \in [p - 1]$  consider the following  $i$  matrices of stage symbols obtained iteratively:

$$\begin{aligned}
 A_i &= \text{mat}^{m_i, n_i}(s^i) && \text{where } m_i = |\mathbb{X}_i|, n_i = \text{lengt}(s^i)/m_i \\
 A^j &= \text{mat}^{m_j, n_j}(\text{vec}(A_{j+1}^t)) && \text{where } m_j = |\mathbb{X}_j|, n_j = \text{lengt}(s^j)/m_j
 \end{aligned}$$

Then we have that for each  $j \leq i$  we can easily read the context, partial and local conditional independences between  $X_{i+1}$  and  $X_j$  from the matrix of stages  $A_j$ . Each row in the matrix  $A_j$  corresponds to a level of the variable  $X_j$  and each column corresponds to a context of the type  $x_{[i] \setminus j}$ . Thus, for example, if a column of  $A_j$  has all elements equal, it means that, for the particular context, the value of  $X_j$  does not affect the conditional probability for  $X_{i+1}$ ; hence  $X_{i+1}$  is conditional independent of  $X_j$  given the specific context. If a subset of a column has all equal elements, the

conditional independence is partial. And finally, if some elements of the matrix  $A_j$  are equal and belong to different columns then the independence is only local.

---

**Algorithm 2** X-Compatible Staged Tree to DAG
 

---

**Require:** An X-compatible staged tree  $T$  encoded with stage vectors  $s^1, \dots, s^{p-1}$ .

**Ensure:** A minimal DAG  $G = ([p], F)$  such that  $\mathcal{M}_T \subseteq \mathcal{M}_G$  and a labeling of its edges  $\psi$  defining an ALDAG.

```

1:  $G = (V, E = \emptyset)$ 
2: for  $i = 1$  to  $p - 1$  do
3:    $a = s^i$ 
4:   for  $j = i$  to  $1$  do
5:      $N = \text{length}(a)$ 
6:      $m = |\mathbb{X}_j|$  and  $n = N/m$ 
7:      $A = \text{mat}^{m,n}(a)$ 
8:      $c_k = |\{A_{u,k} : u \in [m]\}|, k \in [n]$ 
9:      $r_u = |\{A_{u,k} : k \in [n]\}|, u \in [m]$ 
10:    if  $\max\{c_k : k \in [n]\} > 1$  then
11:      #  $j$  is a parent of  $i + 1$ 
12:       $E = E \cup \{(j, i + 1)\}$ 
13:      # check the type of dependence
14:      if  $\min\{c_k : k \in [n]\} = m$  then
15:         $r = \sum_{i \in [m]} |\{A_{u,k} : k \in [n]\}|$ 
16:         $d = |\{A_{u,k} : u \in [m], k \in [n]\}|$ 
17:        if  $r \neq d$  then
18:           $\psi(j, i + 1) = \text{local}$ 
19:        end if
20:      else if  $\min\{c_k : k \in [n]\} = 1$  then
21:        # there is at least one context specific indep.
22:        if  $\exists k \in [n]$  s.t.  $2 < c_k < m$  then
23:          # there is also a partial indep.
24:           $\psi(j, i + 1) = \text{context/partial}$ 
25:        else
26:           $\psi(j, i + 1) = \text{context}$ 
27:        end if
28:      else
29:         $\psi(j, i + 1) = \text{partial}$ 
30:      end if
31:       $a = \text{vec}(A^t)$ 
32:    else
33:       $a = (A_{1,1}, \dots, A_{1,n})$  # all rows of  $A$  are equal, and  $j$  is not a parent of  $i + 1$ 
34:    end if
35:  end for
36: end for

```

---

**Algorithm 3** Context-specific backward hill-climb (CSBHC)

**Require:** A starting  $\mathbf{X}$ -compatible staged tree  $T$  encoded with stage vectors  $s^1, \dots, s^{p-1}$ ; a dataset  $\mathcal{D}$  of observations from  $\mathbf{X}$ ; The score  $f : (T, \mathcal{D}) \mapsto \mathbb{R}$  to be optimized.

**Ensure:** An  $\mathbf{X}$ -compatible staged tree

$f_{best} = f(T, \mathcal{D})$ , compute initial score

**for**  $i = 1$  to  $p - 1$  **do**

**repeat**

$a = s^i$  the stage vector of  $T$  for variable  $X_{i+1}$

$N = \text{length}(a) = \prod_{j \in [i]} |\mathbb{X}_j|$

**for**  $j = i$  to  $1$  **do**

$m = |\mathbb{X}_j|$  and  $n = N/m$

$A = \text{mat}^{m,n}(a)$

**for**  $\alpha$  column of  $A$  **do**

        let  $T'$  be the staged tree obtained from  $T$  by joining together the stages in  $\alpha$

**if**  $f(T', \mathcal{D}) > f_{best}$  **then**

$T_{best} = T'$

$f_{best} = f(T', \mathcal{D})$

**end if**

**end for**

$a = \text{vec}(A^t)$

**end for**

$T = T_{best}$

**until** no improvement in score is possible

**end for**

**References**

- Barclay LM, Hutton JL, Smith JQ (2013) Refining a Bayesian network using a chain event graph. *Int J Approx Reason* 54:1300–1309
- Barclay L, Hutton J, Smith J (2014) Chain event graphs for informed missingness. *Bayesian Anal* 9(1):53–76
- Boutilier C, Friedman N, Goldszmidt M, Koller D (1996) Context-specific independence in Bayesian networks. In: *Proceedings of the 12th conference on uncertainty in artificial intelligence*, pp 115–123
- Cano A, Gómez-Olmedo M, Moral S, Pérez-Ariza CB, Salmerón A (2012) Learning recursive probability trees from probabilistic potentials. *Int J Approx Reason* 53(9):1367–1387
- Carli F, Leonelli M, Riccomagno E, Varando G (2022) The R package stagedtrees for structural learning of stratified staged trees. *J Stat Softw* 102(6):1–30
- Carli F, Leonelli M, Varando G (2023) A new class of generative classifiers based on staged tree models. *Knowl-Based Syst* 268:110488
- Chickering DM, Heckerman D, Meek C (1997) A Bayesian approach to learning Bayesian networks with local structure. In: *Proceedings of 13th conference on uncertainty in artificial intelligence*, pp 80–89
- Collazo R, Görge C, Smith J (2018) Chain event graphs. *Chapmann & Hall, Boca Raton*
- Corander J, Hyttinen A, Kontinen J, Pensar J, Väänänen J (2019) A logical approach to context-specific independence. *Ann Pure Appl Logic* 170(9):975–992
- Cowell RG, Smith JQ (2014) Causal discovery through MAP selection of stratified chain event graphs. *Electron J Stat* 8(1):965–997
- Darwiche A (2009) *Modeling and reasoning with Bayesian networks*. Cambridge University Press, Cambridge
- Dawson RJ (1995) The “unusual episode” data revisited. *J Stat Educ* 3(3)
- Duarte E, Solus L (2021) Representation of context-specific causal models with observational and interventional data. [arXiv:2101.09271](https://arxiv.org/abs/2101.09271)
- Duarte E, Solus L (2023) A new characterization of discrete decomposable graphical models. *Proc Am Math Soc* 151(03):1325–1338
- Freeman G, Smith JQ (2011) Bayesian MAP model selection of chain event graphs. *J Multivar Anal* 102(7):1152–1165

- Friedman N, Goldszmidt M (1996) Learning Bayesian networks with local structure. In: Proceedings of the 12th conference on uncertainty in artificial intelligence, pp 252–262
- Geiger D, Heckerman D (1996) Knowledge representation and inference in similarity networks and Bayesian multinets. *Artif Intell* 82(1–2):45–74
- Görge C, Leonelli M, Smith J (2015) A differential approach for staged trees. In: European conference on symbolic and quantitative approaches to reasoning and uncertainty. Springer, pp 346–355
- Görge C, Bigatti A, Riccomagno E, Smith JQ (2018) Discovery of statistical equivalence classes using computer algebra. *Int J Approx Reason* 95:167–184
- Görge C, Leonelli M, Marigliano O (2022) The curved exponential family of a staged tree. *Electron J Stat* 16(1):2607–2620
- Højsgaard S, Lauritzen SL (2008) Graphical Gaussian models with edge and vertex symmetries. *J R Stat Soc Ser B* 70(5):1005–1027
- Hytinen A, Pensar J, Kontinen J, Corander J (2018) Structure learning for Bayesian networks over labeled DAGs. In: International conference on probabilistic graphical models, pp 133–144
- ISTAT (2014) Multiscopo ISTAT—Aspetti della vita quotidiana. UniData—Bicocca Data Archive, Milano. Codice indagine SN147. Versione del file di dati 2.0
- ISTAT (2015) Italian innovation survey 2010–2012. <http://www.istat.it/en/archive/87787>
- Jaeger M, Nielsen JD, Silander T (2006) Learning probabilistic decision graphs. *Int J Approx Reason* 42(1–2):84–100
- Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press, Cambridge
- Leonelli M, Varando G (2022) Highly efficient structural learning of sparse staged trees. In: International conference on probabilistic graphical models, pp 193–204
- Leonelli M, Varando G (2023) Context-specific causal discovery for categorical data using staged trees. In: International conference on artificial intelligence and statistics, pp 8871–8888
- Leonelli M, Varando G (2024a) Learning and interpreting asymmetry-labeled DAGs: a case study on COVID-19 fear. *Appl Intell* 54(2):1734–1750
- Leonelli M, Varando G (2024b) Structural learning of simple staged trees. *Data Min Knowl Disc*. <https://doi.org/10.1007/s10618-024-01007-0>
- Massam H, Li Q, Gao X (2018) Bayesian precision and covariance matrix estimation for graphical Gaussian models with edge and vertex symmetries. *Biometrika* 105(2):371–388
- Nyman H, Pensar J, Koski T, Corander J (2016) Context-specific independence in graphical log-linear models. *Comput Stat* 31(4):1493–1512
- Pensar J, Nyman H, Koski T, Corander J (2015) Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. *Data Min Knowl Discov* 29(2):503–533
- Pensar J, Nyman H, Lintusaari J, Corander J (2016) The role of local partial independence in learning of Bayesian networks. *Int J Approx Reason* 69:91–105
- Pensar J, Nyman H, Corander J (2017) Structure learning of contextual Markov networks using marginal pseudo-likelihood. *Scand J Stat* 44(2):455–479
- Poole D, Zhang NL (2003) Exploiting contextual independence in probabilistic inference. *J Artif Intell Res* 18:263–313
- Scutari M (2010) Learning Bayesian networks with the bnlearn R package. *J Stat Softw* 35(3):1–22
- Shen Y, Choi A, Darwiche A (2020) A new perspective on learning context-specific independence. In: International conference on probabilistic graphical models, pp 425–436
- Silander T, Leong T (2013) A dynamic programming algorithm for learning chain event graphs. In: Proceedings of the international conference on discovery science, pp 201–216
- Smith J, Anderson P (2008) Conditional independence and chain event graphs. *Artif Intell* 172(1):42–68
- Talvitie T, Eggeling R, Koivisto M (2019) Learning Bayesian networks with local structure, mixed variables, and exact algorithms. *Int J Approx Reason* 115:69–95
- Thwaites PA, Smith JQ (2015) A separation theorem for chain event graphs. [arXiv:1501.05215](https://arxiv.org/abs/1501.05215)
- Thwaites P, Smith JQ, Riccomagno E (2010) Causal analysis with chain event graphs. *Artif Intell* 174(12–13):889–909