Submitted to Bernoulli
arXiv: 1512.00209

Equivalence Classes of Staged Trees

CHRISTIANE GÖRGEN and JIM Q. SMITH

Department of Statistics
University of Warwick
Coventry CV4 7AL
United Kingdom
E-mail: c.gorgen@warwick.ac.uk; j.q.smith@warwick.ac.uk

In this paper we give a complete characterization of the statistical equivalence classes of CEGs and of staged trees. We are able to show that all graphical representations of the same model share a common polynomial description. Then, simple transformations on that polynomial enable us to traverse the corresponding class of graphs. We illustrate our results with a real analysis of the implicit dependence relationships within a previously studied dataset.

MSC 2010 subject classifications: Primary 60E05, 60K35; secondary 62E99.
Keywords: Algebraic Statistics, Chain Event Graphs, Probability Trees, Staged Trees.

1. Introduction

The *Chain Event Graph (CEG)* is a discrete statistical model based on a graphical description given by an event tree [22]. CEGs have now successfully led statistical inference in a whole range of domains [2, 5, 9, 25]. However, a formal analysis of the statistical properties of this class of models is long overdue.

In this paper, it will be most convenient to represent a CEG model by a corresponding *staged tree* [22]. From this colored graph we can read a parametrization rule given by the multiplication of transition probabilities along root-to-leaf paths. Two staged trees are said to be *statistically equivalent* if their parametrization rules parametrize the same model: see Section 2.

The study of these statistical equivalence classes is an important one. The first reason for this is computational: CEGs constitute a massive model space to explore. By identifying a single representative within an equivalence class and a priori selecting across these representatives rather than the full class, we can dramatically reduce the search effort across this space. The second reason concerns coherence: when adopting a Bayesian approach in model selection, [14] and others have argued that two statistically equivalent models (i.e. those always giving the same likelihood) should be given the same prior distribution over its parameters. To apply this principle, it is essential to know when two CEGs make the same distributional assertions. The third reason is inferential: just like a Bayesian network (BN), a CEG or staged tree has a natural causal extension [5, 24]. So, in particular, causal discovery algorithms can be applied to CEGs to elicit a putative causal ordering between various associated variables. A strong argument is that a

necessary condition for a causal deduction to be made from a given dataset is that this deduction is invariant to the choice of one representative within a statistical equivalence class. So again we need to be able to identify equivalence classes of a hypothesized causal CEG in order to perform these algorithms.

Now, unlike for BNs, where model representations making equivalent distributional assumptions can be elegantly characterized through their sharing the same *essential graph* [1, 14], sadly no such common representation is available for staged trees or CEGs. However, we show here that we can instead specify staged trees in terms of a nested polynomial representation. This then provides a natural algebraic index for a class of equivalent staged trees and an analogue of the essential graph. Because staged tree models include discrete BN models as a special case, our polynomial characterization also gives an alternative to the ansatz adopted by [10].

Our central theorem, presented in Section 3, is based on two main findings. First, the *interpolating polynomial* of a staged tree can capture certain context-specific independence structures that are invariant with respect to a class of graphical transformations we call *swaps*. These transformations are analogous to arc reversals sometimes applied to BN models [20]. Second, by substituting various monomial terms of the interpolating polynomial into single factors we can often simplify our representation to capture only its substantive structure. Within our development this corresponds to what we call here a *resize* operator on the staged tree. We show later that in the context of decomposable BNs, this operation is analogous for example to the transformation of a directed acyclic graph into a junction tree [15]. Swaps and resizes enable us to meaningfully incrementally traverse the class of statistically equivalent staged tree representations of a given model. We are able to show that between every two statistically equivalent staged trees there is a map which is a composition of these operators. Statistical equivalence classes of staged tree and CEG models are thus fully characterized through simple relationships between their interpolating polynomials.

We illustrate our methods by giving a full characterization of the statistical equivalence class and a putative causal interpretation of the staged tree representing the Christchurch Health and Development Study [8] in Section 4. We end the paper with a brief discussion.

2. Staged Tree Statistical Models

In this paper we study properties of parametric statistical models which are based on a graphical representation given by a probability tree [21, 22]. We will treat the probability tree not only as some easily interpretable picture but as a directed graphical model in its own right. To properly study equivalence classes of these models, we first need to tighten the formalism introduced in [22].

A finite graph $\mathcal{T} = (V, E)$ with vertex set V and edge set $E \subseteq V \times V$ is called a *tree* if it is connected and has no cycles [21]. In a *directed* tree, each edge $e = (v, v') \in E$ is a pair of ordered vertices. We call vertices $\mathrm{pa}(v) = \{v' \mid \text{there is } (v', v) \in E\}$ the *parents* of $v \in V$ and $\mathrm{ch}(v) = \{v' \in V \mid \text{there is } (v, v') \in E\}$ the set of *children* of a vertex $v \in V$. A

vertex $v_0 \in V$ without parents is called a *root* of the tree and vertices without children are called *leaves*. We use the term *root-to-leaf path* and the sybol λ for a directed and connected sequence of edges $E(\lambda) \subseteq E$ which emanate from a root and terminate in a leaf. We call a directed tree an *event tree* if all vertices except for one unique root have exactly one parent and each parent which is not a leaf has at least two children.

In a tree model (as defined below), every root-to-leaf path represents an atom in a given sample space and depicts one possible history of a unit in a population passing through the tree. Every vertex $v \in V$ denotes a situation that such a unit might find itself in during that progress, and every edge $e = (v, v') \in E$ denotes the possibility of passing from one situation v to the next v'. For any unit in the population there are always at least two possible unfoldings from every situation it might pass through.

We denote the set of all root-to-leaf paths of an event tree by $\Lambda(\mathcal{T})$. For fixed $v \in V$ we define a *vertex-centered event* as $\Lambda(v) = \big\{\lambda \in \Lambda(\mathcal{T}) \mid \text{there is } (\cdot, v) \in E(\lambda)\big\}$ and set $\Lambda(v_0) = \Lambda(\mathcal{T})$. In tree models, the set of all root-to-leaf paths going through one fixed vertex is the set of all atoms for which that situation happens. We call a vertex $v \in V$ together with its emanating edges $E(v) = \{(v, v') \in E \mid v' \in \mathrm{ch}(v)\}$ a *floret*, denoted $\mathcal{F}_v = (v, E(v))$. If v is a leaf then $E(v) = \emptyset$ and $\mathcal{F}_v$ is an empty floret. If the entire event tree is a single floret, it is called a *star*.

The directionality of an event tree induces a natural order on events (and florets) as follows: We say that $\Lambda(v)$ is *upstream* of $\Lambda(v')$ if and only if every root-to-leaf path $\lambda \in \Lambda(v) \cap \Lambda(v')$ is a sequence of edges containing $(v, \cdot)$ before $(v', \cdot)$. Tree models are therefore particularly useful if a model class needs to express a potential ordering of events rather than of random variables [22].

Definition 1 (Probability tree)**.** *Let $\mathcal{T} = (V, E)$ be an event tree with parameters $\theta(e) = \theta(v, v')$ associated to all edges $e = (v, v') \in E$. We call $\boldsymbol{\theta}_v = \big(\theta(e) \mid e \in E(v)\big)$ a* vector of floret parameters*.*

The pair $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ of tree graph and labels $\boldsymbol{\theta}_\mathcal{T} = (\boldsymbol{\theta}_v \mid v \in V)$ is called a probability tree *if every floret parameter vector lies inside a probability simplex, so $\sum_{e \in E(v)} \theta(e) = 1$ and $\theta(e) \in (0, 1)$ for all $v \in V$, $e \in E$. In probability trees, we call each parameter $\theta(e)$, $e \in E$, a* primitive probability*.*

Primitive probabilities can be thought of as (conditional) transition probabilities along root-to-leaf paths. Throughout, we assume these probabilities to be strictly positive in order to avoid various distracting technical issues concerning boundary cases.

We henceforth denote the product of all primitive probabilities along a root-to-leaf path $\lambda \in \Lambda(\mathcal{T})$ in a probability tree by

$$\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \prod_{e \in E(\lambda)} \theta(e) \tag{2.1}$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathcal{T}$ for short. It is straightforward to show that $\pi_{\boldsymbol{\theta},\mathcal{T}}$ is a strictly positive probability mass function. In particular, atomic probabilities sum to unity because of the floret sum-to-1 conditions in Definition 1.

Let $\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} = \big(\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) \mid \lambda \in \Lambda(\mathcal{T})\big)$ denote a vector of atomic probabilities represented by a probability tree. Following standard notation in algebraic statistics, we always think of a family of discrete probability distributions as a set of points. So let

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})} = \Big\{\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} \mid \boldsymbol{\theta} \in \bigtimes_{v\in V} \Delta^\circ_{\#E(v)-1}\Big\} \subseteq \Delta^\circ_{\#\Lambda(\mathcal{T})-1} \tag{2.2}$$

where $\Delta^\circ_{n-1} = \{\boldsymbol{p} \in \mathbb{R}^n \mid \sum_{i=1}^n p_i = 1 \text{ and } p_i \in (0,1) \text{ for all } i \in [n]\}$ denotes a probability simplex, $[n] = \{1, 2, \ldots, n\}$ [7]. We call the parametric statistical model in (2.2) a *(probability) tree model* and say that the elements in $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$ *factorize according to* $\mathcal{T}$. This terminology is analogous to BN models where distributions factorize according to an acyclic digraph [15].

Henceforth, we will call two probability tree representations $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ of the same model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})} = \mathbb{P}_{(\mathcal{S},\boldsymbol{\theta}_\mathcal{S})}$ *statistically equivalent*. We let the symbol $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ denote the set of all probability tree representations of $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$.

We can always identify the set of root-to-leaf paths of a probability tree with a finite space Ω via a bijection

$$\iota_\mathcal{T} : \ \Omega \to \Lambda(\mathcal{T}), \quad \omega \mapsto \big(e \mid e \in E(\iota_\mathcal{T}(\omega))\big) \tag{2.3}$$

which maps an *atom* or atomic event to a sequence of edges. Importantly, $\pi_{\boldsymbol{\theta},\mathcal{T}}$ then induces a measure $P_{\boldsymbol{\theta}} = \pi_{\boldsymbol{\theta},\mathcal{T}} \circ \iota_\mathcal{T}$ on Ω which does not depend on the graph $\mathcal{T}$. We will usually use the symbol $P_{\boldsymbol{\theta}}(\omega)$ to refer to a value in $(0,1)$ and $\pi_{\boldsymbol{\theta},\mathcal{T}}(\iota_\mathcal{T}(\omega))$ to refer to a symbolic product of parameters, $\omega \in \Omega$. To make this distinction, we also call $\pi_{\boldsymbol{\theta},\mathcal{T}}$ an *atomic monomial* rather than an atomic probability. So two statistically equivalent staged trees need to have the same underlying space Ω and the same distribution $P_{\boldsymbol{\theta}}$ over its atoms. Using (2.3), root-to-leaf paths with the same meaning (representing the same atom) can then be identified across different representations.

A tree model does not need to arise from an underlying set of problem variables. However, if it is naturally defined through the relationships between a set of pre-specified random variables then we can identify the state space of these variables with a set of root-to-leaf paths as in (2.3). This enables us for instance to represent a discrete BN by a probability tree.

Example 1. *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a probability tree with $n \in \mathbb{N}$ root-to-leaf paths and a probability mass function $\pi_{\boldsymbol{\theta},\mathcal{T}}$ as above. The vector $\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} \in \Delta^\circ_{n-1}$ can then take any value within the probability simplex and we call $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})} = \Delta^\circ_{n-1}$ a* saturated *tree model.*

Let $\mathcal{F} = (\{v_0\} \cup \mathrm{ch}(v_0), \{e_1, \ldots, e_n\})$ be a star with attached parameter vector $\boldsymbol{\theta}_\mathcal{F} = \big(\theta(e_i) \mid i \in [n]\big) \in \Delta^\circ_{n-1}$. Then $(\mathcal{F}, \boldsymbol{\theta}_\mathcal{F})$ is statistically equivalent to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ if the probabilities associated with the same atoms are identified for any choice of parameters: so $\theta(e_i) = \pi_{\boldsymbol{\theta},\mathcal{T}}(\iota_\mathcal{T}(\omega_i))$ for all $\iota_\mathcal{F}^{-1}(e_i) = \omega_i \in \Omega$ and every $i \in [n]$.

Probability trees are most interesting when two or more vectors of floret parameters take the same values, and the distributions $\pi_{\boldsymbol{\theta},\mathcal{T}}$ factorize according to a "colored" graph $\mathcal{T}$.

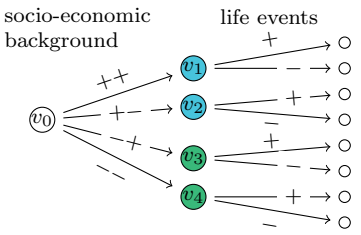


Figure 1: A staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$, simplified version taken from [2]. We label the edges by $+$ and $-$, corresponding to "high" and "low", respectively. See Example 2 for a discussion.

Definition 2 (Staged tree)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ with $\mathcal{T} = (V, E)$ be a probability tree. We define an equivalence relation which relates two vertices $v, w \in V$ if and only if their parameter vectors coincide $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$ up to a permutation of their components. Then v and w are said to be in the same* stage *and $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is said to be a* staged tree*.*

If no related vertices $v, w \in V$ are connected by a root-to-leaf path, $\Lambda(v) \cap \Lambda(w) = \emptyset$, we will call $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ square-free*.*

Whenever two vertices are in the same stage and a unit arrives at one of them, the transition probabilities to all children of that vertex will not depend on which of the two vertices the unit is actually in, and will thus not depend on the path that unit took to arrive in that situation. The transition probabilities from these stages are thus independent of upstream events. We always assign the same *color* to all vertices in the same stage. In this way, all modelling assumptions in staged tree models are coded purely graphically and are very easy to communicate [2, 23].

When having a preassigned set of random variables, setting floret parameter vectors equal to each other can be interpreted as specifying a set of *context-specific* conditional independences as in [3]. Models with these types of constraints are now widely used in BN modelling, especially when the domain of application is large.

Example 2. *The staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ depicted in Fig. 1 is a simplified detail of the graph analyzed in [2]. Here, every atom is represented by a root-to-leaf path with two edges: the first depicts the socio-economic background of a child, the second corresponds to a number of life events. See also Section 4 below.*

Information of the type "if we know the social status of a child's family, then their number of life events does not depend on their economic situation" can be embedded graphically by collecting the vertices $v_1, v_2 \in u_{blue}$ and $v_3, v_4 \in u_{green}$ in the blue- and green-colored stage, respectively. The primitive probabilities on the edges of the corresponding florets $\boldsymbol{\theta}_{v_1} = \boldsymbol{\theta}_{v_2}$ and $\boldsymbol{\theta}_{v_3} = \boldsymbol{\theta}_{v_4}$ are then identified.

3. A Polynomial Characterization of Staged Tree Models

In the development below we will interpret primitive probabilities or parameters which determine statistical models not as place holders for as yet undetermined numerical values but as elements of some formal symbolic (or algebraic) structure. This has been a hugely successful approach in the field of algebraic statistics [7, 18] which has until now not been explored for staged trees. We have found that the stage constraints on the parameter space of a staged tree can be easily translated into polynomial constraints on the atomic probabilities, and that the underlying model can thus be characterized as the solution set of a set of polynomial equations. This result is analogous to a well-known algebraic characterisation of BN models [10]. The process of translating the model-defining equations in (2.1) into equations which do not depend on a fixed parametrisation is straightforward but technical: see [11].

So in this paper we will instead use the idea of embedding model assumptions in an algebraic framework to develop an alternative and rather different approach which is more intuitive for staged trees. In particular, we define a polynomial below which can be used to both represent a staged tree model and to recover *constructively* all possible graph representations: a process not possible using only a set of defining equations.

3.1. Polynomial Equivalence and the Swap Operator

We first characterize a subclass of a class of statistically equivalent staged trees for which the following polynomial is invariant:

Definition 3 (Interpolating polynomial)**.** *Let* $\mathbb{P} = \{P_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ *denote a parametric statistical model on an underlying discrete space* Ω *with the property that every atomic probability* $P_{\boldsymbol{\theta}}(\omega)$ *is a monomial in the parameters* $\boldsymbol{\theta}$*,* $\omega \in \Omega$*. A* network polynomial *is of the form*

$$c_{g,\mathbb{P}}(\boldsymbol{\theta}) = \sum_{\omega \in \Omega} g(\omega) P_{\boldsymbol{\theta}}(\omega) \tag{3.1}$$

where g *is a function that determines a real coefficient for each monomial. If* $g = 1$*, then the symbolic sum of all atomic monomials is called an* interpolating polynomial *of the model* $\mathbb{P}$*.*

Network polynomials where $g = \mathbb{1}_A$ is chosen to be an indicator of an event $A \subseteq \Omega$ have been successfully used to answer probabilistic queries in BN models [6]. Different choices of g also relate the network polynomial to moment generating functions [19]. We have already demonstrated the efficacy of using these polynomials to calculate marginal and conditional probabilities in staged tree models [12], for sensitivity analysis in models with a multilinear parametrization [17] and for causal manipulations [13].

In the following, we will write $c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{\lambda \in \Lambda(\mathcal{T})} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda)$ for the interpolating polynomial $c_{1,\mathbb{P}(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ of a tree model represented by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

Remark 1. *When evaluating the network polynomial of a staged tree for a certain choice of parameters, we find that in a non-symbolic framework the polynomial*

$$c_{\mathbb{1}_A,\mathcal{T}}(\boldsymbol{\theta}) = \sum_{\lambda\in\Lambda(\mathcal{T})} \mathbb{1}_A(\lambda)\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \sum_{\lambda\in A} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = P_{\boldsymbol{\theta}}(\iota_{\mathcal{T}}^{-1}(A)) \tag{3.2}$$

is a function $(A,\boldsymbol{\theta}) \mapsto P_{\boldsymbol{\theta}}(\iota_{\mathcal{T}}^{-1}(A))$ *which maps an event* $A \subseteq \Lambda(\mathcal{T})$ *and a choice of parameters to the probability of that event.*

In a symbolic setting, we usually ignore sum-to-1 conditions and exploit only the formal structure of a polynomial. This is for instance beneficial when obtaining results like (3.2) from differentiation operations [12]. Note that because any choice of floret sum-to-1 conditions on an event tree will yield a probability distribution over the depicted root-to-leaf paths, these conditions can be ignored in the characterisation of equivalence classes of staged trees below and be imposed only after having found a model representation.

Definition 4 (Polynomial equivalence)**.** *Let* $(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})$ *and* $(\mathcal{S},\boldsymbol{\theta}_{\mathcal{S}})$ *be two staged trees with the same underlying space* Ω*. These staged trees are called* polynomially equivalent *if and only if they have the same edge labels and their network polynomials coincide formally* $c_{g,\mathcal{S}} = c_{g,\mathcal{T}}$ *for every choice of the function* g*.*

In general, polynomial equivalence is not necessary for statistical equivalence: see for instance Example 1 where two very different parametrizations can be used for the same model. However, we do have the following result:

Lemma 1. *Polynomial equivalence implies statistical equivalence.*

Proof. Set $g_\omega = \mathbb{1}_{\{\omega\}}$ to be the indicator function of an arbitrary atomic event $\omega \in \Omega$. Then polynomial equivalence of two staged trees $(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S},\boldsymbol{\theta}_{\mathcal{S}})$ implies termwise equality of the probability mass functions $c_{g_\omega,\mathcal{T}}(\boldsymbol{\theta}) = P_{\boldsymbol{\theta}}(\omega) = c_{g_\omega,\mathcal{S}}(\boldsymbol{\theta})$ by Remark 1. □

So all polynomially equivalent staged trees can be characterized as having the same interpolating polynomial plus an identification between their atoms. In square-free staged trees, the probability mass function $\pi_{\boldsymbol{\theta},\mathcal{T}} : \lambda \mapsto \prod_{e\in E(\lambda)} \theta(e)$ is formally injective: that is, the atomic monomials are pairwise different and we can uniquely identify atoms with monomials. So in these trees, the interpolating polynomial is sufficient to identify a class of polynomially equivalent staged trees. We henceforth use the symbol $[\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}}]^c \subseteq [\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}}]$ to denote a class of polynomially equivalent square-free staged trees which share the same interpolating polynomial c. Note that a staged tree $(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})$ is square-free if and only if $c_{\mathcal{T}}$ is linear in every indeterminate. We will restrict all further analysis to this class of models.

Remark 2. *The graph of a staged tree* $(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})$ *yields a way to parenthesize the associated interpolating polynomial as follows. For every floret* $\mathcal{F}_v$ *where* $v \in V$ *is the parent*

of a leaf, we sum all components of its parameter vector $\boldsymbol{\theta}_v$ and multiply the result by its parent label $\theta(\mathrm{pa}(v), v)$. We then sum the result over the parent's labels $\boldsymbol{\theta}_{\mathrm{pa}(v)}$. By repeating this until all floret parameter vectors are summed and $\mathrm{pa}(v) = v_0$, the interpolating polynomial can then be written in terms of a nested factorization

$$c_{\mathcal{T}}(\boldsymbol{\theta}) \;=\; \sum_{v_1 \in \mathrm{ch}(v_0)} \theta(v_0, v_1) \Big(\sum_{v_2 \in \mathrm{ch}(v_1)} \theta(v_1, v_2) \;\; \ldots \;\; \Big(\sum_{v_k \in \mathrm{ch}(v_{k-1})} \theta(v_{k-1}, v_k) \Big) \Big) \tag{3.3}$$

where the final index $k \in \mathbb{N}$ of every inner sum implicitly depends on the length of a root-to-leaf path $((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k))$. See Fig. 2 for an illustration.

The interpolating polynomials of discrete BN models admit a nested bracketing as in (3.3). The parameters in those polynomials are then *potentials* of a probability mass function and are normalized via the florets of an underlying tree representation. This type of representation of a polynomial provides a very efficient way to compute joint probabilities from marginals in a BN model [16] and comes for free when choosing a staged tree representation rather than an acyclic digraph.

Centrally, we can generalize the observation above to a new concept:

Definition 5 (Tree compatibility)**.** *Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ be a parameter vector, $d \in \mathbb{N}$. We call any polynomial* tree compatible *if it admits a representation of the form*

$$c(\boldsymbol{\theta}) \;=\; \sum_{\theta_1 \in A_1} \theta_1 \Big(\sum_{\theta_2 \in A_2(\theta_1)} \theta_2 \Big(\sum_{\theta_3 \in A_3(\theta_2)} \theta_3 \;\; \ldots \;\; \Big(\sum_{\theta_k \in A_k(\theta_{k-1})} \theta_k \Big) \Big) \Big) \tag{3.4}$$

where every $A_1, A_j(\theta_{j-1}) \subseteq \{\theta_1, \ldots, \theta_d\}$ has at least two elements, for $j \in [k]$ and $k \in \mathbb{N}$. We write $s(c(\boldsymbol{\theta}))$ for one fixed order of summation of the terms in $c(\boldsymbol{\theta})$ as above, and call this a tree-compatible factorization*.*

An important aspect of the result in Remark 2 is that it is reversible: not only can we easily read a polynomial from an event tree but we can also construct a tree graph from a tree-compatible factorization. In addition, all polynomially equivalent staged trees arise from a tree-compatible reordering of a given summation. Each of these gives a different representation within the same statistical equivalence class.

Proposition 1. *Let $\mathbb{P}$ be a discrete parametric model whose atomic probabilities are of monomial form and let $c = c_{1,\mathbb{P}}$ denote its interpolating polynomial. Then there exists a probability tree representation $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})} = \mathbb{P}$ if and only if c is tree compatible. The map $\mathfrak{c} : s(c(\boldsymbol{\theta})) \mapsto (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is invertible.*

Proof. Sufficiency of the first part of the claim is straightforward because the interpolating polynomial of a tree model is tree compatible by (3.3).

For necessity assume now the interpolating polynomial of a parametric model to be tree compatible and given by the factorization $s(c(\boldsymbol{\theta}))$ in (3.4). We construct a labelled graph

as follows: for every subsum of (3.4), draw a floret $\mathcal{F}_j = (v_j, \{e \mid \theta(e) = \theta_j \in A_j(\theta_{j-1})\})$ with one edge for every indeterminate in the sum and attach these indeterminates as labels, $j \in [k]$. Then partially order these florets by reversing the steps in Remark 2, such that θ_j is the parent label of the floret whose attached parameters are $A(\theta_j)$, for all $j \in [k]$. In this way, we construct a connected graph with no cycles—and hence a tree—whose leaf-floret edges are labelled by the innermost factors $A_k(\theta_{k-1})$ of $s(c(\boldsymbol{\theta}))$ and the root's edges by the outermost factors A_1. Since by definition every set $A_j(\theta_{j-1})$ has at least two elements, it follows that there are at least two edges in every floret. So the tree-compatible factorisations in (3.3) and (3.4) are componentwise equal and, multiplying out the brackets of $s(c(\boldsymbol{\theta}))$, we find that $c = c_{\mathcal{T}}$. Thus, we have constructed a labelled event tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. Then imposing sum-to-1 conditions on the constructed florets as in Definition 1 is consistent with the sum-to-1 conditions on atomic probabilities in the model. So $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is a probability tree which represents $\mathbb{P} = \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$.

By construction, the steps above are reversible. So the map which identifies a tree-compatible factorisation with a labelled tree is invertible. □

The proposition provides us with a powerful tool to determine whether a parametric model can be represented by a probability tree. This representation is a staged tree only if all constraints on the model are of the form $A_{i+1}(\theta_i) = A_{j+1}(\theta_j)$ for some $i \neq j$ in the notation of Definition 5.

The result above induces two natural streams of research. First, how can we check whether or not a given interpolating polynomial is tree compatible? We will discuss this issue at the very end of this work where we will also outline ideas for an algorithmic implementation.

The second question is: how do we infer all the possible orders of bracketing of a tree-compatible interpolating polynomial $c_{\mathcal{T}}$? Knowing this, we can use the map $\mathfrak{c}$ in Proposition 1 and the construction outlined in the proof to obtain all tree representations in $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c$. We will show how to do this below.

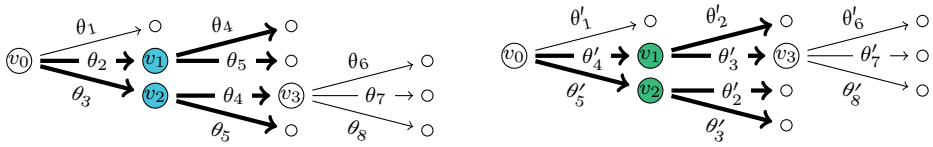
Clearly, a transformation between two tree-compatible factorizations of an interpolating polynomial is an application of the distributive property of addition and multiplication. These correspond to the following intuitive graph transformation.

We henceforth call a subgraph of a probability tree which is an event tree with inherited edge labels a *(probability) subtree*. We call a probability subtree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ a *twin* if it is of the following form: all root-to-leaf paths consist of exactly two edges and all children of its root are in the same stage u. This stage does not contain the root itself.

The interpolating polynomial of a twin $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$ can be written in the form

$$c_{\mathcal{T}_u}(\boldsymbol{\theta}) = \sum_{e \in E(v_0)} \theta(e) \Big(\sum_{e' \in E(v)} \theta(e') \Big) = \sum_{e' \in E(v)} \theta(e') \Big(\sum_{e \in E(v_0)} \theta(e) \Big) \tag{3.5}$$

where $v \in u$ is one representative of the stage $u = \mathrm{ch}(v_0)$ and v_0 is the root of the twin. By Proposition 1, there is a staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_u$ which is polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$: this is the one given by the second tree-compatible factorization in (3.5).



(a) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ with twin $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u$ thick depicted, $u = \{v_1, v_2\}$.

(b) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ with twin $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u$ thick depicted, $u = \{v_1, v_2\}$.

Figure 2: Two polynomially equivalent staged trees with the same indeterminates $\theta_i = \theta'_i$, $i = 1, \ldots, 8$, and interpolating polynomials $c_\mathcal{T}(\boldsymbol{\theta}) = \theta_1 + (\theta_2(\theta_4 + \theta_5) + \theta_4(\theta_6 + \theta_7 + \theta_8) + \theta_5)$ and $c_\mathcal{S}(\boldsymbol{\theta}') = \theta'_1 + \theta'_4(\theta'_2 + \theta'_3(\theta'_6 + \theta'_7 + \theta'_8)) + \theta'_5(\theta'_2 + \theta'_3))$ given in the respective tree-compatible factorization. The map $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is a swap.

Then, $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u$ is a subtree of a tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ which is polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and coincides with that tree everywhere except on $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u$.

Definition 6 (Swap)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a staged tree and $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ a twin around the stage u. Denote by $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u$ the staged tree which is polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u$ and let $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u \subseteq (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ as above. We will call the map $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ a* naïve swap *and call it a* swap *if $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is itself a staged tree.*

Figure 2 illustrates the definition above. We can see here that this operation does indeed "swap" the order of edges before and after the stage u. It is straightforward to show that edge-centred events on the root-edges of a twin are independent of those of edges ending in leaves. Our very plausible discovery is that for these independent events the order in which they are depicted in a tree is reversible within a statistical equivalence class, using the swap operator.

Whilst it is natural for swaps to change the floret structure, as explained below, naïve swaps might also violate stage structure. The simplest case is when the root of a twin is in a stage in the original tree, and a naïve swap rearranges that floret but not an identified one elsewhere in the graph. We henceforth call a composition of swaps for which floret parameter vectors are invariant a *floret-swap* and a composition of swaps which swaps all edges at a fixed distance from the root a *level-swap*. For instance, the swap in Fig. 2 is not a floret-swap because the root-vector $(\theta_1, \theta_2, \theta_3)$ is not a vector in $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$. Conversely, $(\theta_1, \theta_4, \theta_5)$ is not a floret parameter vector in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. This implies that $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ have different local sum-to-1 conditions on their primitive probabilities. By Lemma 1, both represent the same model. So even if the numerical value of say $\theta_1 = \theta(e_1)$ is different in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$—we have indicated this using labels $\theta_1 = \theta'_1$ in Fig. 2—via a renormalization it is still the probability of the event $\iota_\mathcal{T}^{-1}(\{\lambda \in \Lambda(\mathcal{T}) \mid e_1 \in E(\lambda)\}) \subseteq \Omega$ depicted by both graphs. The meaning of this parameter is thus unchanged and can be identified across different representations.

We can now obtain the following result, which enables us to both graphically and

algebraically move around a class of polynomially equivalent trees.

Proposition 2. *Two square-free staged trees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are polynomially equivalent if and only if there exists a finite composition of naïve swaps $\mathfrak{s}_1, \ldots, \mathfrak{s}_l$, $l \in \mathbb{N}$, for which $\mathfrak{s}_l \circ \mathfrak{s}_{l-1} \circ \ldots \circ \mathfrak{s}_1 : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is a swap.*

Proof. Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) = \mathfrak{c}(s_1(c(\boldsymbol{\theta})))$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) = \mathfrak{c}(s_2(c(\boldsymbol{\theta})))$ be polynomially equivalent staged trees with a common interpolating polynomial c and corresponding tree-compatible factorizations s_1 and s_2 as in (3.3). Here, $\mathfrak{c}$ denotes the map from Proposition 1. Clearly, one factorization $s_1(c(\boldsymbol{\theta}))$ is transformed into the other $s_2(c(\boldsymbol{\theta}))$ by applying the distributive law of $+$ and $\cdot$ a finite number of times. Hence, we can define a map $\tilde{s} : s_1(c(\boldsymbol{\theta})) \mapsto s_2(c(\boldsymbol{\theta}))$ performing these calculations on the subsums of c as in (3.5). Therefore,

$$\mathfrak{s} : \ (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \overset{\mathfrak{c}^{-1}}{\mapsto} s_1(c(\boldsymbol{\theta})) \overset{\tilde{s}}{\mapsto} s_2(c(\boldsymbol{\theta})) \overset{\mathfrak{c}^{-1}}{\mapsto} (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) \tag{3.6}$$

is a map which performs a finite number of swaps on the to $\tilde{s}$ corresponding twins and thus transforms $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ into $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$. □

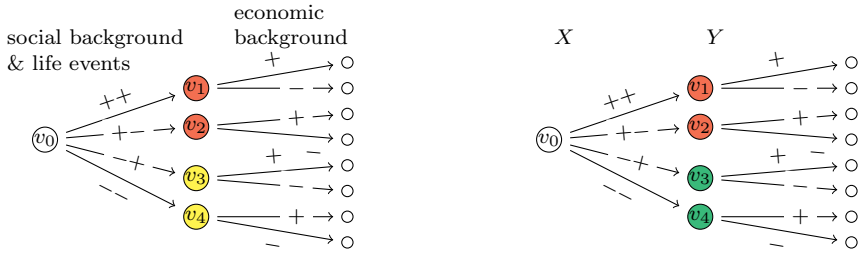
Thus, the polynomial equivalence class of a staged tree can be fully traversed by an algebraic resummation operation or, equivalently, by local graph transformations. Note that this operator is a close analogue to the *arc reversal* in BN models [20]. These, just like swaps, allow us to traverse the class of all graphical representations of the same model, while renormalizing (but not marginalizing) the associated probability mass function.

Example 3 (Example 2 continued)**.** *The staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ in Fig. 1 contains two twins: $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{blue}$ around the stage $u_{blue} = \{v_1, v_2\}$ and $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{green}$ around the stage $u_{green} = \{v_3, v_4\}$. Applying a level-swap on $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ which swaps both of these twins, we obtain a new tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_1$ depicted in Fig. 3a. In $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_1$, the edges emanating from the root now correspond to the random variable "social background and life events" rather than "social and economic background". This application of a swap corresponds to a renormalization of an underlying probability mass function $p(s, e, l) = p(s, e)p(l|s)$ to $p(s, l)p(e|s)$, for $s, e, l \in \{high, low\}$. This result can equivalently be achieved by applying an arc reversal on an alternative representation of this model in terms of a decomposable acyclic digraph.*

Unlike $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_1$, the staged tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_2$ in Fig. 3b where a swap has been applied only on $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{blue}$, cannot be straightforwardly identified with a BN model. In particular, the edges emanating from the root now correspond to a new random variable X "life events in low social background and economic situation in high social background", and the variable Y associated to leaf-edges changes accordingly:

$$X = \begin{cases} (S, L) & \text{if } S = 0 \\ (S, E) & \text{if } S = 1 \end{cases} \quad \text{and} \quad Y = \begin{cases} E|L & \text{if } S = 0 \\ L|E & \text{if } S = 1. \end{cases}$$

The example above provides a very simple illustration of how the statistical equivalence class of a staged tree (or CEG) can be so much larger than that of a BN. It



(a) An "arc reversal" staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$.

(b) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2$ with new associated random variables.

Figure 3: Two staged trees which are polynomially equivalent to the one from Example 2. See Example 3.

also demonstrates how staged trees can implicitly generate relationships between new random variables, constructed as functions of the original ones: possibly useful in later interpretative analysis. A more detailed discussion of this process is given in Section 4.

3.2. Statistical Equivalence and the Resize Operator

We have seen in Remark 1 that the network polynomial can be used to calculate probabilities of events represented by a staged tree. Thus, when leaving the symbolic framework and substituting values for the edge parameters, the interpolating polynomial is clearly invariant for a class of statistically equivalent staged trees. So in order to extend our characterization of polynomial equivalence classes to the whole statistical equivalence class, we will need to reparametrize between two given descriptions without violating the model assumptions. We define a second operator below which will again enable us to achieve this aim constructively.

We henceforth call the pair $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})' \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ a *subgraph* if it is a subtree with inherited edge labels whose root might have only one emanating edge, not necessarily two as required in an event tree.

Definition 7 (Resize)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree and let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})' \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a subgraph. We denote by $\mathfrak{r} : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ the map which transforms this subgraph into a floret $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ with labels $\boldsymbol{\theta}_{\mathcal{F}} = \big(\pi_{\boldsymbol{\theta},\mathcal{T}'}(\lambda') \mid \lambda' \in \Lambda(\mathcal{T}')\big)$, while leaving the remaining graph invariant. We call $\mathfrak{r}$ and its inverse $\mathfrak{r}^{-1}$* naïve resize *operators, and a* resize *if $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ is a staged tree.*

In terms of the atomic monomials, a naïve resize performs a substitution of products of primitive probabilities into degree 1 monomials. By construction, atomic probabilities of root-to-leaf paths are invariant under this operation: see also Example 1. However, $\mathfrak{r}$ is

not necessarily a well-defined map between two *staged* trees. The lemma below establishes useful criteria for a well-defined application of this operator.

Lemma 2. *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a staged tree and $l \in \mathbb{N}$. A composition of naïve resizes $\mathfrak{r} = \mathfrak{r}_l \circ \ldots \circ \mathfrak{r}_1$ applied to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is a resize if one of the following conditions is fulfilled:*

a) $\mathfrak{r}$ only acts on saturated subgraphs.
b) $\mathfrak{r}$ only acts on subgraphs which are polynomially equivalent to each other and whose vertices are not in the same stage as vertices outside these subgraphs.

Proof. a) Because the image $\mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) = (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ of a staged tree is a probability tree and because by assumption the stage sets of image and preimage coincide, clearly also $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) \in [\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ is a staged tree.

b) Because all subgraphs $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})', (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'' \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ that $\mathfrak{r}$ acts on are polynomially equivalent, they are also statistically equivalent: see Lemma 1. So after resizing, we identify the atomic probabilities $\pi_{\boldsymbol{\theta},\mathcal{T}'}(\lambda') = \pi_{\boldsymbol{\theta},\mathcal{T}''}(\lambda'')$ of subpaths λ', λ'' which have the same atomic monomial in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. Thus, the image $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) = \mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is a staged tree where the stages are given by these identified (formerly atomic now) primitive probabilities. □

Note that case (a) in Lemma 2 enables us to contract subgraphs which do not contain any stage information, so are in that sense not informative to the model. Analogous operations are often performed on BN models where the cliques of a decomposable model do not contain any conditional independence information and can hence be treated as a joint random variable without leaving the model class [15]. Case (b) enables us to directly identify atomic monomials of polynomially equivalent subgraphs rather than repeating stage equations edge by edge. Note that if these conditions are violated, then a naïve resize can take us out of the statistical equivalence class of a staged tree.

Lemma 3. *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a staged tree and $\mathfrak{r}$ a resize operator, possibly a composition of naïve resizes. Then $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $\mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ are statistically equivalent staged trees.*

This results follows immediately from the definition.

Finally, the resize in conjunction with the swap operator now enables us to traverse the whole equivalence class of a given staged tree.

Theorem 1. *Two square-free staged trees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are statistically equivalent if and only if there exists a map $\mathfrak{m} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ which is a finite composition of resizes and swaps.*

Proof. First let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ be statistically equivalent staged trees. Then all identified root-to-leaf paths $\lambda' = \iota_\mathcal{S}(\iota_\mathcal{T}(\lambda))$ have equal atomic probabilities, $\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \pi_{\boldsymbol{\theta}',\mathcal{S}}(\lambda')$. If the above equality holds in a formal sense for every $\lambda \in \Lambda(\mathcal{T})$ then $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are polynomially equivalent. In this case, Lemma 1 states that a map exists

between the two staged trees which is a composition of swaps, and thus proves the claim. If this is not the case, we denote by $\Lambda \subseteq \Lambda(\mathcal{T})$ the set of root-to-leaf paths in $\mathcal{T}$ whose atomic monomials do not coincide formally with the corresponding atomic monomials in $\mathcal{S}$. Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})' \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ denote a subtree of $\mathcal{T}$ for which $\Lambda \subseteq \Lambda(\mathcal{T}')$, and define analogously the corresponding $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})' \subseteq (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$. These are the subtrees which are not polynomially equivalent. We define two naïve resize operators, $\mathfrak{r}_\mathcal{T} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})' \mapsto (\mathcal{F}, \boldsymbol{\theta}_\mathcal{F})$ and $\mathfrak{r}_\mathcal{S} : (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})' \mapsto (\mathcal{F}, \boldsymbol{\theta}_\mathcal{F})$ which map those subtrees to the same floret. By Lemma 3, $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})'$, $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'$ and $(\mathcal{F}, \boldsymbol{\theta}_\mathcal{F})$ are statistically equivalent. Thus, there is a composition of resizes $\mathfrak{r} = \mathfrak{r}_\mathcal{S}^{-1} \circ \mathfrak{r}_\mathcal{T} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ between the statistically equivalent staged trees.

Now let $\mathfrak{m}$ be a transformation given by swaps and resizes between two staged trees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$. If $\mathfrak{m}$ is a composition of swaps, then Proposition 1 ensures polynomial equivalence, and thus statistical equivalence by Lemma 1. If $\mathfrak{m}$ is a composition of resizes, then Lemma 3 yields statistical equivalence. Clearly, also for the composition of both of these operators holds that $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $\mathfrak{m}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) = (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are statistically equivalent. The claim follows. □

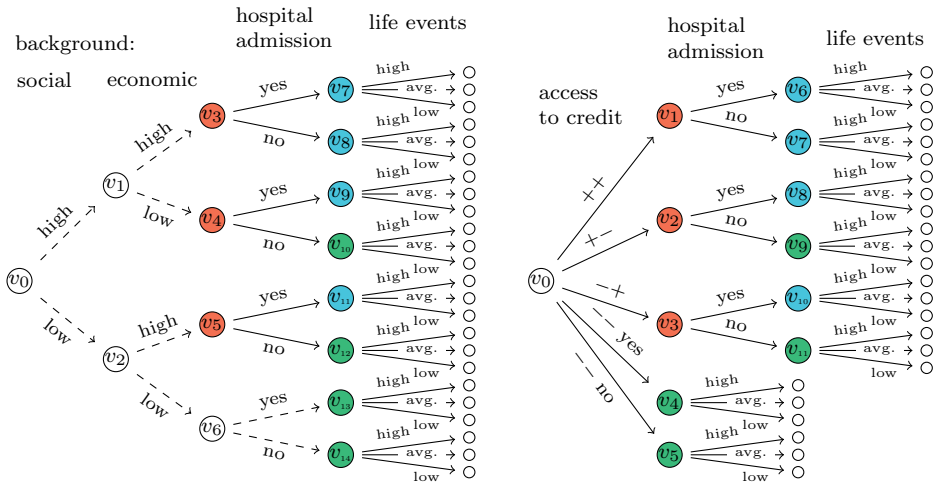
4. Analyzing a full Statistical Equivalence Class

We will now characterize properties of the statistical equivalence class of a staged tree inferred from a dataset. In particular, using the resize operator we will create new random variables describing the system and using the swap operator we will be able to give a putative causal interpretation to a depicted order of events.

A staged tree model for the Christchurch Health and Development Study (CHDS) [8] has been closely analyzed, e.g. in [2, 5], and has been used to describe the interplay of the social support, the economic situation, hospital admissions and possible life events (e.g. divorce, redundancy of a parent) of a group of children over a fixed period of time. The staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ in Fig. 4a was found using an MAP search [5]. We will now apply Theorem 1 to the statistical equivalence class $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ in order to enrich our understanding of the model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})}$ represented by that tree.

We first observe that there is a saturated subtree in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$, depicted by dotted lines in the figure. Because this does not contain twins, within the polynomial equivalence class there is thus an artificial order on the variables "social background" and "economic background". This order cannot be said to have been deduced from the model search. It is therefore helpful for us to transform $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ into the statistically equivalent staged tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ of Fig. 4b, using a resize operator as in Lemma 2(a).

The root's edges $e_i = (v_0, v_i)$, $i = 1, 2, \ldots, 5$, in this new tree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ can now be assigned a meaning different from the one in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. In particular, e_1, e_2 and e_3 correspond to "social background or economic status are high" and e_4 and e_5 to "both social background and economic status are low, hospital admission yes or no". Hence, children passing along e_1 are "from a wealthy background", along e_2 and e_3 are "from a moderately wealthy background" and along e_4 and e_5 are "from a poor background". From



(a) The MAP staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ from [5].

(b) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) \in [\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]$.

Figure 4: Two statistically equivalent staged trees for the CHDS data set.

the stage structure of $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ we can see that the probabilities of certain numbers of life events differ between wealthy and poor children. Interestingly, [4] names the *access to credit* as a possible monetary measurement of poverty. So being able to borrow from a social network or having own savings is a natural indicator of wealth. This gives some external support for moving from $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ to $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$, suggested from the results of our automated MAP search on the CHDS data.

We next analyze the polynomial equivalence class $[\mathcal{S}, \Theta_{\mathcal{S}}]^c$ for $c = c_{\mathcal{S}}$. There are five twins in this tree which have two children in the same stage. These are the ones where $v_1, v_2 \in u_{\text{red}}$, $v_1, v_3 \in u_{\text{red}}$, $v_2, v_3 \in u_{\text{red}}$, $v_4, v_5 \in u_{\text{green}}$ and $v_6, v_7 \in u_{\text{blue}}$ have the same parent and are in the same stage, respectively. For most of these, an application of the swap operator would be naïve and violate the stage constraints in the tree. In fact, there are only two swaps which yield a staged tree: the composition which performs a floret-swap on v_1, v_2 and v_3 simultaneously and the swap on v_6, v_7. These change the order between access to credit and hospital admission for (moderately) wealthy children, and between hospitalisation and life events for wealthy children. It would thus be spurious to assert a potentially causal or chronological order on these events on the basis of the MAP search.

There is no staged tree in the polynomial equivalence class of $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ that would allow for the total order "life events before hospitalisation". This is because no composition of the swaps on the twins can form a level-swap on $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. So a model which treats life events as an explanatory variable of the response variable hospital admission as in the study [2] is less supported by the data than one treating hospitalisation as an explanatory variable of life events as in [5]. Note that no deductions about an ordering of variables

would have been possible within the original BN representation of the data because the MAP model turns out to be decomposable. This demonstrates that the extra structure of the staged tree enables us to draw out new potential causal hypotheses that could not be discovered when using more conventional graphical methods.

5. Discussion

In this paper we have been able to show that a characterization of staged trees in terms of their interpolating polynomials provides an elegant way to fully analyze statistical equivalence classes of models represented by such trees.

For a future implementation of these results it is important to note that the number of tree-compatible factorizations of an interpolating polynomial is usually enormous. For instance, a naïve count of the elements in the class analyzed in Section 4 reveils nearly one thousand elements. This is because every polynomial equivalence class contains $2^{\#\text{twins}}$ elements arising from naïve swaps, each of which can be combined with resize operations as outlined above. Of course we would then need to check how many of these elements actually correspond to staged trees. This will normally reduce the number of amenable tree-compatible factorizations significantly: for instance in that example there would be $2^5 = 32$ naïve representations, only four of which are staged.

The polynomial-based approach we develop here provides a most promising foundation for developing an efficient search across this class using computational algebra. In particular, given any discrete model with multilinear parametrization, we note that every possible tree-compatible factorization of its interpolating polynomial arises from a certain nested order of common divisors of terms in the polynomial. This nesting is naturally reflected in what is called the *primary decomposition* of the ideal spanned by all terms in the polynomial. Every element of such a decomposition which is spanned by degree-one indeterminates will then provide a set of putative root labels of a corresponding tree representation; and if there are no such candidates then the multilinear model would not be a staged tree model. Investigating subnestings of ideals and running a search over putative root-labels obtained from ideal decomposition is much faster than for instance an exhaustive search over all possible nested factorizations of a polynomial. This is because in such an unstructured search we would have in the order of 2^d choices of root labels, one for each subset of labels, where d is the number of indeterminates. Ideal decomposition provides us with much fewer candidate nestings and also provides an elegant way of a priori excluding certain naïve representations which are not staged. As a consequence, we can employ a vast range of freely available software to design algorithms which can efficiently traverse a polynomial equivalence class—so an algorithmic implementation for the swap operator is within reach. As for the resize operator, the computational-algebra algorithm suggested here will then need to be enhanced by a command which allows us to leave a fixed algebraic framework (given by the chosen parametrization) and to substitute terms using the requirements for having non-naïve resizes we discovered in this paper. The design of these algorithms is outside the scope of this publication.

In a second step, once an algorithm as sketched above is in place and we have software

for applying swaps and resizes on a staged tree, we are ready to use these results in inference and model selection. In particular, when data is available we can now develop methods to score an interpolating polynomial (rather than a tree graph) directly and use the methods proposed here to then traverse the whole statistical equivalence class purely algebraically. An important direction for future work is to demonstrate how interpolating polynomials can thus be used in the analysis of tree-based causality, in comparison to analogous concepts developed for BN models.

Acknowledgements

Christiane Görgen was supported by the EPSRC grant EP/L505110/1 and Jim Q. Smith was supported by the EPSRC grant EP/K039628/1.

References

[1] Andersson, S. A., Madigan, D. and Perlman, M. D. (1997). A characterization of Markov Equivalence Classes for Acyclic Digraphs. *Ann. Statist.* **25** 505–541.

[2] Barclay, L. M., Hutton, J. L. and Smith, J. Q. (2013). Refining a Bayesian network using a Chain Event Graph. *Internat. J. Approx. Reason.* **54** 1300–1309.

[3] Boutilier, C., Friedman, N., Goldszmidt, M. and Koller, D. (1996). Context-specific independence in Bayesian networks. In *Uncertainty in Artificial Intelligence (Portland, OR, 1996)* 115–123. Morgan Kaufmann, San Francisco, CA.

[4] Coudouel, A., Hentschel, J. S. and Wodon, Q. T. (2002). *Poverty Measurement and Analysis.* In *A sourcebook for Poverty Reduction Strategies: Core techniques and cross-cutting issues* 27–74. The World Bank.

[5] Cowell, R. G. and Smith, J. Q. (2014). Causal discovery through MAP selection of stratified Chain Event Graphs. *Electron. J. Stat.* **8** 965–997.

[6] Darwiche, A. (2003). A differential approach to inference in Bayesian networks. *J. ACM* **50** 280–305 (electronic).

[7] Drton, M., Sturmfels, B. and Sullivant, S. (2009). *Lectures on algebraic statistics. Oberwolfach Seminars* **39**. Birkhäuser Verlag, Basel.

[8] Fergusson, D. M., Horwood, L. J. and Shannon, F. T. (1986). Social and family factors in childhood hospital admission. *Journal of Epidemiology and Community Health* **40** 50–58.

[9] Freeman, G. and Smith, J. Q. (2011). Bayesian MAP model selection of Chain Event Graphs. *J. Multivariate Anal.* **102** 1152–1165.

[10] Geiger, D., Meek, C. and Sturmfels, B. (2006). On the toric algebra of graphical models. *Ann. Statist.* **34** 1463–1492.

[11] Görgen, C. (2017). An algebraic characterisation of staged tree models. PhD thesis, University of Warwick, Department of Statistics.

[12] Görgen, C., Leonelli, M. and Smith, J. Q. (2015). A Differential Approach for Staged Trees. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Proceedings. Lecture Notes in Artificial Intelligence* 346–355. Springer.

[13] Görgen, C. and Smith, J. Q. (2016). A differential approach to causality in staged trees. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models. JMLR Workshop and Conference Proceedings* **52** 207–215.
[14] Heckerman, D. (1998). *A Tutorial on Learning with Bayesian Networks.* In *Learning in Graphical Models* 301–354. The MIT Press.
[15] Lauritzen, S. L. (1996). *Graphical models. Oxford Statistical Science Series* **17**. The Clarendon Press, Oxford University Press, New York.
[16] Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *J. Roy. Statist. Soc. Ser. B* **50** 157–224. With discussion.
[17] Leonelli, M., Görgen, C. and Smith, J. Q. (2015). Sensitivity analysis, multilinearity and beyond. Pre-print available from *arXiv:1512.02266 [cs.AI].*
[18] Pistone, G., Riccomagno, E. and Wynn, H. P. (2001a). *Algebraic Statistics. Monographs on Statistics and Applied Probability* **89**. Chapman & Hall/CRC.
[19] Pistone, G., Riccomagno, E. and Wynn, H. P. (2001b). Gröbner bases and factorisation in discrete probability and Bayes. *Stat. Comput.* **11** 37–46.
[20] Schachter, R. D. (1988). Probabilistic Inference and Influence Diagrams. *Operations Research* **36** 589–605.
[21] Shafer, G. (1996). *The Art of causal Conjecture.* MIT Press, Cambridge.
[22] Smith, J. Q. and Anderson, P. E. (2008). Conditional independence and Chain Event Graphs. *Artificial Intelligence* **172** 42–68.
[23] Smith, J. Q., Görgen, C. and Collazo, R. A. (2017). *Chain Event Graphs.* In preparation for Chapman & Hall.
[24] Thwaites, P. (2013). Causal Identifiability via Chain Event Graphs. *Artificial Intelligence* **195** 291–315.
[25] Thwaites, P. A., Smith, J. Q. and Cowell, R. G. (2008). Propagation using Chain Event Graphs. In *Proceedings of the 24th Conference on Uncertainty in Artifical Intelligence* 546–553.