International Journal of Approximate Reasoning 95 (2018) 167–184

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Discovery of statistical equivalence classes using computer algebra

Christiane Görgen [a,*], Anna Bigatti [b], Eva Riccomagno [b,c], Jim Q. Smith [d,e]

[a] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany
[b] Dipartimento di Matematica, Università degli Studi di Genova, 16146 Genova, Italy
[c] Institute of Intelligent Systems for Automation, National Research Council, Italy
[d] Department of Statistics, University of Warwick, Coventry CV5 7AL, UK
[e] The Alan Turing Institute, British Library, 96 Euston Road, NW1 2DB London, UK

ARTICLE INFO

Article history:
Received 26 May 2017
Received in revised form 2 October 2017
Accepted 3 January 2018
Available online 13 February 2018

Keywords:
Graphical models
Staged tree models
Computer algebra
Ideal decomposition
Algebraic statistics

ABSTRACT

Discrete statistical models supported on labeled event trees can be specified using so-called interpolating polynomials which are generalizations of generating functions. These admit a nested representation which is a notion formalized in this paper. A new algorithm exploits the primary decomposition of monomial ideals associated with an interpolating polynomial to quickly compute all nested representations of that polynomial. It hereby determines an important subclass of all trees representing the same statistical model. To illustrate this method we analyze the full polynomial equivalence class of a staged tree representing the best fitting model inferred from a real-world dataset.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

Families of finite and discrete multivariate models have been extensively studied, including many different classes of graphical models [1,2]. Because these families of probability distributions can often be expressed as polynomials – or collections of vectors of polynomials – this has spawned a deep study of their algebraic properties [3–5]. These can then be further exploited using the discipline of computational commutative algebra and computer algebra software such as `CoCoA` [6,7] which has proved to be a powerful though somewhat neglected tool of analysis.

In this paper, we demonstrate how certain computer algebra techniques – especially the primary decomposition of ideals – can be routinely applied to the study of various finite discrete models. Throughout we pay particular attention to an important class of graphical models based on probability trees and called *staged trees* or *chain event graph* models [8]. These contain the familiar class of discrete (and context-specific) Bayesian networks as a special case. In particular, Görgen and Smith [9] gave a mathematical way of determining the statistical equivalence classes of staged tree models but did not give algorithms to actually find these. Here we use computer algebra in a novel way to systematically find a staged tree representation of a given family – if it indeed exists – and to uncover statistically equivalent staged trees in an elegant, systematic and useful way. This is an extensions of the techniques developed by Andersson et al. [2] and others to determine Markov-equivalence classes of Bayesian networks where, instead of algebra, graph theory was used as a main tool.

* Corresponding author.
E-mail addresses: goergen@mis.mpg.de (C. Görgen), bigatti@dima.unige.it (A. Bigatti), riccomagno@dima.unige.it (E. Riccomagno), j.q.smith@warwick.ac.uk (J.Q. Smith).

https://doi.org/10.1016/j.ijar.2018.01.003
0888-613X/© 2018 Elsevier Inc. All rights reserved.

So our methodology supports a new analysis of a very general but fairly recent statistical model class in a novel algebraic way and serves as an illustration of how more generally computer algebra can be a useful tool not only to the study of conventional classes of graphical model but other families of statistical model as well.

2. Staged trees and interpolating polynomials

2.1. Labeled event trees and staged trees

In this work we will exclusively consider graphs which are trees, so those which are connected and without cycles. We first review the theory of staged trees which represent interesting and very general discrete models in statistics [8].

Definition 1 *(Labeled event trees).* Let $T = (V, E)$ be a finite directed rooted tree with vertex set V and edge set $E \subseteq V \times V$. We denote the root vertex of T by v_0.

The tree T is called an *event tree* if every vertex $v \in V$ has either no, two or more than two emanating edges. For $v \in V$, let $E_v = \{(v, w) \mid w \in V\} \cap E$ denote the set of the edges emanating from v. The pair (v, E_v) is called a *floret*.

Let Θ be a non-empty set of symbol/labels and let a function $\theta : E \longrightarrow \Theta$ be such that for any floret (v, E_v) the labels in $\theta(E_v)$ are all distinct. We call $\theta(E_v)$ the *floret labels* of v and denote this set by θ_v. The pair $\mathcal{T} = (T, \theta)$ of graph and function is called a *labeled event tree*. When θ takes values in $(0, 1)$ and $\sum_{e \in E_v} \theta(e) = 1$, $\mathcal{T}$ is called a *probability tree*.[1]

For $v \in V$, the *labeled subtree rooted in* v is $\mathcal{T}_v = (T', \theta')$, where T' is the largest subtree of T rooted in v, and θ' is the restriction of θ to the edges in T'.

For any leaf $v \in V$, so for any vertex with no emanating edges, we trivially have that $E_v = \emptyset$, and hence $\theta_v = \emptyset$.

Labeled event trees are well-known objects in probability theory and decision theory where they are used to depict discrete unfoldings of events. The labels on edges of a probability tree then correspond to transition probabilities from one vertex to the next and all edge probabilities belonging to the same floret sum to unity. See [10] for the use of probability trees in probability theory and causal inference, and see for instance [11] for how such a tree representation can be used in computational statistics.

As an illustration consider a particle moving through a tree. It moves from vertex to vertex according to a Markov process whose states are the non-leaf vertices of the tree and whose transition probabilities are the edge labels. The root-to-leaf paths are the elementary events and an event of interest could thus be a transition through an internal node or through a set of nodes or it could be the arrival to a leaf node. Given the skeleton of the tree, the model is described in terms of the transition probabilities, i.e. the edge labels. It is assumed that the particles move across the tree independently from each other. Typically and in Section 5, an observation of the process is a recording of arrivals to the leaves of a particle starting from the root.

In this paper, we generally do not require the labels on a labeled event tree to be probabilities.

Definition 2 *(Staged trees).* A labeled event tree $\mathcal{T} = (T, \theta)$, with $T = (V, E)$, is called a *staged tree* if for every pair of vertices $v, w \in V$ their floret labels are either equal or disjoint, $\theta_v = \theta_w$ or $\theta_v \cap \theta_w = \emptyset$. A *stage* is a set of vertices with the same floret labels.

In illustrations of staged trees, all vertices in the same stage are usually assigned a common *color*: compare Fig. 1. Staged trees were first defined as an intermediate step to building *chain event graphs* as graphical representations for certain discrete statistical models [12]. Every chain event graph is uniquely associated to a staged tree and vice versa. In this way, the graphical redundancy of staged trees can be avoided, and elegant conjugate analyzes can be applied to staged tree models [13–17]. In particular, every discrete and context-specific Bayesian network can alternatively be represented by a staged tree where stages indicate equalities of conditional probability vectors. We give examples of this later in the text.

For the development in this paper it is important to observe that staged trees with labels evaluated as probabilities are always also probability trees. This is however not the case for all labeled event trees because sum-to-1 conditions imposed on florets can be contradictory. See also Examples 1 and 8 below.

Example 1. Fig. 1(a) shows a staged tree where all blue-colored vertices are in the same stage. Fig. 1(b) depicts a staged tree where the two green vertices are in the same stage. Fig. 1(c) shows a labeled event tree which is not staged because the floret labels of the two black vertices are neither equal nor disjoint.

2.2. Network polynomials and interpolating polynomials

We next define a polynomial associated to a labeled event tree which is the key tool used in this paper: see also [9].

[1] We should say more precisely: when the symbols $\theta(e)$ are evaluated in $(0, 1)$ for all $e \in E$.

Download English Version:

https://daneshyari.com/en/article/6858809

Download Persian Version:

https://daneshyari.com/article/6858809

Daneshyari.com