Learning Discrete Partially Directed Acyclic Graphical Models in Multitype Branching Processes

Pierre Fernique, *University of Montpellier 2, I3M and CIRAD, UMR AGAP and Inria, Virtual Plants*, pierre.fernique@inria.fr
Jean-Baptiste Durand, *University of Grenoble Alpes, Laboratoire Jean Kutzmann and Inria, Mistis*, jean-batiste.durand@imag.fr
Yann Guédon, *CIRAD, UMR AGAP and Inria, Virtual Plants*, guedon@cirad.fr

Abstract. We address the inference of discrete-state models for tree-structured data. Our aim is to introduce parametric multitype branching processes that can be efficiently estimated on the basis of data of limited size. Each generation distribution within this macroscopic model is modeled by a partially directed acyclic graphical model. The estimation of each graphical model relies on a greedy algorithm for graph selection. We present an algorithm for discrete graphical which is applied on multivariate count data. The proposed modeling approach is illustrated on plant architecture datasets.

Keywords. Partially directed graphical model, graph selection, multivariate distribution, tree pattern, branching process, plant architecture modeling, multivariate count data

1 Introduction

We consider discrete-state stochastic processes indexed by a rooted tree. Our aim is to introduce parametric models that can be efficiently estimated on the basis of data of limited size and that are easily interpretable. These models rely on local dependency assumptions between parent and child vertices and belong to the family of multitype branching processes (MTBPs). In our practical setting of plant architecture analysis, the combinatorics induced by the variable and high number of child vertices in each state induces an inflation in the number of model parameters. We thus introduce parametric MTBPs incorporating parsimonious graphical models for each generation distribution.

Data of interest are tree-indexed sets $\boldsymbol{x} = (x_t)_{t \in \mathcal{T}}$ where $\mathcal{T} \subset \mathbb{N}$ is the set of vertices of a rooted tree graph $\tau = (\mathcal{T}, \mathcal{A})$ and $\mathcal{A} \subset \mathcal{T} \times \mathcal{T}$ the set of directed edges representing lineage

relationships between vertices. By convention, the root of the tree graph has index 0. Let $x_t \in \mathcal{V} = \{0, \ldots, K-1\}$ denote the label of vertex t. Let $pa\,(.)$ denote the parent of a vertex, $ch\,(.)$ the children set of a vertex, $an\,(.)$ the ancestor set of a vertex and $de\,(.)$ the descendant set of a vertex. These notations also apply to set of vertices – see [8] for graph terminology. We here assume that x_t (resp. $\boldsymbol{x}$, τ) is the outcome of a discrete random variable X_t (resp. discrete random vector $\boldsymbol{X}$, random rooted tree T).

MTBPs are based on local dependency assumptions between parent and child vertices, more precisely on the following Markovian property – children are independent of their non-descendants given their parent

$$\forall t \in \mathcal{T},\ \boldsymbol{X}_{ch(t)} \perp\!\!\!\perp \boldsymbol{X}_{\mathcal{T}\setminus de(t)} \big| X_{pa(t)}\ ,$$

and a permutation invariance property – see [6] for details – in order to obtain a more parsimonious model. As a consequence, the joint distribution can be factorized as follows

$$P\left(T = (\mathcal{T}, \mathcal{A})\,, \boldsymbol{X} = \boldsymbol{x}\right) \propto P[X_0 = x_0] \prod_{t \in \mathcal{T}} P\left(\boldsymbol{N}_t = \boldsymbol{n}_t | X_t = x_t\right), \tag{1}$$

where $\boldsymbol{N}_t | X = x_t$ is the discrete random vector of the number of children of t in each state given x_t. Therefore the outcome to model is a discrete random vector $\boldsymbol{N}_t$ for each vertex

$$\boldsymbol{n}_t = \left(|\{s \in ch\,(t) | X_s = k\}|\right)_{k \in \mathcal{V}}\,,$$

MTBPs are thus specified by K discrete multivariate distributions called generation distributions.

In the case where x_t is not the outcome of a discrete random variable X_t but more generally of a random vector mixing discrete and continuous random variables, an extension of the hidden Markov tree model used for the modeling of plant growth [3] is considered. Such models combine a non-observable state process $\boldsymbol{S} = (S_t)_{t \in \mathcal{T}}$ with the observed random vector $\boldsymbol{X}$ and the model is estimated using an EM algorithm. Once the model has been estimated, a restoration algorithm is applied to obtain the optimal state tree.

A MTBP is specified by K generation distributions. In order to have interpretable results, we propose to focus on a family of multivariate discrete generation distributions such that :

- children states that tend to appears simultaneously or on the contrary to be incompatible can be identified,
- parametric distributions can be used since the direct estimation of probability masses on the basis of multivariate count histograms is unreliable except for very large data sets.
- these multivariate parametric distributions can have zero-inflated and right-skewed marginals, so that multivariate Gaussian distributions are not appropriate.

To achieve this goal, an approach based on probabilistic graphical models [8] to represent the conditional independence relationships for each generation distribution is considered. Three kinds of graphical models are usual: undirected (UG), directed acyclic (DAG), and partially directed acyclic graphical (PDAG) model.

Methods for graph identification were proposed for UGs, using either frequencies to directly estimate probability masses (so-called nonparametric estimation) or mutual information – see

[11] and references therein. Under a multivariate Gaussian distribution assumption, an approach based on a L_1 penalization (Lasso) was proposed in [5], with some extension to Poisson distributions and more generally to GLMs [14].

Specific models and methods were developed for DAGs. Most methods for graph identification in DAGs are based on exploring the set of possible graphs using some heuristic (e.g. hill climbing [1]) and by scoring the visited graphs (e.g. using BIC), the graph with highest score being eventually selected – see [8] for a review.

The case of PDAGs, generalization of UGs and DAGs such as both marginal independence relationships and cyclic dependencies between quadruplets of variables (at least) can be represented, has been considered less often in the literature. A family of such models was proposed using conditional Gaussian distributions, but the problem of graph identification was not addressed [2]. Lee & Hastie [10] addressed the problem of graph identification in graphical models with both continuous and discrete random variables, but in a restrictive setting of UGs with conditional Gaussian and multinomial distributions.

2 Discrete PDAG modeling and learning

Undirected [14] and directed acyclic [8] graphical models incorporating univariate distribution (binomial, Poisson and negative binomial) and corresponding regressions have been previously studied. We here propose an extension to PDAGs based on an enlarged family of discrete parametric distributions incorporating multivariate generalizations of the classical univariate discrete parametric distributions: multinomial, negative multinomial and multivariate Poisson [7] distributions and corresponding regressions.

The class of considered PDAGs is such that the joint distribution factorizes as [9]

$$P(\boldsymbol{N} = \boldsymbol{n}) = \prod_{c \in \mathcal{C}} P(\boldsymbol{N}_c = \boldsymbol{n}_c | \boldsymbol{N}_{\mathrm{pa}(c)} = \boldsymbol{n}_{\mathrm{pa}(c)}), \tag{2}$$

where $\mathcal{C}$ denotes a partition of $\mathcal{V}$ such that in each set, the induced subgraph – so-called chain component – is a connected undirected graph and each set is connected – if connected – by directed edges.

Usually for each c in $\mathcal{C}$, on can factorize $P(\boldsymbol{N}_c = \boldsymbol{n}_c | \boldsymbol{N}_{\mathrm{pa}(c)} = \boldsymbol{n}_{\mathrm{pa}(c)})$ as a product of clique factors [9]. But in the case of multinomial, negative multinomial and multivariate Poisson distribution or regressions each chain component is complete. PDAGs where chain components are not cliques could be introduced using UGs framework [14]. In such UGs, the graph is in fact a cyclic bidirected graphs which renders far more difficult and less reliable the exploration of long-range patterns in such models using simulation and even model scoring as many normalization constants – for a given clique one have a normalization constant for each different predictor value – have to be computed. Therefore we chose to consider PDAGs such that chain components are not complete.

Definition 2.1. *A clique directed acyclic graph (CDAG) is a PDAG such that:*

- *each chain component is a clique,*
- *each vertex of a clique has the same parent set,*

- *each parent set belongs to the power set of clique set the parents sets are built using combinations of present cliques vertices together and not each vertex separatly.*

Proposition 2.2.
A PDAG such as:

- *each source vertex of the graph is associated with some univariate distribution chosen among the binomial, negative binomial and Poisson families and mixtures of such distributions.*
- *each non-singleton source component of the graph is associated with some multivariate distribution chosen among diverse extensions of the multinomial family, the multivariate Poisson distribution and mixtures of such distributions,*
- *each component of the graph with at least one parent is associated with the corresponding families of univariate and multivariate regression models defined above in the case of source components,*

is equivalent to a CDAG associated with the same distributions and regressions such that for each edge in the CDAG that is not in the PDAG, the corresponding coefficient is null.

Proof. Consider the PDAG $G = (\mathcal{V}, \mathcal{E})$ and let $\tilde{G} = \left(\mathcal{V}, \tilde{\mathcal{E}}\right)$ be a CDAG with $\tilde{\mathcal{E}} = \mathcal{E}' \cup \mathcal{E}''$ – where $\mathcal{E}' \cap \mathcal{E}'' = \emptyset$ – such that

$$\mathcal{E}' = \{(u, v) \in \mathcal{E} | (v, u) \in \mathcal{E}\} \tag{3}$$

and

$$\mathcal{E}'' = \left\{(s, t) \in \mathcal{V} \times \mathcal{V} \middle| \exists\, (u, v) \in ne\,(s) \times ne\,(t) \cap \mathcal{E} \setminus \mathcal{E}'\right\} \tag{4}$$

where $ne\,(.)$ is denoting the set of neighbors of a vertex. Because of equation (3), $\tilde{G}$ has the same chain components as G, since $\mathcal{E}'$ is the set of undirected edges in both $\mathcal{E}$ and $\tilde{\mathcal{E}}$. Equation (4) implies that the set of directed edges in G is included in $\tilde{\mathcal{E}}$: only edges from all the neighbors of a parent of a child clique are added to every child clique vertices. As setting the regression coefficient to 0 does not change the conditional distribution, the two models are equivalent. □

As a consequence, given a CDAG and using ML estimators combined with Lasso type estimators [13] for parametric regressions, one can select among all PDAGs sharing the same CDAG a sparse PDAG solution with the previously presented parametric distributions. Therefore the PDAG learning task is performed using a graph search within a CDAG space which has a cardinal a little bit higher than the DAG space one but far less important than the PDAG space one (see tab. 1). This graph search can be achieved as for previous algorithms presented in [8] for DAGs using hill climbing, greedy search, first ascent and simulated annealing algorithms. For defining such an algorithm, DAG operators (add/remove/reverse directed edges) have been applied to each CDAG's DAG (see lemma 2.3). Since the space search graph is not connected using these 3 operators – chain components remains unchanged – two operators specific to CDAGs have been added: chain merging and splitting. On the one hand, two chain components c and c' of $\mathcal{C}$ such that

$$pa\,(c) = pa\left(c'\right) \cup c \;\wedge\; ch\,(c) \setminus c' = ch\left(c'\right)$$

will be merged in one chain component c'' which results from the removal of one chain component. On the other hand, a vertex from a chain component c can be removed in the chain component and set to be a parent or a child of c resulting into the addition of one chain component.

Lemma 2.3. *For any CDAG $G = (\mathcal{V}, \mathcal{E})$ with chain components $\mathcal{C}$ one can define a pair $\left(\tilde{G} = \left(\tilde{\mathcal{V}}, \tilde{\mathcal{E}}\right), \sigma^{-1}\right)$ where $\tilde{G}$ is a DAG and σ a bijection from $\mathcal{C}$ to $\tilde{\mathcal{V}}$ such as:*

$$\tilde{\mathcal{E}} = \left\{(s,t) \in \tilde{\mathcal{V}} \middle| \forall (u,v) \in \sigma^{-1}(s) \times \sigma^{-1}(t), (u,v) \in \mathcal{E} \wedge (v,u) \notin \mathcal{E}\right\} \tag{5}$$

The informative content of the CDAG and the pair is the same.

Proof. As σ is a bijection from $\mathcal{C}$ to $\tilde{\mathcal{V}}$ one only need to illustrate that no information is lost considering the parents sets of $\tilde{G}$. σ can be viewed as a numbering of the cliques in $\mathcal{C}$. The definition 2.1 ensure that if any vertex of a clique is parent of another clique vertices all vertices of the former one are also parents of the vertices in the latter one. Therefore considering separtely all vertices of the former clique or only the clique number is equivalent. □

Proposition 2.4.
Let b_K (resp. a_K) be the number of labelled CDAGs (resp. DAGs) of K vertices. One have:

$$b_K = \sum_{k=1}^{K} \left\{ K \atop k \right\} a_k \tag{6}$$

where $\left\{ K \atop k \right\}$ denote the Stirling number of second kind.

Proof. Consider a set of K vertices. The Stirling number of second kind gives the number of ways of partitioning such vertex set into k non-empty cliques. For each of these partitions one can define a DAG (see lemma 2.3) and there are a_k such labeled DAGs. We then just need to consider that the number of cliques can vary from 1 to K for CDAGs of K vertices to prove proposition 2.4. □

a_K	b_K	c_K	K
1	1	1	1
3	4	4	2
25	34	50	3
543	715	1,688	4
29,281	35,381	142,624	5
3,781,503	4,258,357	28,903,216	6
1,138,779,265	1,222,487,933	13,663,125,680	7
783,702,329,343	816,625,721,787	14,762,428,500,992	8

Table 1. Number of DAGs, CDAGs and PDAGs [12] (c_K) on up to 8 vertices (see prop. 2.4)

3 Insights of the apple tree irregular bearing phenomenon using MTBP and CDAG models

Recently, statistical indices have been proposed to characterize alternation in flowering at whole plant scale with a yearly timestep in the case of different apple tree cultivars [4]. A correlation has been highlighted between synchronicity of flowering within plants, alternation along axes, and alternation at whole plant scale. However, little is known about structural factors that may induce heterogeneity in the fates (vegetative or flowering) of sibling shoots, and thus improve regularity at whole plant scale despite alternation along axes. Considering the methodology described in [3], a tree structure (see fig. 1) was built from the apple tree dataset provided by E. Costes (UMR AGAP, AFEF Team, Inra, Montpellier, France) in order to illustrate the interest of MTBPs to investigate this phenomenon.

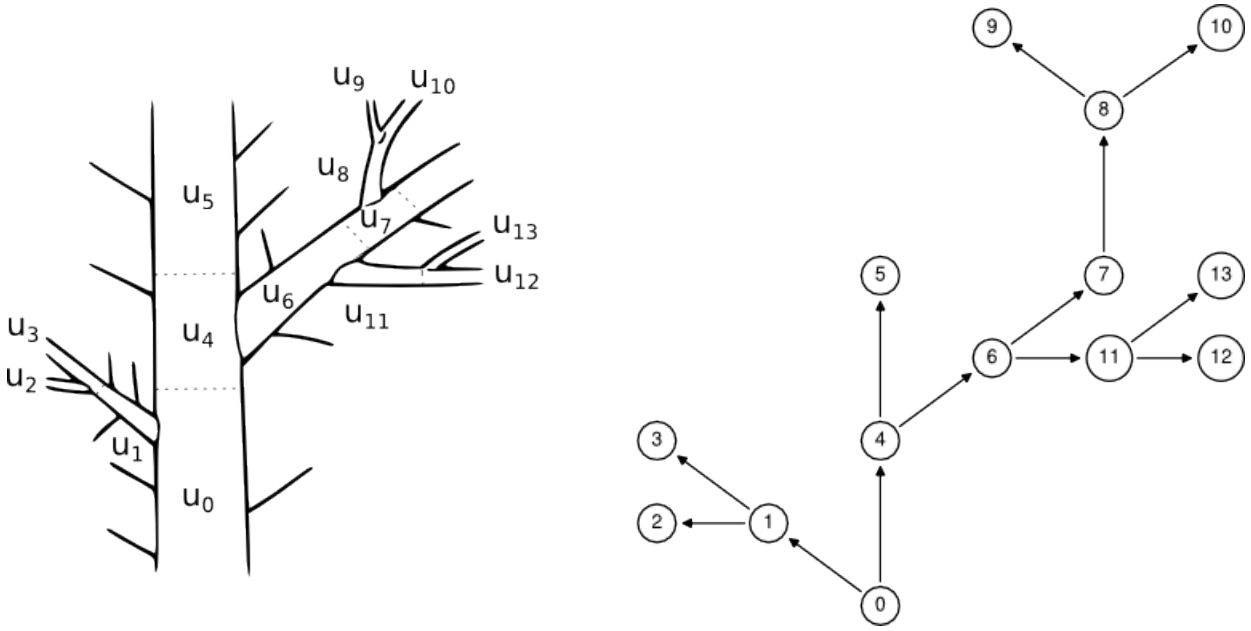

Figure 1. The tree is a formal representation of the plant topological information (drawing issued from [3]). Each label of this tree is the nature of the annual shoot.

MTBPs are used to model the number of flowering or vegetative shoots for parent shoots of different natures: length and fate (see tab. 2). The aim is to identify parent states associated with homogeneous children fates from parents that may have heterogeneous children fates. As the dataset is composed of two trees per cultivar it is also to compare the two cultivars Fuji and Braeburn that have different behaviors regarding the irregular bearing phenomenon.

CDAG-based generations distribution better fitted the data than DAG-based generation distributions according to BIC. In the worst case, we obtained the same fit with CDAG-based and DAG-based generation distributions (see fig. 2).

We obtained very different graphs for the different parent states of a given cultivar. We also obtained different graphs for the two cultivars for some parent states. This was very informative for cultivar comparison (see fig. 3) The exam of the different graphs for a given cultivar highlights the more or less regular bearing behavior at the whole plant scale. Moreover

State	Length	Fate
0	Long	Vegetative
1	Long	Flowering
2	Medium	Vegetative
3	Medium	Flowering
4	Short	Vegetative
5	Short	Flowering

Table 2. State space and corresponding natures

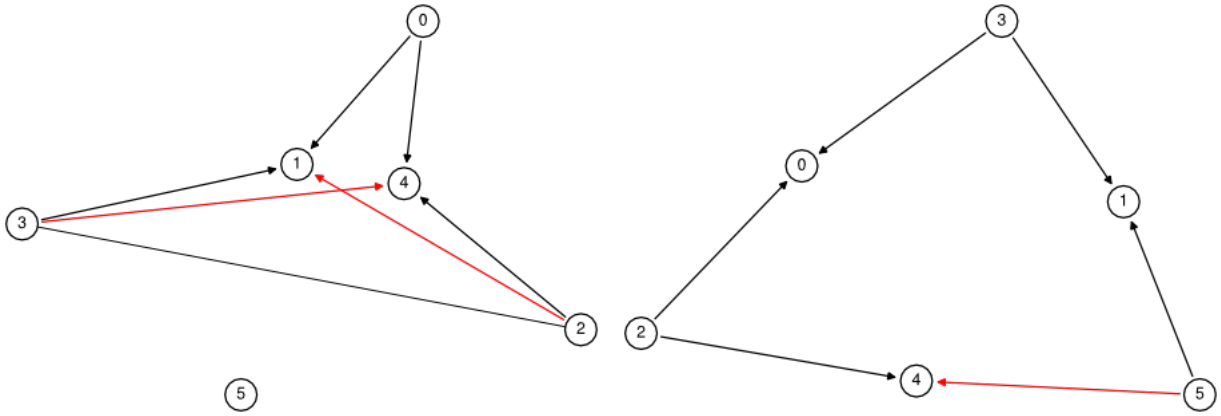

Figure 2. CDAG and DAG selected for the parent states 0 (left hand) and 1 (right hand) for the Braeburn cultivar. Edges associated with negative (resp. positive) covariances are in red (resp. black).

comparing the graphs for the two cultivars leads to a better understanding of the biological functions underlying bearing behavior. This approach seems therefore promising to highlight pattern formation such as irregular bearing in tree structure development .

Bibliography

[1] Chickering, D. (2002) *Learning equivalence classes of bayesian-network structures.* The Journal of Machine Learning Research, **2**, 445–498.

[2] Drton, M., and Eichler, M. (2006) *Maximum likelihood estimation in Gaussian chain graph models under the alternative markov property.* Scandinavian journal of statistics, **33**, 2, 247–257.

[3] Durand, J.-B., Guédon, Y., Caraglio, Y., and Costes, E. (2005) *Analysis of the plant architecture via tree–structured statistical models: the hidden Markov tree models.* New Phytologist, **166**, 3, 813–825.

[4] Durand, J.-B., Guitton, B., Peyhardi, J., Holtz, Y., Guédon, Y., Trottier, C., and Costes, E. (2013) *New insights for estimating the genetic value of segregating*

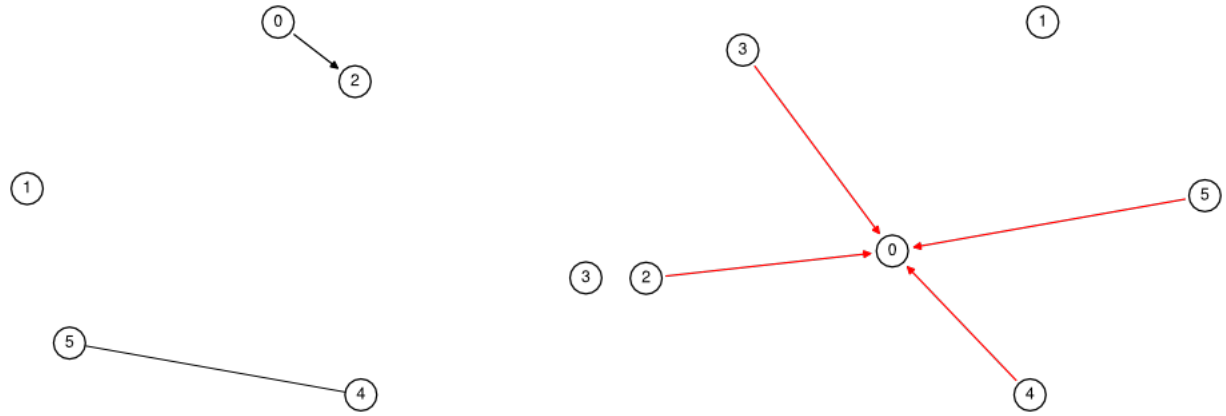

Figure 3. CDAG selected for the parent state 3 for the Braeburn (left hand) and the Fuji cultivar (right hand). Edges associated with negative (resp. positive) covariances are in red (resp. black).

apple progenies for irregular bearing during the first years of tree production. Journal of experimental botany, **64**, 16, 5099–5113.

[5] Friedman, J., Hastie, T., and Tibshirani, R. (2008) *Sparse inverse covariance estimation with the graphical lasso.* Biostatistics, **9**, 3, 432–441.

[6] Haccou, P., Jagers, P., and Vatutin, V. A. (2005) *Branching processes: variation, growth, and extinction of populations.* Cambridge University Press.

[7] Karlis, D. (2003) *An EM algorithm for multivariate Poisson distribution and related models.* Journal of Applied Statistics, **30**, 1, 63–77.

[8] Koller, D., and Friedman, N. (2009) *Probabilistic graphical models: principles and techniques.* MIT press.

[9] Lauritzen, S. (1996) *Graphical models*, vol. 17. Oxford University Press, USA.

[10] Lee, J. D., and Hastie, T. J. *Structure learning of mixed graphical models.* Submitted to the Journal of Machine Learning Research. Available: http://www.stanford.edu/ hastie/Papers/structmgm.pdf, 2012.

[11] Meyer, P., Lafitte, F., and Bontempi, G. (2008) *minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information.* BMC bioinformatics, **9**, 1, 461.

[12] Steinsky, B. (2003) *Enumeration of labelled chain graphs and labelled essential directed acyclic graphs.* Discrete Mathematics, **270**, 1, 267–278.

[13] Tibshirani, R. (1996) *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society. Series B (Methodological),, 267–288.

[14] Yang, E., Ravikumar, P., Allen, G., and Liu, Z. (2012) *Graphical models via generalized linear models.* In *Advances in Neural Information Processing Systems 25* (2012), pp. 1367–1375.