Modelling Interactions among Plant Architecture Components using Multitype Branching Processes

Fernique Pierre, Durand Jean-Baptiste, Guédon Yann

CIRAD, INRIA (Virtual Plants, Montpellier), INRA, UM2

February 7, 2012

Introduction

Testing biological hypothesis concerning demographic properties of plants entities using MTBP and model selection approaches

1. Plant as multiscale tree graphs,
2. MTBP as parsimonious models for tree graphs data structure,
3. Generation distributions and interaction modelling as Graphical Model structure learning,
4. MTBP for Apple Trees Architecture example

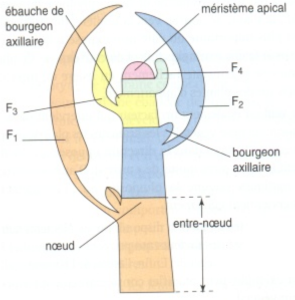


Figure: Apical Bud cross section

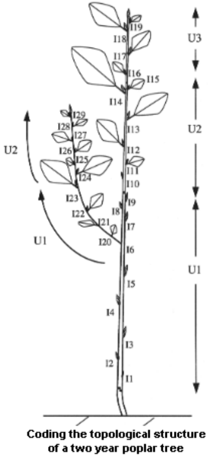
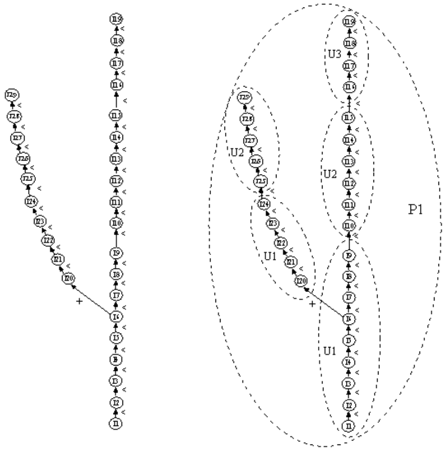


Figure: A 2 years old poplar



a) tree graph at internode scale   b) multiscale tree graph (MTG)

Figure: Graph representation of the 2 years old poplar [Godin and Costes, 1996]

Plant as multiscale tree graphs

Plant polycyclic growth example : poplar

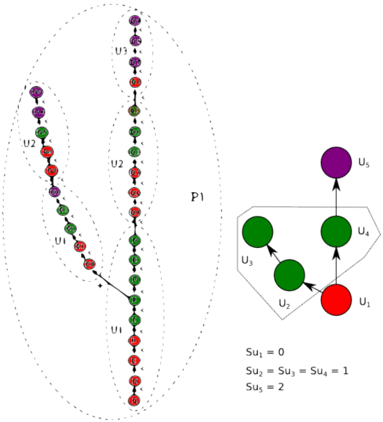


Figure: Colored graph for the 2 years old poplar : giving a shoot a state [Durand et al, 2005]

Scale choice : type of growth, pattern to study...

We consider now 4 states :

- "Long and vegetative" (0).
- "Long and floral" (1).
- "Short and vegetative" (2).
- "Short and floral" (3).

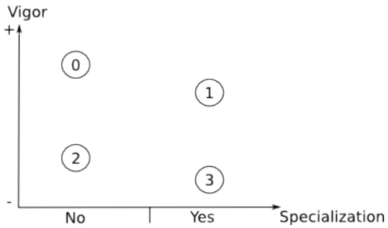


Figure: Considered states in Apple Trees and their significance

Tree process distribution :

$$P\left[\mathbf{S} = \mathbf{s}\right]$$

With some limitation to local dependencies considering biological process :

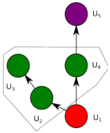


Figure: Mother/Daughters relationships among GU

Daughter shoots are independent from their ancestors knowing only their mother :

$$P\left[\mathbf{S} = \mathbf{s}\right] = P\left[S_0 = s_0\right] \prod_{u \in \mathcal{V}(T)} P\left[S_{c(u)} = s_{c(u)} \middle| S_u = s_u\right]$$

$$P\left[\mathbf{S}=\mathbf{s}\right]=P\left[S_0=s_0\right]\prod_{u\in\mathcal{V}(T)}P\left[S_{c(u)}=s_{c(u)}\middle|S_u=s_u\right]$$

Is considering order between descendants but :

- Order not always present in data,
- Order can be really hard to determine,
- Is order really important ?

MTBP are assuming that order is not relevant :

$$P\left[\mathbf{S}=\mathbf{s}\right]\propto P\left[S_0=s_0\right]\prod_{u\in\mathcal{V}(T)}P\left[\mathbf{N}_u=\mathbf{n}_u|S_u=s_u\right]$$

A MTBP distribution :

$$P\left[\mathbf{S}=\mathbf{s}\right] \propto P\left[S_0=s_0\right] \prod_{u \in \mathcal{V}(T)} P\left[\mathbf{N}_u=\mathbf{n}_u | S_u=s_u\right]$$

needs the specification of [Haccou et al., 2005] :

- 1 Initial distribution
- K generation distributions

In each generation distribution is a multivariate discrete distribution of K dimensions :

$$P\left[\mathbf{N}_u=\mathbf{n}_u | S_u=s_u\right]$$

Interaction modeling can be seen as a distribution factorization :

No Interaction		All Interactions
$\prod_{k=0}^{K-1} P\left[N_{u,k}=n_{u,k} \| S_u=s_u\right]$	$\leftrightarrow$	$P\left[\mathbf{N}_u=\mathbf{n}_u \| S_u=s_u\right]$
Covariances $=0$		Covariances $<0, >0$

- $n_{u,k} \in \mathbb{N} \Rightarrow$ A parsimonious model : Parametric distributions
- Environnement, Species ... have an influence on number of children $\Rightarrow$ MGLMM approach

MGLMM :

$$\phi\left(E\left[\mathbf{N}|S_u = s_u, \mathbf{X} = \mathbf{x}, T = t\right]\right) = \boldsymbol{\alpha} + < \mathbf{x}, \boldsymbol{\beta} > + \boldsymbol{\zeta}_T$$

LM : Covariates
G + LM : LG distribution $\rightarrow$ Count distributions
M + GLM : Multivariate count distributions
MGLM + M : Random effects ζ_T realization of LG distribution $\mathcal{N}(0, \tau)$

- MTBP : No distinction between ramification $r(u)$ and succession $s(u)$
  $\leftarrow$ partial order :

$$P\left[\mathbf{S}_{ch(u)} = \mathbf{s}_{ch(u)} \middle| S_u = s_u\right] =$$
$$P\left[S_{s(u)} = s \middle| S_u = s_u\right] P\left[\mathbf{N}_{r(u)} = \mathbf{n}_{r(u)} \middle| S_u = s_u, S_{s(u)} = s\right]$$

- Dependences pattern research is complex : does not depends on covariates.

No interaction $\leftrightarrow$ All interactions

$$\prod_{k=0}^{K-1} P\left[N_{u,k} = n_{u,k} | S_u = s_u\right] \qquad P\left[\mathbf{N}_u = \mathbf{n}_u | S_u = s_u\right]$$

- Guided by graphs encoding distribution factorization
- Different approach to build the graph of factorization :
  - Contingency tables and loglinear models one
  - Information theory one
  - Combinatorial optimization one

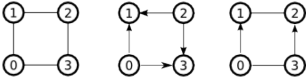


Figure: Graphical models

Independencies using separation properties [Lauritzen, 1996] :

- Undirected Graph

$$0 \perp\!\!\!\perp 2 | 1, 3 \quad 1 \perp\!\!\!\perp 3 | 2, 0$$

- Directed Acyclic Graph

$$0 \perp\!\!\!\perp 2$$

- Partially Directed Acyclic Graph

$$0 \perp\!\!\!\perp 2 | 1, 3 \quad 1 \perp\!\!\!\perp 3 | 2, 0$$

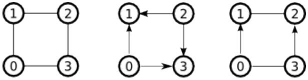


Figure: Graphical models

Factorizations [Lauritzen, 1996] :

- Undirected Graph

$$P\left[\mathbf{N}=\mathbf{n}\right]=\frac{1}{Z}\prod_{c\in\mathcal{C}(G)}\phi_{\mathbf{N}_c}\left(\mathbf{n}_c\right)$$

$$P\left[\mathbf{N}=\mathbf{n}\right]=\frac{1}{Z}\phi_{\mathbf{N}_{0,1}}\left(\mathbf{n}_{0,1}\right)\phi_{\mathbf{N}_{1,2}}\left(\mathbf{n}_{1,2}\right)\phi_{\mathbf{N}_{2,3}}\left(\mathbf{n}_{2,3}\right)\phi_{\mathbf{N}_{3,1}}\left(\mathbf{n}_{3,1}\right)$$

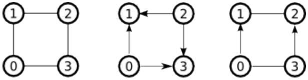


Figure: Graphical models

Factorizations [Lauritzen, 1996] :

- Directed Acyclic Graph

$$P[\mathbf{N} = \mathbf{n}] = \prod_{v \in \mathcal{V}} P\left[N_v = n_v \middle| \mathbf{N}_{pa(v)} = \mathbf{n}_{pa(v)}\right]$$

$$P[\mathbf{N} = \mathbf{n}] = P[N_0 = n_0]\, P[N_2 = n_2]$$
$$P[N_1 = n_1 | \mathbf{N}_{0,2} = \mathbf{n}_{0,2}]\, P[N_3 = n_3 | \mathbf{N}_{0,2} = \mathbf{n}_{0,2}]$$

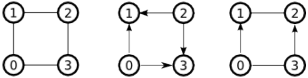


Figure: Graphical models

Factorizations [Lauritzen, 1996] :

- Partially Directed Acyclic Graph

$$P[\mathbf{N}=\mathbf{n}] = \prod_{c \in \mathcal{C}(G)} P\left[\mathbf{N}_c = \mathbf{n}_c \middle| \mathbf{N}_{pa(c)} = \mathbf{n}_{pa(c)}\right]$$

$$P[\mathbf{N}=\mathbf{n}] = P[\mathbf{N}_{0,3} = \mathbf{n}_{0,3}]\, P[\mathbf{N}_{1,2} = \mathbf{n}_{1,2} | \mathbf{N}_{0,3} = \mathbf{n}_{0,3}]$$

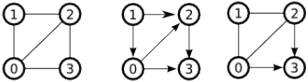


Figure: Different I-equivalent graphical models

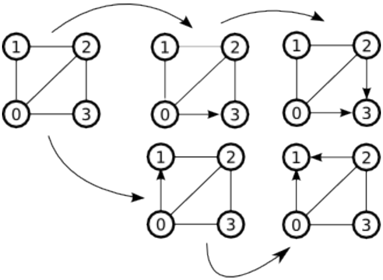


Figure: From undirected graphical models to mixed graphical models

Number of vertices	1	2	3	4	5	6
Number of DAG	1	3	25	543	29,281	3,781,503
Number of UG	1	2	8	64	1,024	32,768

Table: Number of possible graphs when vertices number is increasing [Robinson, 1973]

Structure Learning :

- Restricted Family of graphs for which solutions can be easily found : Tree Graphs [Chow and Liu, 1968]
- Stepwise research with restricted search space by choice of a heuristic [Agresti, 2002; Koller, 2009].

		$N_2 = 0$	$N_2 = 1$
$N_0 = 0$	$N_1 = 0$	5	10
$N_0 = 0$	$N_1 = 1$	4	2
$N_0 = 1$	$N_1 = 0$	6	10
$N_0 = 1$	$N_1 = 1$	20	0

Table: A contingency table

Loglinear model [Agresti, 2002]:

$$\log\left(\mu^{(i_0,i_1,i_2)}\right) = \beta + \beta^0_{i_0} + \beta^1_{i_1} + \beta^2_{i_2} + \beta^{0,1}_{i_0,i_1} + \beta^{0,2}_{i_0,i_2} + \beta^{1,2}_{i_1,i_2} + \beta^{0,1,2}_{i_0,i_1,i_2}$$

When $\beta^{0,2}_{i_0,i_2} = \beta^{0,1,2}_{i_0,i_1,i_2} = 0$, following independency model : $2 \perp\!\!\!\perp 0|1$

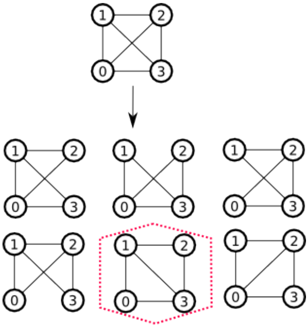


Figure: Loglinear models choice

The weakest p-value if $p - value > 0.05$

- Entropy :

$$H(N_i, N_j) = -\sum_{n_i, n_j} P\left[N_i = n_i, N_j = n_j\right] \log\left(P\left[N_i = n_i, N_j = n_j\right]\right)$$

If $N_i \perp\!\!\!\perp N_j$, $H(N_i, N_j) = H(N_i) + H(N_j)$

- Mutual Information :

$$I(N_i, N_j) = -\sum_{n_i, n_j} P\left[N_i = n_i, N_j = n_j\right] \log\left(\frac{P\left[N_i = n_i, N_j = n_j\right]}{P\left[N_i = n_i\right] P\left[N_j = n_j\right]}\right)$$

If $N_i \perp\!\!\!\perp N_j$, $I(N_i, N_j) = 0$

- Kullback divergence :

$$KL\left(P_0, P_1\right) = \sum_{\mathbf{n}} P_0\left(\mathbf{N} = \mathbf{n}\right) \log\left(\frac{P_0\left(\mathbf{N} = \mathbf{n}\right)}{P_1\left(\mathbf{N} = \mathbf{n}\right)}\right)$$

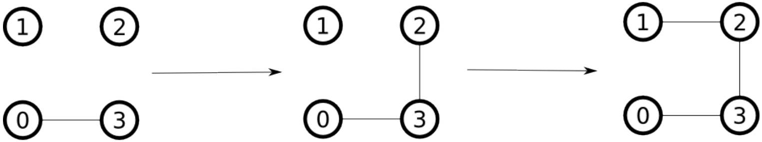
If $P_0 = P_1$, $KL\left(P_0, P_1\right) = 0$

With fixed structure [Chow and Liu, 1968] :

$$P\left[\mathbf{N}=\mathbf{n}\right]=P\left[N_{(0)}=n_{(0)}\right]\prod_{i=1}^{K-1}P\left[N_{(i)}=n_{(i)}\middle|N_{pa(i)}=n_{pa(i)}\right]$$

Minimize $KL\left(P_f,P\right)\Rightarrow$ Maximum spanning tree with mutual information as edge weight

$$MIM=\begin{pmatrix} * & . & . & . \\ 1 & * & . & . \\ 4 & 3 & * & . \\ 6 & 2 & 5 & * \end{pmatrix}$$



With no fixed structure :
Theoretically

$$I(N_i, N_j) = 0$$

If $N_i \perp\!\!\!\perp N_j$ but,

$$I(N_i, N_j) \approx 0$$

with frequencies.

- Calculate an Information derived score for each edge,
- Fixing a threshold,
- Add every edge such as its score is superior to the threshold

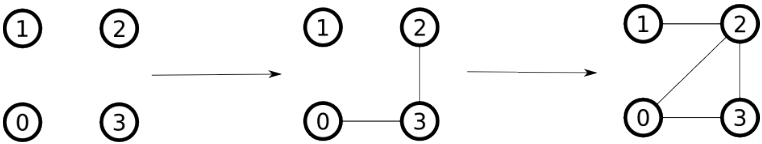


Figure: Threshold Algorithms

$I(N_i, N_j)$ do not take into account $(N_k)_{k \neq \{i,j\}}$ (Relevance algorithm [Butte, 2000])
So some other scores were considered :

- CLR Algorithm [Faith, 2007],

$$Z(N_i, N_j) = \sqrt{\left(\frac{I(N_i, N_j) - \mu_i}{\sigma_i}\right)^2 + \left(\frac{I(N_i, N_j) - \mu_j}{\sigma_j}\right)^2}$$

- ARACNE [Margolin, 2006], if $i \perp\!\!\!\perp j | k$:

$$I(N_i, N_j) \leq \min\left(I(N_i, N_k), I(N_k, N_j)\right) \Rightarrow \text{New threshold}$$

- MRNET [Meyer, 2009],
  Each step : Optimal pairwise approximation of $I\left(N_i, N_j \middle| S^{(t)}(N_i, N_j)\right)$

Edges : Present or Absent
⇒ Finite number of possible graphs for a given number of vertices but exhaustive research not feasible
For each graph : possible scoring (likelihood, BIC, AIC...)
⇒ Structure learning as optimization of score function with finite number of solutions : Combinatorial optimization
Greedy Algorithms :

- Starting point
- Score Function
- Search Operators

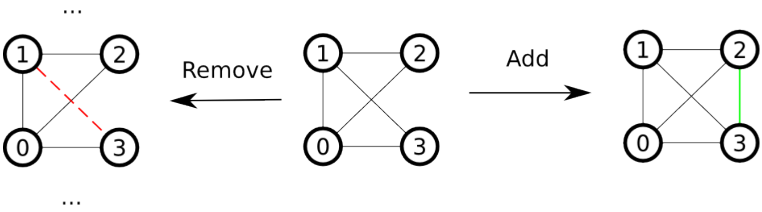


Figure: Search operators for undirected graphs [Koller, 2009]

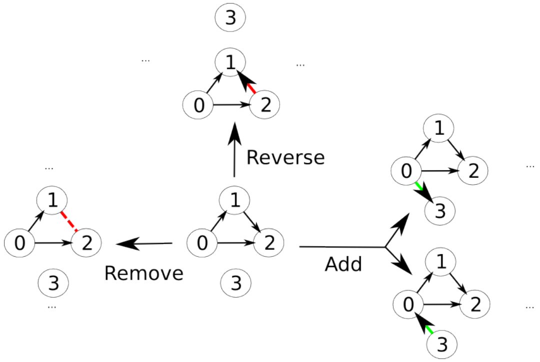
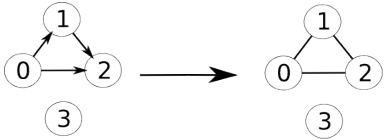


Figure: Search operators for directed graphs [Koller, 2009]

Mixed graphs : complex and not well documented

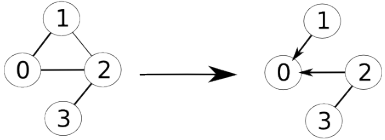
2 examples :

- From DAG to PDAG : clique operator



- From UG to PDAG : v-shape operator

...



...

2 approaches : parametric estimation and non-parametric then parametric one.

Partially Directed Acyclic Graph

$$P\left[\mathbf{N}=\mathbf{n}\right]=\prod_{c\in\mathcal{C}(G)}P\left[\mathbf{N}_c=\mathbf{n}_c\middle|\mathbf{N}_{pa(c)}=\mathbf{n}_{pa(c)}\right]$$

Discrete Distributions to consider :

- Univariate
- Univariate Conditional
- Multivariate
- Multivariate Conditional

Generation distributions and interaction modelling
Parametric Univariate Discrete Distributions [Johnson, 1997]

Figure: Univariate Distribution subgraph

- Binomial distribution
- Negative Binomial distribution
- Poisson distribution

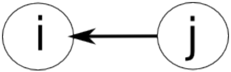


Figure: Univariate Conditional Distribution subgraph

$$\phi\left(E\left[N_i|\mathbf{N}_j=\mathbf{n}_j\right]\right)=\alpha+<f\left(\mathbf{n}_j\right),\beta>$$

- link function : identity, log, logit...
- family : Binomial, Poisson, Negative Binomial

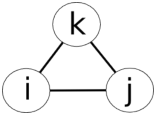


Figure: Multivariate Distribution subgraph

Univariate distribution generalization [Johnson, 1997; Karlis, 2003]:

- Binomial → Multinomial (-)
- Negative Binomial → Negative Multinomial (+)
- Poisson → Multivariate Poisson (+)

Multinomial → Compound Multinomial (+ : Negative Binomial, - : Binomial, 0 : Poisson)

Only one sign for covariances !

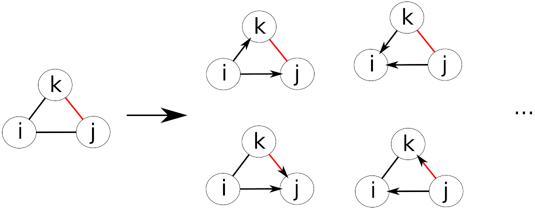


Figure: Multivariate Distribution subgraph

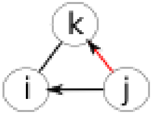


Figure: Multivariate Conditional Distribution subgraph

Compound Multinomial :

$$N_i + N_k \sim \mathcal{P}_\theta$$

$$\mathbf{N}_{i,k} | N_i + N_k = n \sim \mathcal{M}\left(n, \mathbf{p}\right)$$

Multivariate Conditional Distribution :

$$N_i + N_k | \mathbf{N}_j = \mathbf{n}_j \sim \mathcal{P}_{\theta\left(\mathbf{n}_j\right)}$$

4 states :

- "Long and vegetative" (0).
- "Long and floral" (1).
- "Short and vegetative" (2).
- "Short and floral" (3).

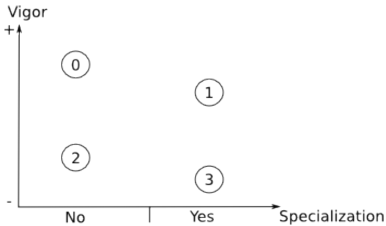


Figure: Considered states in Apple Trees and their significance

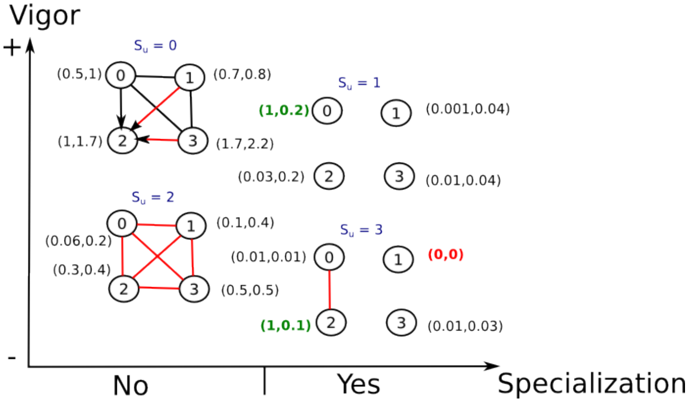


Figure: Generation distributions for Apple Trees

OpenAlea :

- Open source project for plant research community.
- Collaborative effort to develop Python libraries and tools for Plant Architecture modeling.

VPlants modules developped by Virtual Plants team (INRIA : Chistophe Godin, CIRAD : Yann Guedon)

Our work :

- Structure Analysis (statistic tools)
- Tree Analysis

With : Yann Guedon, Jean-Baptiste Durand, Jean Peyhardi (PhD Std 2nd year), Me (PhD Std 1rst year).

C++ programming with Python wrappers

- Discrete Distributions :
  - Univariate [Yann Guedon]
  - Univariate conditional
  - Multivariate [Jean-Baptiste Durand]
  - Multivariate conditional
  - Graphical
- Probabilistic Graphical Models structure learning :
  - CT and loglinear models (Apple Tree)
  - Information Theory algorithms
  - Greedy algorithm (UG adding edges)
- MTBP [Jean-Baptiste Durand]

- Discrete Distributions :
  - Multivariate Conditional (with Jean Peyhardi)
  - Inheritance between C++ classes (and then python)
  - Random effects (with Jean Peyhardi)
- Probabilistic Graphical Models :
  - Greedy algorithms (DAG and PDAG)
  - PDAG Learing approaches comparison
  - PDAG : LWF and AMP factorization properties
  - Incremental (or not) algorithms $\rightarrow$ Dynamic algorithms (strictly connected components, maximum clique, vertices ordering)
- MTBP :
  - Hidden MTBP [Jean Baptiste Durand]
  - MTBP with partial order [Jean Baptiste Durand]
  - Model discussion