

A statistical modeling framework for analyzing tree-indexed data

Pierre Fernique

▶ **To cite this version:**

Pierre Fernique. A statistical modeling framework for analyzing tree-indexed data: Application to plant development on microscopic and macroscopic scales. Statistics [math.ST]. Universite de Montpellier 2, 2014. English. NNT : . tel-01095420

HAL Id: tel-01095420

https://theses.hal.science/tel-01095420

Submitted on 10 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

Docteur

délivré par l'**Université Montpellier II**

préparée au sein de l'école doctorale **I2S***
et des unités de recherche **UMR 5149, UMR AGAP**
dans l'équipe projet Inria **Virtual Plants**

présentée par **Pierre Fernique**
le **10/12/2014**

Titre :

A statistical modeling framework for analyzing tree-indexed data

Application to plant development at microscopic and macroscopic scales

Discipline : **Mathématiques appliquées**
Spécialité : **Biostatistique**

Thèse soutenue devant le jury composé de :

M. Philippe Leray	Polytech'Nantes, P[r]	Rapporteur
M. Stéphane Robin	AgroParisTech, DR	Rapporteur
M. Jean-Michel Marin	Université Montpellier 2, P[r]	Président
M. Yann Guédon	CIRAD Montpellier, DR	Directeur de thèse
M. Jean-Baptiste Durand	Université Grenoble Alpes, MC	Co-directeur de thèse
M. Pierre-Éric Lauri	CIRAD Montpellier, IR	Éxaminateur

* **I2S:** Information Structures Systèmes

"Je remercie Yann de m'avoir appris que les arbres parlaient"

Yves Caraglio lors d'un des ateliers annuels de l'équipe Virtual Plants

".. Puisque les arbres ont décidé de parler, gageons que ce n'est pas pour raconter des conneries..."

Robin des bois dans *La légende de Robin des bois* de Manu Larcenet

Remerciements

Résumé

Titre Un cadre de modélisation statistique pour l'analyse de données indexées par des arborescences – Application au développement des plantes à l'échelle microscopique et macroscopique

Résumé Nous nous intéressons à des modèles statistiques pour les données indexées par des arborescences. Dans le contexte de l'équipe Virtual Plants, équipe hôte de cette thèse, les applications d'intérêt portent sur le développement de la plante et sa modulation par des facteurs environnementaux et génétiques. Nous nous restreignons donc à des applications issues du développement de la plante, à la fois au niveau microscopique avec l'étude de la lignée cellulaire du tissu biologique servant à la croissance des plantes, et au niveau macroscopique avec le mécanisme de production de branches. Le catalogue de modèles disponibles pour les données indexées par des arborescences est beaucoup moins important que celui disponible pour les données indexées par des chemins. Cette thèse vise donc à proposer un cadre de modélisation statistique pour l'étude de patterns pour données indexées par des arborescences. À cette fin, deux classes différentes de modèles statistiques, les modèles de Markov et de détection de ruptures, sont étudiées.

Mots-clés Architecture des plantes; données indexées par des arborescences lignage cellulaire; modèle de détection de ruptures; modèle de Markov; modèle graphique

---

Title A statistical modeling framework for analyzing tree-indexed data – Application to plant development at microscopic and macroscopic scales

Abstract We address statistical models for tree-indexed data. In the context of the Virtual Plants team, host team of this thesis, applications of interest focus on plant development and its modulation by environmental and genetic factors. We thus focus on plant developmental applications, both at the microscopic level with the study of the cell lineage in the biological tissue responsible for the plant growth, and at the macroscopic level with the mechanism of production of branches. The catalog of models available for tree-indexed data is far less important than the one available for path-indexed data. This thesis therefore aims at proposing a statistical modeling framework for studying patterns in tree-indexed data. To this end, two different classes of statistical models, Markov and change-point models, are investigated.

Keywords Cell lineage; change-point model; graphical model; Markov model; plant architecture; tree-indexed data

Virtual Plants
Campus St Priest
860 rue de St Priest, Bat. 5
39095 Montpellier Cedex 5, France

Contents

List of Acronyms ii

List of Notations vi

List of Figures xi

List of Tables xii

Introduction 2

1 Graphs and graphical models frameworks 8
- 1.1 Introduction to graph theory . . . 9
  - 1.1.1 Definitions . . . 9
  - 1.1.2 Drawings . . . 10
  - 1.1.3 Graph properties . . . 19
- 1.2 Graphical model framework . . . 27
  - 1.2.1 Random vectors and independencies . . . 27
  - 1.2.2 From graphs to distributions . . . 30
  - 1.2.3 From distributions to graphs . . . 35
- 1.3 Gaussian graphical models . . . 35
  - 1.3.1 Parametrizations . . . 36
  - 1.3.2 Inference . . . 38

References . . . 40

2 Tree-indexed data and Markov Tree (MT) models 46
- 2.1 Introduction to tree-indexed data . . . 47
  - 2.1.1 Definitions . . . 47
  - 2.1.2 Drawing tree-indexed data . . . 48
- 2.2 Tree-indexed data and plants . . . 51
  - 2.2.1 Tree-indexed data at cellular scale . . . 52
  - 2.2.2 Tree-indexed data at whole plant scale . . . 55
- 2.3 Markov models for tree indexed-data . . . 59
  - 2.3.1 Markov models . . . 60
  - 2.3.2 Hidden Markov Tree (HMT) models . . . 61

References . . . 62

3 Semi-parametric Hidden Markov Out-Tree (HMOT) models for cell lineage analysis 66
- 3.1 Introduction . . . 67
- 3.2 Definitions . . . 68
  - 3.2.1 Markov Out-Tree (MOT) models . . . 68
  - 3.2.2 Hidden Markov Tree (HMT) models . . . 72

3.3 Computational methods for Hidden Markov Out-Tree (HMOT) models . 73
3.3.1 Upward-downward smoothing algorithm 73
3.3.2 Application of the EM algorithm 76
3.3.3 Dynamic programming restoration algorithm 78
3.4 Application to cell lineage trees 79
3.4.1 Results 79
3.4.2 Discussions 83
References 86

4 Inference of Mixed Acyclic Graphical Models (MAGMs) in Multi-Type Branching Processes (MTBPs) 88
4.1 Introduction 89
4.2 Definitions 90
4.2.1 Multi-Type Branching Processes (MTBPs) 90
4.2.2 Poisson Mixed Acyclic Graphical Models (PMAGMs) 91
4.2.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) 95
4.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) inference 95
4.3.1 Parameter inference 96
4.3.2 Structure inference 96
4.4 Application to Multi-Type Branching Processes (MTBPs): the case of mango tree asynchronisms 99
4.5 Concluding remarks 102
References 104

5 Quantification of plant patchiness via tree-structured statistical models: a tree-segmentation/clustering approach 108
5.1 Introduction 109
5.2 Material and methods 110
5.2.1 Tree-structured representation of plants 110
5.2.2 Modeling plant patchiness with tree segmentation/clustering models 111
5.2.3 Plant material 113
5.3 Results 115
5.3.1 Tree segmentation 115
5.3.2 Subtree clustering 116
5.3.3 Cultivar comparisons 118
5.4 Discussion 118
References 121

Work in progress and perspectives 124
StatisKit: graphical model inference in C++ and Python 125
Hidden Markov In-Tree (HMIT) models 127
Multivariate mixture models in Multi-Type Branching Processes (MTBPs) . . 129
Integrative models for deciphering mango tree asynchronisms 131
References 132

Index of references 136

List of Acronyms

AIC Akaike Information Criterion. 96

ALT Automated Lineage Tracking. 53, 54

AMP Alternate Markov Property. 35

API Application Programming Interface. 126

AR Auto-Regressive. 128

BIC Bayesian Information Criterion. 39, 80, 83, 90, 96

DAG Directed Acyclic Graph. 23, 24, 26, 32, 39, 89, 93, 96–98, 125

DAGM Directed Acyclic Graphical Model. 89, 90, 94, 96, 97

DF Directed Factorization property. 33, 37, 94

DFS Depth-First Search. 17, 50

DGM Directed Global Markov property. 33, 37

DLM Directed Local Markov property. 33, 37

DNA DesoxyriboNucleic Acid. 2

DPM Directed Pairwise Markov property. 32, 33, 37

DPMAGM Discrete Parametric Mixed Acyclic Graphical Model. 88, 90, 95, 97–99, 101, 103, 124, 129, 132

EDAG Equivalent Directed Acyclic Graph. 97

EM Expectation-Maximization. 5, 62, 66, 67, 73, 75–77, 84, 96, 112, 124, 131

F Factorization property. 31–34, 37, 93

FC Factorization Chain property. 34, 37, 92

GCM Global Chain Markov property. 34, 37

GDAGM Gaussian Directed Acylic Graphical Model. 103

GES Greedy Equivalent Search. 97

GLM Generalized Linear Model. 89, 94

GM Global Markov property. 31–33, 37

GMAGM Gaussian Mixed Acylic Graphical Model. 103, 124

GU Growth Unit. 56–60, 99–104, 109, 110, 113–116, 118

HIMOT Independent Hidden Markov Out-Tree. 62, 74, 76, 84

HMC Hidden Markov Chain. 3, 61, 67, 72

HMIT Hidden Markov In-Tree. 6, 62, 125, 128, 130

HMM Hidden Markov Model. 67

HMOOT Hidden Markov Ordered Out-Tree. 73

HMOT Hidden Markov Out-Tree. 4–6, 66, 67, 72, 73, 77–79, 83, 124, 125, 127, 129, 130, 132

HMT Hidden Markov Tree. 3, 61, 62, 72, 73, 110, 125

HMUIT Hidden Markov Unordered In-Tree. 129, 130

HMUOT Hidden Markov Unordered Out-Tree. 73, 82–85, 124, 130, 132

I-map Independence map. 4, 35, 37, 38, 95–99

IMOT Independent Markov Out-Tree. 60, 62

LCM Local Chain Markov property. 34, 37

LM Local Markov property. 31, 37

LWF Lauritzen, Wermuth and Frydenberg property. 35

MAG Mixed Acyclic Graph. 5, 26, 27, 33, 40, 92, 97–99, 103, 104, 124, 125

MAGM Mixed Acyclic Graphical Model. 89–92, 94–96, 103

MAP Maximum A Posteriori. 112, 118, 119

MARS Multi-angle Acquisition, 3 dimensional Reconstruction and Segmentation. 53, 54

MC Markov Chain. 3, 61, 70, 127, 128

MCEM Monte Carlo EM. 77

MCMC Monte-Carlo Markov Chain. 94

MIT Markov In-Tree. 3, 60, 61, 69, 127, 128

ML Maximum Likelihood. 38, 39, 73, 95–97, 111

MMMC Mixed Memory Markov Chain. 128

MOOT Markov Ordered Out-Tree. 69, 71, 73, 85

MOT Markov Out-Tree. 3–5, 60, 67, 69, 71, 127

MT Markov Tree. 3, 6, 61, 68, 69, 76

MTBP Multi-Type Branching Process. 3–5, 60–62, 71, 88–91, 93, 102, 104, 124

MTG Multiscale Tree Graph. 58, 110, 120

MUIT Markov Unordered In-Tree. 128

MUOT Markov Unordered Out-Tree. 71, 73, 85

NP Non-deterministic Polynomial-time. 20

PCM Pairwise Chain Markov property. 34, 37

PDAG Partially Directed Acyclic Graph. 97

PDAGM Poisson Directed Acyclic Graphical Model. 94, 95

PM Pairwise Markov property. 31, 37

PMAGM Poisson Mixed Acyclic Graphical Model. 88, 90, 92–94, 101

QAG Quotient Acyclic Graph. 97–99, 101, 103, 104, 124

RAM Root Apical Meristem. 52

S-equivalent Separation equivalent. 24, 27, 28, 97

SAM Shoot Apical Meristem. 52–55

List of Notations

$\cdot \perp\!\!\!\perp \cdot$ Marginal independence of two sets of variables. 29

$\cdot \perp\!\!\!\perp \cdot \mid \cdot$ Conditional independence of two sets of variables given a third set. 29

$\mathcal{I}(\cdot)$ Set of all independencies holding in a distribution. 30

$\mathcal{G}$ A simple graph. 9

$\mathcal{H}_{\mathcal{G}}$ Chain set of $\mathcal{G}$. 25

$\mathcal{K}_{\mathcal{G}}$ Maximal clique set of $\mathcal{G}$. 19

$\mathcal{E}$ Edge set of $\mathcal{G}$. 9

$\mathcal{E}^r$ Reversed edge set of $\mathcal{G}$. 23

$\mathcal{I}_{\mathcal{G}}$ Immoralities set of $\mathcal{G}$. 23, 26

$\mathcal{L}$ Leaf vertices of graph $\mathcal{G}$. 23

$\underline{\mathcal{G}}$ Adjacency or incidence matrix of $\mathcal{G}$. 17

$\mathcal{G}^m$ Moral graph of $\mathcal{G}$. 23, 26

$\mathcal{R}$ Root vertices of graph $\mathcal{G}$. 23

$\cdot \perp \cdot$ Graph separation, d-separation or m-separation of two set of vertices. 21, 24, 27

$\cdot \perp \cdot \mid \cdot$ Graph separation, d-separation or m-separation of two set of vertices given a third. 21, 24, 27

$\mathcal{S}(\cdot)$ Set of all separations holding in a graph. 21

$\mathcal{D}(\cdot)$ Set of directed graphs with given vertex set. 9

$\mathcal{M}(\cdot)$ Set of mixed graphs with given vertex set. 10

$\mathcal{U}(\cdot)$ Set of undirected graphs with given vertex set. 9

$\mathcal{G}^u$ The undirected version of $\mathcal{G}$. 23, 26

$\mathbf{an}(\cdot)$ Ancestor set of a vertex or vertex set. 22

$\mathbf{An}(\cdot)$ Ancestor set closure of a vertex or vertex set. 22

$\mathbf{bd}(\cdot)$ Boundary set of a vertex or vertex set. 25

$\mathbf{Bd}(\cdot)$ Boundary set closure of a vertex or vertex set. 25

ch $(\cdot)$ Child set of a vertex or vertex set. 22

Ch $(\cdot)$ Child set closure of a vertex or vertex set. 22

cn $(\cdot)$ Connected set of a vertex or vertex set. 19

Cn $(\cdot)$ Connected set closure of a vertex or vertex set. 20

$\deg(\cdot)$ Degree of a vertex. 19

$\deg^{-}(\cdot)$ In-degree of a vertex. 22

$\deg^{+}(\cdot)$ Out-degree of a vertex. 22

de $(\cdot)$ Descendant set of a vertex or vertex set. 22

De $(\cdot)$ Descendant set closure of a vertex or vertex set. 22

ne $(\cdot)$ Neighbors set of a vertex or vertex set. 19

Ne $(\cdot)$ Neighbors set closure of a vertex or vertex set. 19

nd $(\cdot)$ Non-descendant set of a vertex or vertex set. 22

Nd $(\cdot)$ Non-descendant set closure of a vertex or vertex set. 22

pa $(\cdot)$ Parent set or parent of a vertex or vertex set. 22

Pa $(\cdot)$ Parent set closure of a vertex or vertex set. 22

$\mathcal{V}$ Vertex set of $\mathcal{G}$. 9

$\wedge$ Logical *and* operator. 10

$\vee$ Logical *or* operator. 18

Π Partition of $\mathcal{G}$ vertex set. 10

$\mathcal{E}_{\Pi}$ Edge set of $\mathcal{G}_{\Pi}$. 10

$\mathcal{G}_{\Pi}$ A quotient graph of $\mathcal{G}$ quotiented by the vertex set partition Π. 10

$\mathcal{V}_{\Pi}$ Vertex set of $\mathcal{G}_{\Pi}$. 10

$|\cdot|$ Cardinality of a set. 11

$\mathcal{P}(\cdot)$ Set of pairs of distinct elements of a set. 9

$\mathfrak{S}(\cdot)$ Set of all permutations of a set. 18

$\uplus$ Union of disjoint sets. 10

$\mathcal{G}_{\mathcal{A}}$ Subgraph of $\mathcal{G}$ induced by the vertex subset $\mathcal{A}$. 10

$\mathcal{E}_{\mathcal{A}}$ Edge set of $\mathcal{G}_{\mathcal{A}}$. 10

$\mathcal{V}_{\mathcal{A}}$ Vertex set of $\mathcal{G}_{\mathcal{A}}$. 10

$\cdot^T$ Transpose of a vector or a matrix. 36

$||\cdot||$ Norm of a vector. 13

$\tilde{\cdot}$ Normalization of a vector. 13

X A random variable. 27

x An outcome of the random variable X. 27

$\mathcal{X}$ Observation space of the random variable X. 27

$\boldsymbol{X}$ A random vector. 29

$\boldsymbol{x}$ An outcome of the random vector $\boldsymbol{X}$. 29

$\boldsymbol{\mathcal{X}}$ Observation space of the random variable $\boldsymbol{X}$. 29

List of Figures

1.1 Examples of force-directed undirected graphs layout evolution 13
1.2 Examples of force-directed directed graphs layout evolution 16
1.3 Examples of movements induced by a magnetic field 16
1.4 Examples of magnetic fields 17
1.5 Adjacency matrix drawing 18
1.6 Remarkable undirected graphs 20
1.7 Remarkable directed graphs 24
1.8 The d-separation property 25
1.9 Remarkable mixed graphs 27
1.10 The m-separation property 28
1.11 Separation relations among the classes of graphs 28
1.12 Moussouris (1974) chordless four-cycle graph 33
1.13 Examples of Gaussian graphical models 38

2.1 Quotient tree graphs 48
2.2 Drawing of trees 51
2.3 Meristems and example of tree-indexed data at microscopic scale 53
2.4 Example of a 3D + t images reconstruction and segmentation 54
2.5 Shoot apical meristem and stem organization 56
2.6 Tree-indexed data representation of plants 57
2.7 Plant modularity 58
2.8 Illustration of mango tree patchiness 59

3.1 Example of a 3D + t images and meristem early stages 79
3.2 Spatial regions on the floral meristem at stage 3 80
3.3 Observation distributions of the hidden Markov unordered out-tree model 81
3.4 Restoration of hidden states using the Viterbi-like algorithm for hidden Markov unordered out-tree 82
3.5 Iterations of the expectation-maximization algorithm for the independent and dependent Markov unordered out-tree models 84

4.1 Local search in mixed acyclic graph and quotient acyclic graph search spaces 99
4.2 Scheme of mango tree growth cycles 101
4.3 States of growth units in mango trees 102
4.4 Mixed acyclic graph and quotient acyclic graph of a generation distribution 103

5.1 Tree-indexed data extraction from plants 110
5.2 Mango tree growth cycles 114
5.3 Illustration of mango tree patchiness 116
5.4 Ternary plots of the outputs of the segmentation/clustering algorithm . . 117
5.5 Comparisons of patch patterns for the different cultivars 119

5.6 Performance of the segmentation heuristic for tree-indexed data 120
6.7 Comparison of the restoration of hidden states using the Viterbi-like algorithm for hidden Markov unordered in-tree and hidden Markov unordered out-tree 130

List of Tables

1.1 Forces used in classical force-directed algorithms 14
1.2 Usual magnetic fields for force-directed algorithms 15

3.1 Number of parameters of Markov ordered out-tree models as a function of the number of states and the maximal degree. 70
3.2 Number of parameters of Markov unordered out-tree models as a function of the number of states and the maximal degree. 72
3.3 Confusion table regarding the most probable state tree for the model with number of children in each state independent against the model with number of children in each state dependent 85

4.1 Number of parameters in non-parametric and worst case Poisson multi-type branching processes as a function of the number of states 93
4.2 Number of directed acyclic graphs, quotient acyclic graphs and mixed acyclic graphs as function of the vertices number 98

6.1 Repartition of lines of code in *StatisKit* 126

Introduction

We address statistical models for structured data. In the case where statistical individuals are structured, graph-indexed data – based on graphs which are mathematical objects composed of vertices and edges – are used to represent and store data. Each vertex represents an elementary entity of an individual and edges represent either temporal precedence, topological or spatial adjacency between these entities. Most widespread examples of graph-indexed data are:

- Data indexed by directed path graphs. In such graphs there is an order among vertices and each vertex (except the last one), called parent vertex, is connected to the next one, called child vertex. Edges are directed and represent the order between vertices. Path-indexed data, also known as sequences or chains, are used to describe either time-evolution of individuals or topological sequences (e.g. DNA sequences or succession of nodes along plant shoots).

- Data indexed by grid graphs. In such graphs there is no order among vertices and a vertex is connected to a set of vertices, called neighbor vertices. As there is no order, edges are undirected and they represent direct connectivity of vertices. Regular grid-indexed data with a fixed size neighborhood are especially used to describe images and more generally grid graphs provide an efficient representation of spatial data.

We focus here on less common data which are indexed by directed tree graphs. Tree-indexed data can be seen as a generalization of path-indexed data since directed path graphs are directed tree graphs where there is at most one child per vertex. Let us consider the simple example of one cell followed-up throughout time. The directed path representation would enable to represent the evolution of this cell throughout time but as soon as it divides either we could consider two new paths representing the evolution of child cells, or a cell would be arbitrarily chosen as the continuation of the initial path while the other would be treated as the beginning of a new path. Since directed tree vertices can have more than one child, we are able to keep track of cell divisions using tree-indexed data where a given cell is connected to its two child cells. Among other applications in statistics, tree-indexed data have been used in multiscale representation of images (Choi and Baraniuk, 2001) or more generally signals (Crouse et al., 1998; Durand et al., 2004), cell lineages (Olariu et al., 2009) and plant representation (Durand et al., 2005). In the context of the Virtual Plants team, host team of this thesis, applications of interest focus on plant development and its modulation by environmental and genetic factors. Plants are branching living organisms that develop throughout their lifetimes. Organs are created by small embryogenetic regions at the tip of axis, called apical meristems. One of the main objectives of the Virtual Plants team is to study plant apical meristem functioning and development. Tree-indexed data can be found in this context at two complementary scales:

- At a macroscopic scale. The methodology consists in analyzing the structures produced by meristems. This can be seen as a methodology that aims at solving an

inverse problem in which one infers the meristem functioning from the whole plants they produce. Each vertex represents a botanical entity (elementary constituent of plants) and edges encode either the temporal precedence of two botanical entities produced by the same meristem or the direct lineage relationships among two meristems (branching process in plants).

- At a microscopic scale. The aim is to understand how physiological and genetic processes control meristem functioning at tissular scale. Recent scientific and technological advances in developmental biology enable to access data at the tissular scale and especially cell lineages. Each vertex represents a cell and edges encode either the tracking of a cell throughout time or the lineage relationships among parent and child cells.

While trees are closely related to paths, the catalog of models available for tree-indexed data is far less important than the one available for sequences. Historically, first interest in tree-indexed data only concerned tree topology without considering attributes attached to vertices. When considering the problem of family name extinction, Watson and Galton (1875) proposed a simple branching stochastic process only considering the topology. This process was later generalized under the name Multi-Type Branching Process (MTBP), considering both topology and categorical outcomes in tree-indexed data (see Harris, 2002). Such an improvement rendered this model applicable in many biological area (see Haccou et al., 2005; Kimmel and Axelrod, 2002, for examples). It's worth noting that these approaches are suitable for modeling tree-indexed data but were originally applied on univariate (resp. multivariate) counts data corresponding to the number of children of each vertex (resp. the number of children in each category for each parent category). The corresponding estimated distributions were therefore called generation distributions. More recently, an effort has been made to develop limit theorems (Yang, 2003) and algorithms for Markov Tree (MT) models applied to tree-indexed data with missing categorical values (Ronen et al., 1995) or non-categorical values (Crouse et al., 1998; Durand et al., 2004; Bacciu et al., 2010). MT models are stochastic processes where in the simplest case future events of the process are assumed to be independent of the past ones given the present one. Hidden Markov Tree (HMT) models introduced by Crouse et al. (1998) are to MT models what Hidden Markov Chain (HMC) models are to Markov Chain (MC) models (see Ephraim and Merhav, 2002, for a review). The basic idea of HMT models is to define an unobserved categorical MT process that is linked to the observation process by simple probabilistic mappings. Hidden Markov models are thus not restricted to categorical outcomes but can deal with multidimensional outcomes combining heterogeneous variables. Note that contrarily to path graphs where the structure is unchanged whichever the chosen direction, directed tree graphs are non-symmetrical structures. In fact, as presented by Durand et al. (2005), one can distinguish two types of MT models:

- Markov In-Trees (MITs) studied by Bacciu et al. (2010) where the edges are directed from the leaves to the root,
- Markov Out-Trees (MOTs) introduced by Ronen et al. (1995) where the edges are directed from the root to the leaves.

Due to restricted scope of applications – mostly image segmentation, signal classification/denoising or image document categorization – where the tree structure is fixed by the user, the topology is therefore not considered in such models contrarily to the MTBP case.

This thesis aims at proposing a statistical modeling framework for studying patterns in tree-indexed data. To this end, two different classes of statistical models are investigated:

- An enlarged family of Hidden Markov Out-Tree (HMOT) models is considered. Such models based on the modeling of local dependencies between child and parent vertices are particularly suited for motif detection in trees such as alternation along paths within the tree or succession of homogeneous zones concerning botanical entity fates or cell identities. In order to model highly-structured motifs, the classical HMOT model family is enlarged to take account of dependencies between children and randomness of tree structures (i.e. variable number of child vertices). In a first step, this new family of model is introduced in the case of trees with strong topological constraints (binary trees), and semi-parametric HMOT models with general observation processes are applied to cell lineages. In a second step, general trees are considered and combinatorics induced by variable and high number of children are modeled with parametric MOT models in order to obtain parsimonious models that could be applied to the HMOT case as well.

- The generalization of multiple change-point models from path-indexed data to tree-indexed data is investigated. Such models belong to the class of long-range dependency models and are particularly suited when tree-indexed data exhibit roughly homogeneous zones separated by marked change points. Application of such models enable the segmentation of tree-indexed data.

As a consequence, graphs, probabilistic graphical models and latent state models emerged as transversal thematics in this thesis.

Chapter 1 introduces graphs and graphical models. First, general definitions, properties and visualization algorithms are given. In particular, the different types of graphs used in statistical modeling – undirected, directed and mixed – are introduced. Secondly, the general graphical model framework relying on a graph for compactly encoding complex distributions is developed. Focus is therefore first on Markov and factorization properties defined in order to ensure that a distribution and its graph representation are consistent. This formalism enables the derivation of rich sets of independence assertions holding in a probability distribution by simply examining graphs or defining distributions from graphs. Nevertheless, since for a given distribution, many graph representations are consistent but not necessarily optimal, the concepts of minimal and perfect Independence maps (I-maps) are thus introduced. Afterward, the use of graphical models in terms of interpretability of models, efficiency of inference and distribution manipulation are illustrated with the example of multivariate Gaussian distributions.

In chapter 2 graphs and graphical models defined in chapter 1 are used in the particular context of directed tree graphs. This encompasses the formal definition of tree-indexed data, their visualization and presentation of statistical models available in lit-

erature to deal with such structured data. Particular emphasis is placed on the two different data sets studied in this thesis and the associated modeling issues:

- At a microscopic scale, tree-indexed data are used to represent cell lineages observed in meristems. The case of a floral meristem is considered and the objective is to recover cell identities during the first stages of morphogenesis (from the initial undifferentiated stage to the emergence of sepals). Cell identities are not directly observable but only indirectly through different geometrical, mechanical and hormonal features attached to cells. The recovery of cell identities enables to characterize the cell division process through time and to identify homogeneous regions in terms of cell identities by spatial projection.

- At a macroscopic scale, tree-indexed data are used to represent whole plants. We here consider the example of mango trees. As other tropical trees, mango trees are characterized by strong phenological asynchronisms, between and within trees, entailing patchiness. Patchiness is characterized by clumps of either vegetative, reproductive or resting botanical entities within the canopy. Latent states are therefore assimilated to patch fates, clump mostly composed of vegetative, resting or flowering botanical entities for example. In this particular case, segmenting the canopy in homogeneous regions is relevant for patch identification at a given date while motif identification allows to understand how such patterns can be set up throughout plant lifetime.

In chapter 3, we focus on models relying on local dependency assumption. An enlarged family of HMOT models is introduced in order to relax the assumption of independence between children given their parent in state-of-the-art HMOT models. As a consequence, the concept of generation distributions of MTBPs is here re-introduced into HMOT models. The upward-downward smoothing algorithm which enables to implement efficiently the E-step of the Expectation-Maximization (EM) algorithm and the dynamic programming algorithm for restoring the most probable state tree are derived. The interest of such models is illustrated on cell lineages in floral meristems where non-parametric generation distributions are coupled with parametric observation models in order to define semi-parametric HMOT models.

Cell lineages can be considered as simple tree-indexed data since there is at most two children for a vertex (and at least one child). In the 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 semi-parametric HMOT models parameters. We address the inference of discrete-state models for complex tree-structured data in chapter 4. Our aim is to introduce parametric MOT that can be efficiently estimated on the basis of data of limited size. Each generation distribution, corresponding to a discrete multivariate distribution within this macroscopic model, is modeled by a graphical model where each variable corresponds to a number of children in a given state. In order to address the inference of these generation distributions, a new method for the inference of Mixed Acyclic Graph (MAG) models is proposed. The estimation of each graphical model relies on a greedy algorithm for graph selection. The proposed modeling approach is illustrated on mango tree architecture analysis in the context of the set-up of patches within trees.

In chapter 5, the classical multiple change-point models for path-indexed data are transposed to tree-indexed data. The objective of multiple change-point models is to partition a heterogeneous tree-indexed data into homogeneous subtree-indexed data of consequent sizes. Contrarily to MT models, relying on local dependencies, multiple change-point models are relevant for tree-indexed data within which long-range dependencies have to be modeled. Since optimal algorithms of multiple change-point models for sequences cannot be transposed to trees, we propose here an efficient heuristic for tree segmentation. The segmented subtrees are grouped in a post-processing phase, and this segmentation/clustering approach is justified by the occurrence of similar disjoint patches in the canopy. Application of such models is illustrated on mango trees where subtrees are assimilated to plant patches and clusters of patches to patch types (e.g. vegetative, flowering or resting patch).

In the last chapter we focus on works currently in progress and perspectives. An originality of the Virtual Plants team is the effort dedicated to software development. All methods and models developed by team members are integrated in a common software component, *V-Plants*, within the *OpenAlea* platform (Pradal et al., 2008). This chapter gives an overview of the software resulting from the implementation of statistical models and methods developed in this thesis in order to make them available to the team members and partners. Chapter 3 focus was on HMOT models, however Hidden Markov In-Tree (HMIT) – discussed by Durand et al. (2005) and developed by Bacciu et al. (2010) – are related models that also take into account dependencies between children. Such models are therefore discussed with respect to HMOT models. Concerning the generation distributions of HMOT, we considered the use of graphical models in order to reveal exclusion and inclusion patterns in child fates. An alternative model, based on mixture models, is presented and the different hypotheses induced by these two models are hereafter discussed. Finally, we revisit the patchiness phenomenon and present an integrative analysis that could be conducted in order to decipher mango tree asynchronisms and patchiness phenomena.

Chapter 1

Graphs and graphical models frameworks

Abstract In this chapter graphs and graphical models are introduced and illustrated using Gaussian multivariate distributions.

First, general definitions, properties (e.g. topological notions, remarkable graphs...) and visualization algorithms are given for usual graphs (i.e. undirected, directed and mixed graphs).

Then, the graphical model framework is developed. This encompasses the presentation of Markovian properties, factorization properties and concepts of minimal and perfect independence maps. These properties are used to factorize multivariate distributions from a given graph, for the inquiry of independence patterns holding in such distributions and to discuss to which extent graphs proposed for distributions are relevant.

Finally, the Gaussian graphical models are discussed in order to illustrate the general concepts formerly derived. In particular advantages of graphical model in terms of parametrization and inference of Gaussian multivariate distributions are reviewed.

Keywords factorization property; Gaussian multivariate distribution; graph; graph drawing; graphical model; I-map; Markov property; quotient graph; subgraph

Contents

1.1 Introduction to graph theory 9
1.1.1 Definitions 9
1.1.2 Drawings 10
1.1.3 Graph properties 19
1.2 Graphical model framework 27
1.2.1 Random vectors and independencies 27
1.2.2 From graphs to distributions 30
1.2.3 From distributions to graphs 35
1.3 Gaussian graphical models 35
1.3.1 Parametrizations 36
1.3.2 Inference 38
References 40

1.1 Introduction to graph theory

In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to describe relations in systems consisting in many related objects. This introduction to graph theory is made in order to define and use graphs in the context of statistical analysis. It is worth noting that as many scientific fields use graphs, the terminology depends on the context. We are using here the usual terminology that can be found in statistical reference textbooks about graphical models (see Lauritzen, 1996; Koller and Friedman, 2009).

1.1.1 Definitions[1]

Graphs Let $\mathcal{G}$ be a graph. $\mathcal{G}$ is defined by a pair $(\mathcal{V}, \mathcal{E})$ where:

- The vertex set, noted $\mathcal{V}$, is a finite subset of $\mathbb{N}$.
- The edge set, noted $\mathcal{E}$, is a finite subset of $\mathcal{V} \times \mathcal{V}$ pairs of distinct vertices in $\mathcal{V}$,

$$\emptyset \subseteq \mathcal{E} \subseteq \mathcal{P}(\mathcal{V}).$$

with $\mathcal{P}(\cdot)$ the set of pairs of distinct elements of a set,

$$\mathcal{P}(\mathcal{V}) = \left\{(u, v) \in \mathcal{V}^2 \,\middle|\, u \neq v\right\}.$$

Note

We are here only considering simple graphs where no loop edge can be found,

$$\forall v \in \mathcal{V},\ (v, v) \notin \mathcal{E}.$$

For an edge $(s, t) \in \mathcal{E}$, the vertex s is called the source vertex and t the target vertex and vertices are said to be adjacent. If an edge (s, t) is in $\mathcal{E}$ and if:

- (t, s) is not in $\mathcal{E}$, it is a directed edge.
- (t, s) is also present in $\mathcal{E}$, it is an undirected edge.

Therefore, considering elements present in $\mathcal{E}$ different types of graphs can be considered:

- Undirected graphs containing only undirected edges,

$$\mathcal{G} \in \mathcal{U}(\mathcal{V}) \Rightarrow \forall (u, v) \in \mathcal{P}(\mathcal{V}),\ (u, v) \in \mathcal{E} \Leftrightarrow (v, u) \in \mathcal{E},$$

where $\mathcal{U}(\cdot)$ is the set of all undirected graphs with given vertex set.

- Directed graphs containing only directed edges,

$$\mathcal{G} \in \mathcal{D}(\mathcal{V}) \Rightarrow \forall (u, v) \in \mathcal{P}(\mathcal{V}),\ (u, v) \in \mathcal{E} \Rightarrow (v, u) \notin \mathcal{E},$$

where $\mathcal{D}(\cdot)$ is the set of all directed graphs with given vertex set.

[1] This section is largely based on Lauritzen (1996)

- Mixed graphs, containing both undirected ($\mathcal{E}'$) and directed edge sets ($\mathcal{E}''$),

$$\mathcal{G} \in \mathcal{M}(\mathcal{V}) \Rightarrow \mathcal{E} = [\mathcal{E}' \uplus \mathcal{E}''] \wedge \left[\forall (u,v) \in \mathcal{P}(\mathcal{V}), \begin{cases} (u,v) \in \mathcal{E}' \Leftrightarrow (v,u) \in \mathcal{E}' \\ (u,v) \in \mathcal{E}'' \Rightarrow (v,u) \notin \mathcal{E}'' \end{cases} \right],$$

where $\mathcal{M}(\cdot)$ is the set of all mixed graphs with given vertex set, $\uplus$ denotes the union of disjoint sets and $\wedge$ the logical *and* operator.

Note

Mixed graphs such as $\mathcal{E}' = \emptyset$ (respectively $\mathcal{E}' = \mathcal{E}$), are directed graphs (respectively undirected graphs). Mixed graphs are therefore considered as a generalization of undirected graphs and directed graphs.

Induced subgraphs A subgraph $\mathcal{G}_\mathcal{A}$ is a graph induced by a given subset $\mathcal{A}$ of the vertex set of $\mathcal{G}$. $\mathcal{G}_\mathcal{A} = (\mathcal{V}_\mathcal{A}, \mathcal{E}_\mathcal{A})$ is defined by the vertices $\mathcal{A}$ and all edges of $\mathcal{G}$ having both source and target in $\mathcal{A}$,

$$\mathcal{E}_\mathcal{A} = \mathcal{E} \cap \mathcal{P}(\mathcal{A}).$$

Quotient graphs A quotient graph $\mathcal{G}_\Pi$ is a graph induced by a given partition Π of the vertex set of $\mathcal{G}$. $\mathcal{G}_\Pi = (\mathcal{V}_\Pi, \mathcal{E}_\Pi)$ is defined by:

- Its vertex set represents an indexing of the partition blocks,

$$\Pi = \{\Pi_i\}_{i \in \mathcal{V}_\Pi}.$$

- Its edge set represents edges between partitions blocks,

$$\mathcal{E}_\Pi = \{(i,j) \in \mathcal{P}(\mathcal{V}_\Pi) \,|\, \Pi_i \times \Pi_j \cap \mathcal{E} \neq \emptyset\}.$$

1.1.2 Drawings

A strong advantage of those mathematical objects is that using some conventions, information abouts pairs $(\mathcal{V}, \mathcal{E})$ defining $\mathcal{G}$ can be easily interpreted using drawings. Such drawings are pictorial representations of the vertex and edge sets depicting the relational information encoded in graphs for visualization purposes. We will present here principal conventions used to draw those figures.

1.1.2.1 Node and link diagrams

Principle A widespread graph drawing type is the node and link diagram. Let consider a graph $\mathcal{G}$ and vertex coordinates noted $\bar{\boldsymbol{r}} = (\boldsymbol{r}_v)_{v \in \mathcal{V}}$ where

$$\forall v \in \mathcal{V}, \quad \boldsymbol{r}_v \in \mathbb{R} \times \mathbb{R},$$

as we here only consider 2 dimensional layouts. For each vertex $v \in \mathcal{V}$ one draws a circle centered on the coordinate $\boldsymbol{r}_v$ labeled v. Considering the edges set $\mathcal{E}$ one distinguishes different cases of edges:

- an undirected edge (s, t) is represented by a straight line connecting the two labeled circles s and t.

- a directed edge (s, t) is represented by an arrow pointing from the source labeled circle s to the target labeled circle t.

Given these conventions, node and link diagrams are concerned with the automatic computation of vertices coordinates in order to draw the graph. As presented in Tamassia (2007, and references therein) there exists a lot of algorithms in the literature in order to do so. Such algorithms can be separated into two principal classes:

- algorithms for small graphs (i.e. $|\mathcal{V}| \lessapprox 100$),

- algorithms for large graphs (i.e. $|\mathcal{V}| \gtrapprox 100$).

Where $|\cdot|$ denotes the cardinality of a set.

As in this thesis, we will mostly deal with small graphs, we will focus here on the former class where algorithms are intuitive, simple to implement and produce layouts that tend to be clear for graphs of small size.

Undirected graphs layouts Some of the most flexible algorithms for computing layouts of simple undirected graphs belong to the class of force-directed algorithms. Such algorithms compute the layout of a graph using mechanical models to produce layouts respecting some generally accepted criteria (see Kobourov, 2012, for more details):

1. minimize edge crossings,

2. make edge lengths uniform,

3. reflect inherent symmetry.

In the algorithm of Eades (1984) the graph is abstracted into a mechanical system composed of steel rings and springs. Each vertex is assimilated to a steel ring and each edge to a spring attached to corresponding steel rings. Therefore, once the steel rings are placed at initial positions, the system is evolving according to:

- $\vec{F}_{\bar{\mathcal{E}}}$, a force exerted between non-adjacent vertices. This repulsive force is due to an electric charge γ of steel rings and its norm is inversely proportional to the square-root of the vertices distance.

- $\vec{F}_{\mathcal{E}}$, a force exerted between adjacent vertices. This force due to strings has a norm logarithmically proportional to the vertices distance. Actually when two adjacent vertices are too close, the force is repulsive and when they are too far apart the force is attractive. The distance of equilibrium is defined by parameter β and the characteristic of springs by α.

This system tends to reach a state of minimal energy corresponding to an aesthetic layout of undirected graphs. As at each iteration all vertices are moving simultaneously, the quantity of movement computed can be damped using $f_{\boldsymbol{\Delta}}$ in order to prevent too important displacement that would not be relevant. When publishing their algorithm of undirected graphs layouts Fruchterman and Reingold (1991) added the aesthetic criterion:

4. distribute the vertices evenly.

As other algorithms developed afterward (see Kobourov, 2012, and references therein) this algorithm is very similar to the one proposed by Eades (1984). Main differences are:

- $\vec{F}_{\bar{\mathcal{E}}}$, the force exerted between non-adjacent vertices is replaced by $\vec{F}_{\mathcal{P}}$, a force applied to all pairs of vertices. This force is also a repulsive one with a norm inversely proportional for vertices distances.
- $\vec{F}_{\mathcal{E}}$, the force exerted between adjacent vertices has now a norm proportional to the square of the distance of adjacent vertices.

These two forces are inspired from forces exerted between atomic particles or celestial bodies. Norm of forces computed can be particularly excessive, they therefore proposed to limit the quantity of movements by decreasing the temperature of the system instead of damping. This last phenomenon is assimilated to the well-known annealing effect in metallurgy often used in computer science as a meta-heuristic (see Kirkpatrick et al., 1983).

We embedded these two algorithms in algorithm 1 – quadratic in time and space complexities – to compute graphs layouts (see table 1.1 and figure 1.1).

Algorithm 1 Computing vertex positions of undirected graphs

Require: $\bar{\boldsymbol{r}} = (\boldsymbol{r}_v)_{v\in\mathcal{V}}$ initial vertex positions
function ForceDirectedPlacement($\mathcal{G}$)
  for $k \in \{1, \cdots, M\}$ **do** ▷ Iterate the procedure M times
    $\bar{\boldsymbol{\Delta}} \leftarrow ((0,0))_{i\in\mathcal{V}}$ ▷ Resulting forces applied on vertex
    for $u \in \mathcal{V}$ **do**
      for $v \in \mathcal{V} \setminus \{u\}$ **do**
        if $(v,u) \in \mathcal{E}$ **then**
          $\boldsymbol{\Delta}_u \leftarrow \boldsymbol{\Delta}_u + \vec{F}_{\mathcal{E}}(\boldsymbol{r}_u, \boldsymbol{r}_v)$
          $\boldsymbol{\Delta}_v \leftarrow \boldsymbol{\Delta}_v - \vec{F}_{\mathcal{E}}(\boldsymbol{r}_u, \boldsymbol{r}_v)$
        else
          $\boldsymbol{\Delta}_u \leftarrow \boldsymbol{\Delta}_u + \vec{F}_{\bar{\mathcal{E}}}(\boldsymbol{r}_u, \boldsymbol{r}_v)$
        $\boldsymbol{\Delta}_u \leftarrow \boldsymbol{\Delta}_u + \vec{F}_{\mathcal{P}}(\boldsymbol{r}_u, \boldsymbol{r}_v)$
    for $u \in \mathcal{V}$ **do**
      $\boldsymbol{r}_u \leftarrow \boldsymbol{r}_u + \tilde{\boldsymbol{\Delta}}_u \min\left(f_{\boldsymbol{\Delta}}(\boldsymbol{\Delta}_u), f_k(k)\right)$
  return $\bar{\boldsymbol{r}}$

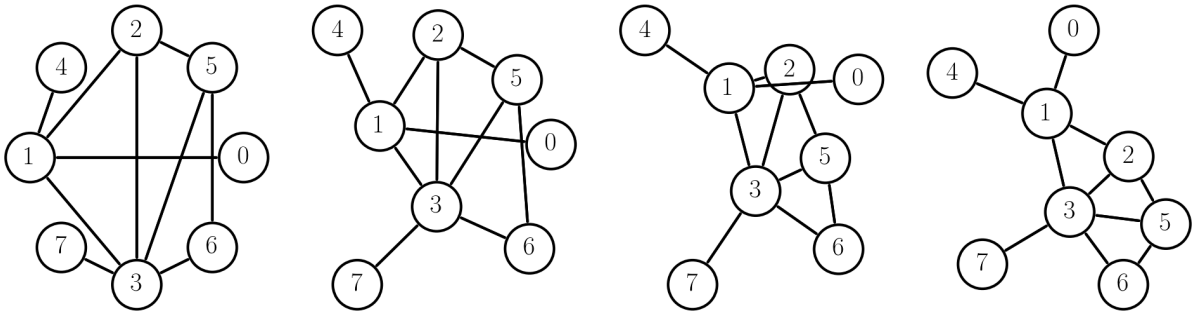


Figure 1.1 – Examples of force-directed undirected graphs layout evolution. From left to right the initial positions of vertices and then, the results of the second, tenth and hundredth iterations are draw. Starting from given positions, the graph is viewed as a spring system which evolves to reach a stable configuration. Algorithm 1 have been configured to used the Fruchterman and Reingold (1991) algorithm but with the Eades (1984) forces $\vec{F}_{\mathcal{E}}$ and $\vec{F}_{\bar{\mathcal{E}}}$ respectively in place of $\vec{F}_{\mathcal{E}}$ and $\vec{F}_{\mathcal{P}}$ forces.

Directed graphs and mixed graphs layouts While force-directed algorithms are widely used for drawing undirected graphs, there are less used in the context of directed graphs. The principal reason is that they do not produce layouts highlighting the direction of edges. More sophisticated algorithms – known as hierarchical drawing algorithms (see Tamassia, 2007, chapter 13) – were therefore developed for directed graphs in order to take account of hierarchies generally encoded in directed graphs into the drawing. Nevertheless, such algorithms require that all edges are oriented and can therefore not be used for mixed graphs without a preliminary transformation of undirected edges into directed ones which introduces in the drawing an erroneous impression of hierarchy.

When one is limited to small directed graphs or mixed graphs, the extension of force-directed algorithms presented by Sugiyama and Misue (1995) is very interesting. In order to suit force-directed algorithms to directed graphs and mixed graphs they added to the list of aesthetic criteria the following one:

5. conform links to specified orientations.

They integrated this aesthetic criterion by considering magnetic fields. Actually, each spring in the physical model is henceforth magnetized and the system is evolving in a magnetic field defined by:

- a space dependent orientation vector noted $\omega(.)$ (see table 1.2),
- a strength noted $\nu \in [0, 1]$,

with:

- undirected edges as bi-directional magnetic springs,
- directed edges as uni-directional magnetic springs.

Eades (1984)	Fruchterman and Reingold (1991)
$\vec{F}_{\mathcal{E}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto \alpha \cdot \log\left(\|\boldsymbol{\delta}\|/\beta\right) \cdot \tilde{\boldsymbol{\delta}}$	$\vec{F}_{\mathcal{E}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto \kappa \cdot \|\boldsymbol{\delta}\|^2 \cdot \tilde{\boldsymbol{\delta}}$
$\vec{F}_{\bar{\mathcal{E}}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto -\gamma/\|\boldsymbol{\delta}\| \cdot \tilde{\boldsymbol{\delta}}$	$\vec{F}_{\bar{\mathcal{E}}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto 0$
$\vec{F}_{\mathcal{P}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto \mathbf{0}$	$\vec{F}_{\mathcal{P}} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ $\boldsymbol{r}_u, \boldsymbol{r}_v \mapsto -\tilde{\boldsymbol{\delta}}/\kappa^2 \cdot \|\boldsymbol{\delta}\|$
$f_{\boldsymbol{\Delta}} : \mathbb{R}^2 \to \mathbb{R}$ $\boldsymbol{\Delta} \mapsto \epsilon \cdot \|\boldsymbol{\Delta}\|$	$f_{\boldsymbol{\Delta}} : \mathbb{R}^2 \to \mathbb{R}$ $\boldsymbol{\Delta} \mapsto \|\boldsymbol{\Delta}\|$
$f_k : \mathbb{N} \to \mathbb{R}$ $k \mapsto +\infty$	$f_k : \mathbb{N} \to \mathbb{R}$ $k \mapsto \lambda \cdot \exp\left(-k/\tau\right)$

Table 1.1 – Forces used in classical force-directed algorithms. The notation $\|\cdot\|$ denotes the norm of a vector, $\tilde{\cdot}$ its normalization and $\boldsymbol{\delta} = \boldsymbol{r}_u - \boldsymbol{r}_v$ the vector from vertex v to vertex u. $\vec{F}_{\mathcal{P}}$ (resp. $\vec{F}_{\mathcal{E}}$, $\vec{F}_{\bar{\mathcal{E}}}$) is a force applied to all pairs of vertices (resp. adjacent vertices, non-adjacent vertices). Since the relevance of quantity of movement computed for a vertex is limited to a certain amount, functions $f_{\boldsymbol{\Delta}}$ (resp. f_k), limiting the norm of forces applied, are introduced. Usual values for parameters are $\alpha = 2.0$, $\beta = 1.0$, $\gamma = 1.0$ $\epsilon = 0.1$, $\kappa = 1.0$, $\lambda = 273.0$ and $\tau = 10.0$ for a number of iterations $M = 100$.

The magnetic force applied is the following

$$\begin{array}{rccl} \vec{F}_m : & \mathbb{R}^2 \times \mathbb{R}^2 & \to & \mathbb{R}^2 \\ & \boldsymbol{r}_u, \quad \boldsymbol{r}_v & \mapsto & -\begin{pmatrix} 1-\cos(\nu\theta) & -\sin(\nu\theta) \\ \sin(\nu\theta) & 1-\cos(\nu\theta) \end{pmatrix} \boldsymbol{\delta}, \end{array}$$

where θ is the direct angle between the vectors

$$\boldsymbol{\delta} = \boldsymbol{r}_u + \frac{\boldsymbol{r}_u + \boldsymbol{r}_v}{2},$$

representing the second half of edge (u, v) and

$$\omega\left(\frac{\boldsymbol{r}_u + \boldsymbol{r}_v}{2}\right),$$

the orientation of the field at the middle of the edge. The introduction of this magnetic force $\vec{F}_m$ do not change the algorithm 1 as $\vec{F}_{\mathcal{E}}$ is henceforth considered as the resulting force of the one chosen in the undirected case and the magnetic force induced by the

Field name	Orientation vector
South parallel	$\begin{aligned}\omega : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \boldsymbol{r} &\mapsto (0,-1)\end{aligned}$
Centrifugal polar	$\begin{aligned}\omega : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \boldsymbol{r} &\mapsto \tilde{\boldsymbol{r}}\end{aligned}$
Clockwise concentric	$\begin{aligned}\omega : \mathbb{R}^2 &\to \mathbb{R}^2 \\ \boldsymbol{r} &\mapsto \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}\tilde{\boldsymbol{r}}\end{aligned}$

Table 1.2 – Usual magnetic fields for force-directed algorithms (Sugiyama and Misue, 1995). $\boldsymbol{r}$ denotes the position considered in magnetic field and $\tilde{\boldsymbol{r}}$ its renormalization. The usual value for the field strength parameter is $\nu = 0.1$.

field (see figure 1.2). As a consequence of this rotational force, directed edges tend to be in the direction of the magnetic field whereas undirected ones in the orthogonal one (see figure 1.3 and figure 1.4).

Note

The force presented here is not the same as the one presented by Sugiyama and Misue (1995). The main reason of this modification is that considering their force the edge lengths are highly modified by the magnetic field. Therefore results produced are in contradiction with the aesthetic criteria '2. make edge lengths uniform' and this behavior is reinforced as soon as the force of the field is increased. The force we propose produces for an edge (u, v) a rotation at the center of the edge of an angle proportional:

- to the angle between the edge and the orientation vector,
- to the force of the field.

without modifying the edge length.

Initialization In this thesis, small graphs drawings are achieved using algorithms 1 for vertex positioning and a customized interface to the matplotlib package (Hunter, 2007) for the drawing production. As force-directed algorithms are highly sensitive to the initial vertex positions we considered the following strategies:

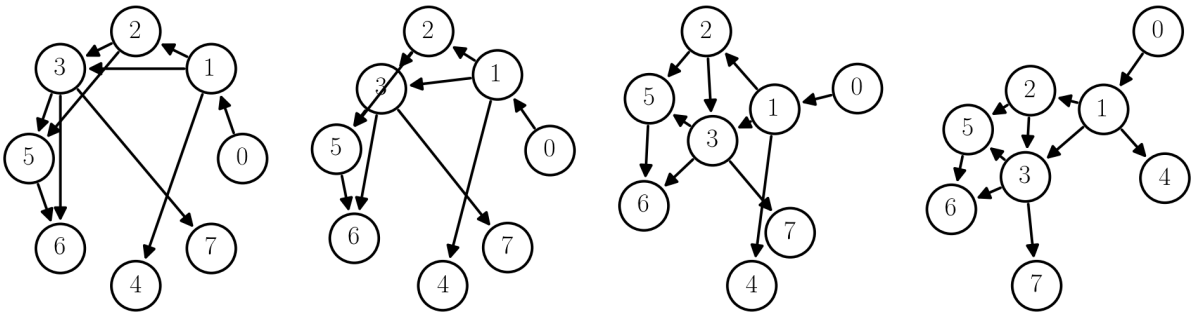


Figure 1.2 – Examples of force-directed directed graphs layout evolution. From left to right the initial positions of vertices and then, the results of the second, tenth and hundredth iterations are draw. Starting from given positions, the graph is viewed as a magnetized spring system which evolve to reach a stable configuration given a magnetic field (the parallel one in this case).

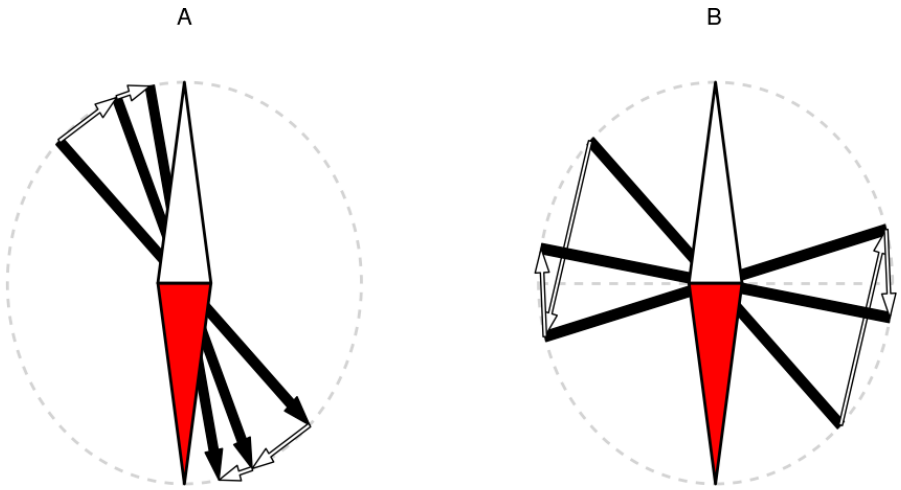


Figure 1.3 – Examples of movements induced by a magnetic field. The field orientation vector is represented by a compass directed from the south (in white) to the north pole (in red). Forces resulting from the magnetic field are represented by white arrows applied at both extremities of edges. On (A) the successive movements of a directed edge are represented and on (B) the movements of an undirected one. The directed edge converges to the direction of the magnetic filed while the undirected one oscillates around the orthogonal direction.

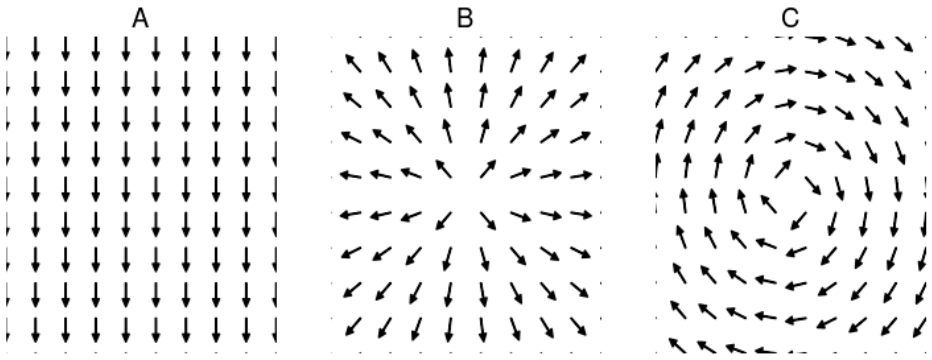


Figure 1.4 – Examples of magnetic fields. (A) The south parallel field. (B) The polar one. (C) The clockwise concentric one. The ideal direction of directed edges are represented at given positions considering 3 *of the magnetic fields proposed by Sugiyama and Misue (1995).*

- random, each coordinates is randomly set on the square $\left[\sqrt{|\mathcal{V}|}/2, \sqrt{|\mathcal{V}|}/2\right]^2$.
- circular, each coordinates is deterministically and evenly set on the circle centered at $(0, 0)$ and of radius $\sqrt{|\mathcal{V}|}$. Instead of considering the vertices with their natural ordering, relevant ordering of vertices (see Tarjan, 1972, for more details and in particular the Depth-First Search (DFS) ordering) based on adjacencies can be used in order to improve this initialization.

It is worth noting that if algorithm 1 is automatically producing a nice layout, the result could be furthermore improved by manual corrections. We therefore configured our matplotlib interface in order to allow *a posteriori* vertex position corrections while updating link conformations.

1.1.2.2 Matrix plots

The utility of the basic force-directed algorithms to compute vertex positions is limited to small graphs and results are of poor quality for graphs with more than a few hundred vertices. They are multiple reasons why the traditional force-directed approach does not perform well for large graphs. One of the main obstacles to the scalability of these approaches is the fact that the underlying physical model has many local minima. In cases of large graphs another useful drawing approach is inspired from matrix drawings. A graph $\mathcal{G}$ can be represented using $\mathcal{V} \times \mathcal{V}$ square matrices noted $\underline{\mathcal{G}}$:

- The adjacency matrix of a graph of general element $\underline{\mathcal{G}}_{u,v}$ defined as follows

$$\forall\,(u,v) \in \mathcal{V}^2, \quad \begin{cases} \underline{\mathcal{G}}_{u,v} = 1 & \text{if} \quad (u,v) \in \mathcal{E}, \\ \underline{\mathcal{G}}_{u,v} = 0 & \text{otherwise.} \end{cases}$$

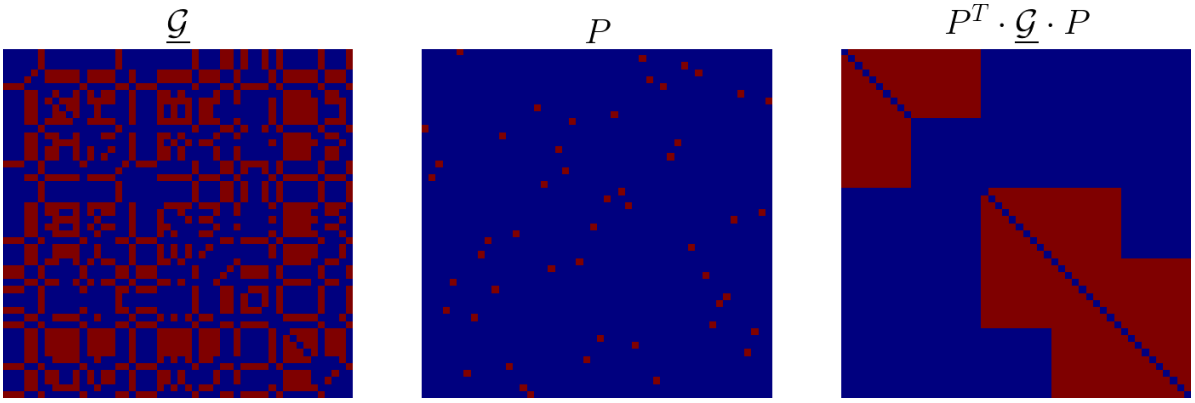


Figure 1.5 – Adjacency matrix drawing. Here an adjacency matrix $\underline{\mathcal{G}}$, a permutation matrix P and the corresponding rows and columns permutation of the adjacency matrix of an undirected graph are represented. If the adjacency matrix describes the topological information encoded in the graph it can be relevant to rearrange the rows and columns of this matrix in order to highlight structural information. Whereas the adjacency matrix $\underline{\mathcal{G}}$ of the graph with 50 vertices does not reveal particular topological information, the permuted matrix corresponding to a permutation of vertex labels highlights the presence of 5 clusters of vertices.

- The incidence matrix of a graph $\mathcal{G}$ of general element $\underline{\mathcal{G}}_{u,v}$ defined as follows

$$\forall\,(u,v)\in\mathcal{V}^2,\quad \begin{cases} \underline{\mathcal{G}}_{u,v}=1 & \text{if} \quad [(u,v)\in\mathcal{E}\wedge(v,u)\in\mathcal{E}]\vee[(u,v)\in\mathcal{E}\wedge(v,u)\notin\mathcal{E}]\,,\\ \underline{\mathcal{G}}_{u,v}=-1 & \text{if} \quad [(u,v)\notin\mathcal{E}]\wedge[(v,u)\in\mathcal{E}]\,,\\ \underline{\mathcal{G}}_{u,v}=0 & \text{otherwise,}\end{cases},$$

  where $\vee$ denotes the logical *or* operator.

To represent these matrices a square surface of $\mathcal{V}^2$ pixels is considered and each pixel located at position $(u,v)\in\mathcal{V}^2$ is colored according to the value of the element $\underline{\mathcal{G}}_{(u,v)}$. In order to map the different values of $\underline{\mathcal{G}}$ to colors, we used color-maps defined in matplotlib package (Hunter, 2007, see figure 1.5).

Let consider a permutation $\sigma(.)\in\mathfrak{S}(\mathcal{V})$ where $\mathfrak{S}(\cdot)$ is the set of all permutations of a set. Graph adjacency matrix drawing can sometimes be enhanced by the drawing of a permuted version of the matrix $P^t\cdot\underline{\mathcal{G}}\cdot P$ where the permutation matrix P general element $P_{u,v}$ is defined as follows

$$\forall\,(u,v)\in\mathcal{V}^2,\quad \begin{cases} P_{u,v}=1 & \text{if} \quad v=\sigma(u)\\ P_{u,v}=0 & \text{otherwise.}\end{cases}$$

As for the circular initialization for node and link diagrams such relevant permutations are related to vertex ordering (Tarjan, 1972) induced by the topological information encoded in graphs which we will further illustrate.

1.1.3 Graph properties[2]

Mixed graphs are a generalization of undirected graphs and directed graphs but for the sake of understanding concepts regarding graphs will be first presented for undirected graphs and directed graphs. In order to generalize these definitions to mixed graphs, an effort has been made to define undirected graphs and directed graphs notions in a manner that they hold for all types of graphs.

1.1.3.1 Undirected graphs

Topological notions Two distinct vertices u, $v \in \mathcal{V}$ are said to be neighbors if the edges (u, v) and (v, u) are present in $\mathcal{E}$. The set of neighbors of vertex is noted $\mathrm{ne}(\cdot)$ and its cardinality – called degree – $\deg(\cdot)$,

$$\begin{aligned} \forall v \in \mathcal{V},\, \mathrm{ne}(v) &= \{u \in \mathcal{V} \,|\, [(u, v) \in \mathcal{E}] \wedge [(v, u) \in \mathcal{E}]\}, \\ \deg(v) &= |\mathrm{ne}(v)|. \end{aligned}$$

For a subset $\mathcal{A}$ of $\mathcal{V}$, its neighborhood is defined as the union of the neighborhood of each of its elements discarding its own elements,

$$\forall \mathcal{A} \subseteq \mathcal{V},\, \mathrm{ne}(\mathcal{A}) = \{\cup_{v \in \mathcal{A}} \mathrm{ne}(v)\} \setminus \mathcal{A},$$

and its closure is noted $\mathrm{Ne}(\cdot)$,

$$\forall \mathcal{A} \subseteq \mathcal{V},\, \mathrm{Ne}(\mathcal{A}) = \mathrm{ne}(\mathcal{A}) \cup \mathcal{A}.$$

The subset $\mathcal{A} \subseteq \mathcal{V}$ such that in the subgraph $\mathcal{G}_{\mathcal{A}}$ all vertices have all the other vertices as neighbors is a clique. When $\mathcal{A}$ is a clique such that for any other vertex $i \in \mathcal{V} \setminus \mathcal{A}$, $\mathcal{A} \cup \{i\}$ is no more a clique, $\mathcal{A}$ is said to be a maximal clique. The set of all maximal cliques in the undirected graph $\mathcal{G}$ is noted $\mathcal{K}_{\mathcal{G}}$. The notion of maximal clique is central in undirected graphs and the listing of all maximal cliques can be done using the Bron and Kerbosch (1973) algorithm which has a worst case time complexity of $\mathcal{O}\left(3^{\frac{K}{3}}\right)$ (Tomita et al., 2006). Although other algorithms for computing $\mathcal{K}_{\mathcal{G}}$ have been designed since 1973 this algorithm and optimized variants are reported as being more efficient in practice than the alternatives (Cazals and Karande, 2008).

A path of length l from a vertex $u \in \mathcal{V}$ toward a vertex $v \in \mathcal{V}$ is a sequence $\alpha_0 = u, \ldots, \alpha_{l-1} = v$ of distinct vertices such as $(\alpha_{k-1}, \alpha_k) \in \mathcal{E}$ for all $k \in [\![0, l[\![$. If there is a path from a vertex $u \in \mathcal{V}$ to a vertex $v \in \mathcal{V}$ and another one from v to u, vertices u and v are said to be connected. The set of vertices connected to a vertex, noted $\mathrm{cn}(\cdot)$, is defined as follows

$$\forall v \in \mathcal{V}, \quad \mathrm{cn}(v) = \left[\left\{\cup_{u \in \mathrm{ne}(v)} \mathrm{cn}(u)\right\} \cup \mathrm{ne}(v)\right] \setminus \{v\}.$$

For a subset $\mathcal{A}$ of $\mathcal{V}$, its connected vertex set is defined as the union of vertices connected to each of its elements discarding its own elements,

$$\forall \mathcal{A} \subseteq \mathcal{V}, \quad \mathrm{cn}(\mathcal{V}) = \{\cup_{v \in \mathcal{A}} \mathrm{cn}(v)\} \setminus \mathcal{A},$$

[2] This section is largely based on Lauritzen (1996)

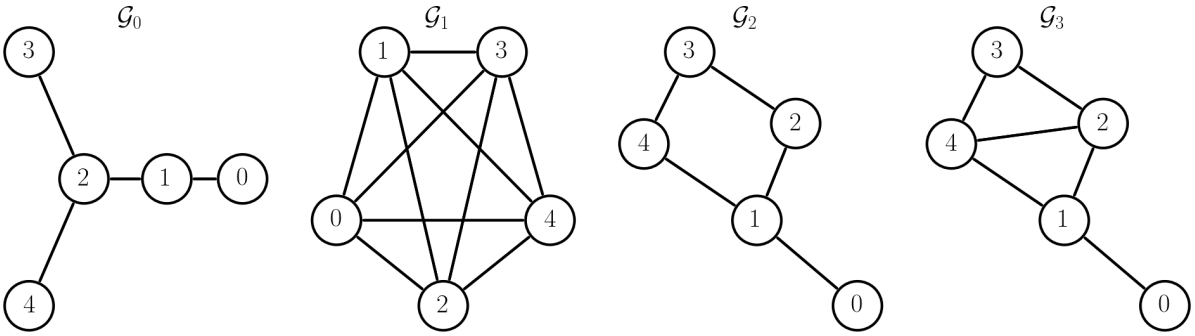


Figure 1.6 – Remarkable undirected graphs. Graph $\mathcal{G}_0$ is a tree and $\mathcal{G}_1$ a complete graph. Graph $\mathcal{G}_2$ is not chordal as there is a diamond shape 1, 2, 3, 4 *missing a chord* $(1, 3)$ *and/or* $(2, 4)$*. Graph $\mathcal{G}_3$ is one of the chordal embedding of $\mathcal{G}_2$ where the chord* $(2, 4)$ *has been added. The* 4*-cycle* 1, 2, 3, 4 *has therefore been split in* 2 3*-cycles* 1, 2, 4 *and* 2, 4, 3.

and its closures is noted $\mathrm{Cn}(\cdot)$,

$$\forall \mathcal{A} \subseteq \mathcal{V}, \quad \mathrm{Cn}(\mathcal{V}) = \mathrm{cn}(\mathcal{A}) \cup \mathcal{A}.$$

A graph such that every vertex is connected to the others is said to be a connected graph. A connected component of an undirected graph $\mathcal{G}$ is a subset $\mathcal{A}$ of $\mathcal{V}$ such that $\mathcal{G}_{\mathcal{A}}$ is connected and that every every vertex of $\mathcal{A}$ is not connected to any vertices in $\mathcal{V} \setminus \mathcal{A}$.

A path of length $l > 2$ pointing from a vertex v to the same vertex v is a l-cycle. A chord is an edge linking 2 non-consecutive vertices in a cycle. A diamond shape is a l-cycle with $l \geq 4$ containing no chords.

Remarkable graphs Considering edge sets, important classes of undirected graphs can be defined (see figure 1.6).

An undirected graph $\mathcal{G}$ is said to be a complete graph if $\mathcal{V}$ is a clique

$$\mathcal{K}_{\mathcal{G}} = \{\mathcal{V}\}.$$

If an undirected graph has no l-cycles and is connected, it is a tree. If it has more than one connected component but all connected components are trees, it is a forest.

If any l-cycles for $l > 3$ in a graph have a chord then the graph is a chordal graph. By extension undirected graphs with no cycles are also said to be chordal. In fact, if an undirected graph does not containing any diamond shape, it is chordal. A chordal embedding $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ of a non-chordal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a chordal graph such as $\mathcal{E} \subset \mathcal{E}'$. Computing a minimal chordal embedding – a chordal embedding such as the number of edges added is minimal – of a non-chordal graph is NP-hard (Yannakakis, 1981), a good heuristic for computing a chordal embedding is to generate a well chosen ordering of vertices such as the number of chords added is reasonable (see Rose et al., 1976; Amestoy et al., 1996; Berry et al., 2004, for more details).

Separation property For three disjoints subsets of $\mathcal{V}$, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, the subset $\mathcal{C}$ is said to be $(\mathcal{A},\mathcal{B})$-separator if all paths from vertices in $\mathcal{A}$ to vertices in $\mathcal{B}$ intersect vertices in $\mathcal{C}$. This property is noted

$$\mathcal{A} \overset{G}{\perp} \mathcal{B}|\mathcal{C}.$$

When no confusion could arise the notations $\mathcal{A} \perp \mathcal{B}|\mathcal{C}$ if $\mathcal{C}$ is not the empty set, or $\mathcal{A} \perp \mathcal{B}$ otherwise, will be used. If $\mathcal{C}$ is $(\mathcal{A},\mathcal{B})$-separator and for any vertices $i \in \mathcal{C}$, $\mathcal{C} \setminus \{i\}$ is no more $(\mathcal{A},\mathcal{B})$-separator, $\mathcal{C}$ is a minimal $(\mathcal{A},\mathcal{B})$-separator.

When considering a fourth disjoint subset of $\mathcal{D} \subset \mathcal{V}$, some important properties can be derived from graph separation:

- The symmetry property,

$$\mathcal{A} \perp \mathcal{B} \,|\, \mathcal{C} \Leftrightarrow \mathcal{B} \perp \mathcal{A} \,|\, \mathcal{C}.$$

- The decomposition property,

$$\mathcal{A} \perp \mathcal{B} \cup \mathcal{C} \,|\, \mathcal{D} \Rightarrow \mathcal{A} \perp \mathcal{B} \,|\, \mathcal{D}.$$

- The weak union property,

$$\mathcal{A} \perp \mathcal{B} \cup \mathcal{C} \,|\, \mathcal{D} \Rightarrow \mathcal{A} \perp \mathcal{B} \,|\, \mathcal{D} \cup \mathcal{C}.$$

- The contraction property,

$$[\mathcal{A} \perp \mathcal{B} \,|\, \mathcal{D}, \mathcal{C}] \wedge [\mathcal{A} \perp \mathcal{C} \,|\, \mathcal{D}] \Rightarrow \mathcal{A} \perp \mathcal{B}, \mathcal{C} \,|\, \mathcal{D}.$$

- The intersection property,

$$[\mathcal{A} \perp \mathcal{B} \,|\, \mathcal{C}] \wedge [\mathcal{A} \perp \mathcal{C} \,|\, \mathcal{B}] \Rightarrow \mathcal{A} \perp \mathcal{B}, \mathcal{C}.$$

Let $\mathcal{S}(\mathcal{G})$ denote the set that contains all the separations that holds in $\mathcal{G}$,

$$\mathcal{S}(\mathcal{G}) = \left\{ (\mathcal{A},\mathcal{B},\mathcal{C}) \in \mathfrak{P}^3(\mathcal{V}) \,\middle|\, \mathcal{A} \perp \mathcal{B} \,|\, \mathcal{C} \right\}.$$

For any two undirected graphs $\mathcal{G}$ and $\mathcal{G}'$ sharing the same vertex set, if they do not have the same set of edges, their sets of separations differ,

$$\forall (\mathcal{G}, \mathcal{G}') \in \mathcal{U}^2(\mathcal{V}), \mathcal{E} \neq \mathcal{E}' \Leftrightarrow \mathcal{S}(\mathcal{G}) \neq \mathcal{S}(\mathcal{G}').$$

1.1.3.2 Directed graphs

Topological notions A vertex s is said to parent of a vertex t if the edge (s,t) is in $\mathcal{E}$ but not the edge (t,s). Correspondingly t is said to be a child of the vertex s. One denotes by:

- $\mathrm{pa}(\cdot)$, the set of parents of a vertex,

$$\begin{aligned}\forall v \in \mathcal{V},\, \mathrm{pa}(v) &= \{u \in \mathcal{V} \,|\, [(u,v) \in \mathcal{E}] \wedge [(v,u) \notin \mathcal{E}]\},\\ \deg^-(v) &= |\mathrm{pa}(v)|,\end{aligned}$$

  where $\deg^-(\cdot)$ is the in-degree (i.e. number of parents) of a vertex.

- $\mathrm{an}(\cdot)$ the set of ancestors of a vertex,

$$\forall v \in \mathcal{V}, \quad \mathrm{an}(v) = \left\{\cup_{u \in \mathrm{pa}(v)} \mathrm{an}(u)\right\} \cup \mathrm{pa}(v).$$

- $\mathrm{ch}(\cdot)$, the set of children of a vertex,

$$\begin{aligned}\forall v \in \mathcal{V}, \quad \mathrm{ch}(v) &= \{u \in \mathcal{V} \,|\, (v,u) \in \mathcal{E} \wedge (u,v) \notin \mathcal{E}\}.\\ \deg^+(v) &= |\mathrm{ch}(v)|,\end{aligned}$$

  where $\deg^+(\cdot)$ is the out-degree (i.e. number of children) of a vertex.

- $\mathrm{de}(\cdot)$, the set of descendants of a vertex,

$$\forall v \in \mathcal{V}, \quad \mathrm{de}(v) = \left\{\cup_{u \in \mathrm{ch}(v)} \mathrm{de}(u)\right\} \cup \mathrm{ch}(v).$$

- $\mathrm{nd}(\cdot)$, the set of non-descendants of a vertex,

$$\forall v \in \mathcal{V}, \quad \mathrm{nd}(v) = \mathcal{V} \setminus [\mathrm{de}(v) \cup \{v\}].$$

Similarly, for any subset $\mathcal{A}$ of $\mathcal{V}$, same notations are used,

$$\begin{aligned}\mathrm{pa}(\mathcal{A}) &= \{\cup_{v \in \mathcal{A}} \mathrm{pa}(v)\} \setminus \mathcal{A},\\ \mathrm{ch}(\mathcal{A}) &= \{\cup_{v \in \mathcal{A}} \mathrm{ch}(v)\} \setminus \mathcal{A},\\ \mathrm{an}(\mathcal{A}) &= \{\cup_{v \in \mathcal{A}} \mathrm{an}(v)\} \setminus \mathcal{A},\\ \mathrm{de}(\mathcal{A}) &= \{\cup_{v \in \mathcal{A}} \mathrm{de}(v)\} \setminus \mathcal{A},\end{aligned}$$

and capitalized ones for their closures,

$$\begin{aligned}\mathrm{Pa}(\mathcal{A}) &= \mathrm{pa}(\mathcal{A}) \cup \mathcal{A},\\ \mathrm{Ch}(\mathcal{A}) &= \mathrm{ch}(\mathcal{A}) \cup \mathcal{A},\\ \mathrm{An}(\mathcal{A}) &= \mathrm{an}(\mathcal{A}) \cup \mathcal{A},\\ \mathrm{De}(\mathcal{A}) &= \mathrm{de}(\mathcal{A}) \cup \mathcal{A},\\ \mathrm{Nd}(\mathcal{A}) &= \mathrm{nd}(\mathcal{A}) \cup \mathcal{A}.\end{aligned}$$

The set of roots, noted $\mathcal{R}$, is the set of vertices with no parents,

$$\mathcal{R} = \left\{ v \in \mathcal{V} \,\middle|\, \deg^-(v) = 0 \right\},$$

and the set of leaves, noted $\mathcal{L}$, is the set of vertices with no children,

$$\mathcal{L} = \left\{ v \in \mathcal{V} \,\middle|\, \deg^+(v) = 0 \right\},$$

A directed path of length l from a vertex $u \in \mathcal{V}$ toward a vertex $v \in \mathcal{V}$ is a sequence $\alpha_0 = u, \ldots, \alpha_{l-1} = v$ of vertices such as $(\alpha_{k-1}, \alpha_k) \in \mathcal{E}$ for all $k \in [\![0, l[\![$ and there is at least one $(\alpha_k, \alpha_{k-1}) \notin \mathcal{E}$. A directed l-cycle is a directed path of length $l > 1$ from a vertex $v \in \mathcal{V}$ to the same vertex v.

A v-shape is a set of 3 distinct vertices u, v and w of $\mathcal{V}$ such that (see $\mathcal{G}_{\{1,2,3\}}$ in figure 1.7):

- w is a child of u and v,
- u is not a child of v and conversely.

In a directed graph a v-shape is also called an immorality. On denotes by $\mathcal{I}_\mathcal{G}$ the set of immoralities in $\mathcal{G}$ defined as follows

$$\mathcal{I}_\mathcal{G} = \left\{ (u, v, w) \in \mathcal{V}^3 \,\middle|\, [u \notin \mathrm{pa}(v)] \wedge [v \notin \mathrm{pa}(u)] \wedge [w \in ch(u)] \wedge [w \in ch(v)] \right\}.$$

Remarkable graphs Considering the edge set it is possible to define important classes of directed graphs (see figure 1.7).

A Directed Acyclic Graph (DAG) is a directed graph which does not contain any directed l-cycles. By opposition directed cyclic graphs are directed graphs that are not DAGs.

A directed forest is a DAG such that none of its vertices has an in-degree superior to 1 and at least one vertex has an in-degree equal to 0. Moreover, if a directed forest has only one vertex having a null in-degree, it is a directed tree.

There are two transformations of a directed graph $\mathcal{G}$ into an undirected graph generally considered:

- Its undirected version noted $\mathcal{G}^u = (\mathcal{V}, \mathcal{E}^u)$, is the undirected graph obtained by dropping edge directions,

  $$\mathcal{E}^u = \mathcal{E} \cup \mathcal{E}^r,$$

  where $\mathcal{E}^r$ is the set of reversed edges,

  $$\mathcal{E}^r = \{(t, s) \in \mathcal{P}(\mathcal{V}) \setminus \mathcal{E} \,|\, (s, t) \in \mathcal{E}\}.$$

- Its moral graph noted $\mathcal{G}^m = (\mathcal{V}, \mathcal{E}^m)$, is the undirected graph obtained by adding all edges corresponding to immoralities into its undirected version,

  $$\mathcal{E}^m = \mathcal{E}^u \cup \left\{ (u, v) \in \mathcal{P}(\mathcal{V}) \,\middle|\, \exists w \in \mathcal{V},\ (u, v, w) \in \mathcal{I}_\mathcal{G} \right\}.$$

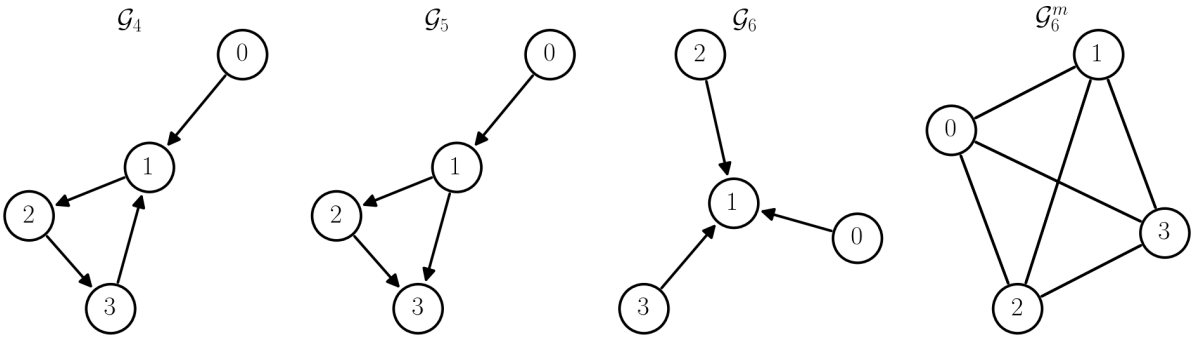


Figure 1.7 – Remarkable directed graphs. Contrary to graph $\mathcal{G}_4$ in which we have the directed 3-cycle $1,2,3,1$, graph $\mathcal{G}_5$, where the only difference is the reversal of the edge $(3,1)$ into $(1,3)$, is a DAG. Graph $\mathcal{G}_6$ is a DAG with 3 immoralities $(0,3)$, $(2,3)$ and $(0,2)$ therefore many those edges were added into its undirected version in order to build its moral graph $\mathcal{G}_6^m$, which is complete.

Separation property Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be three disjoints subsets of $\mathcal{V}$. The subset $\mathcal{C}$ is said to be $(\mathcal{A},\mathcal{B})$-d-separator if $\mathcal{C}$ is $(\mathcal{A},\mathcal{B})$-separator in $\mathcal{G}^m_{an(\mathcal{A}\cup\mathcal{B}\cup\mathcal{C})}$[3],

$$\mathcal{A} \underset{d}{\overset{\mathcal{G}}{\perp}} \mathcal{B} \,|\, \mathcal{C} \Rightarrow \mathcal{A} \overset{\mathcal{G}^m_{an(\mathcal{A}\cup\mathcal{B}\cup\mathcal{C})}}{\perp} \mathcal{B} \,|\, \mathcal{C}.$$

When no confusion could arise the notations $\mathcal{A} \perp \mathcal{B}|\mathcal{C}$ if $\mathcal{C}$ is not the empty set, or $\mathcal{A} \perp \mathcal{B}$ otherwise, will be used.

If as in the undirected graph case the same important properties can be derived from graph d-separation, there are few important remarks to be done concerning the set of separations in a directed graph:

- The set of separations of an undirected graph can be represented by a directed graph with no immoralities if and only if the undirected graph graph is chordal. Such graphs are said to be Separation equivalent (S-equivalent). A conversion from a chordal graph into a directed graph can be made by considering a vertex as the center of the undirected graph and orienting edges in a centrifugal way. Conversely, as soon as a directed graph has an immorality it has no S-equivalent in the undirected graph space (see figure 1.8).

- Contrarily to undirected graphs, for any two directed graphs $\mathcal{G}$ and $\mathcal{G}'$ sharing the same vertex set, if they do not have the same set of edges, their sets of separations do not necessarily differ,

  $$\forall\,(\mathcal{G},\mathcal{G}') \in \mathcal{D}^2\,(\mathcal{V})\,,\ \mathcal{E} \neq \mathcal{E}' \not\Leftrightarrow\ \mathcal{S}\,(\mathcal{G}) \neq \mathcal{S}\,(\mathcal{G}')\,.$$

  Actually, these two directed graphs are S-equivalent if and only if they have the same undirected version and the same set of v-shapes,

  $$\forall\,(\mathcal{G},\mathcal{G}') \in \mathcal{D}^2\,(\mathcal{V})\,,\ \mathcal{S}\,(\mathcal{G}) = \mathcal{S}\,(\mathcal{G}') \Leftrightarrow \begin{cases} \mathcal{E}^u = \mathcal{E}'^u, \\ \mathcal{I}_{\mathcal{G}} = \mathcal{I}_{\mathcal{G}'}. \end{cases}$$

[3]The subgraph operator has the precedence over the moralization

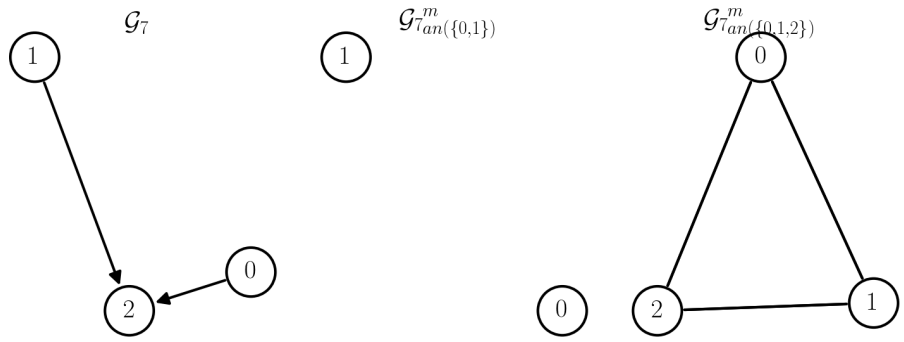


Figure 1.8 – The d-separation property. In graph $\mathcal{G}_7$ as the vertices 0 and 1 are separated in $\mathcal{G}^m_{7_{an(\{0,1\})}}$, $1 \perp 0$ does hold but not $1 \perp 0 \,|\, 2$ as the immorality $(0,1)$ in $\mathcal{G}^m_{7_{an(\{0,1,2\})}}$ is moralized thus vertices 0 and 1 are not separated by vertex 2.

1.1.3.3 Mixed graphs

Topological notions Mixed graphs can be viewed as a generalization of undirected graphs and directed graphs, therefore notions issued from both undirected graphs and directed graphs have to be extended. In order to describe vertices and vertex subset relations, notions defined only for directed graphs such as:

- parents,
- children,
- ancestors,
- descendants,
- non-descendants,

and the notion of neighbors or connected vertices only defined for undirected graphs remain the same, as each of these notation closures. As mixed graphs combine lineage and neighborhood relations, it is convenient to work with the notion of boundary, noted $\mathrm{bd}(\cdot)$, defined as the union of parents and neighbors of the vertex,

$$\forall v \in \mathcal{V},\ \mathrm{bd}(v) = \mathrm{pa}(v) \cup \mathrm{ne}(v).$$

As for previous notations the boundary of a subset $\mathcal{A}$ of $\mathcal{V}$, it is the union of the boundaries of each vertex of $\mathcal{A}$ minus all elements belonging to $\mathcal{A}$,

$$\forall \mathcal{A} \subseteq \mathcal{V},\ \mathrm{bd}(\mathcal{A}) = \{\cup_{v \in \mathcal{A}} \mathrm{bd}(v)\} \setminus \mathcal{A},$$

and its closure is noted $\mathrm{Bd}(\cdot)$,

$$\mathrm{Bd}(\mathcal{A}) = \mathrm{bd}(\mathcal{A}) \cup \mathcal{A}.$$

A chain component is a set $\mathcal{A}$ of $\mathcal{V}$ such that $\mathcal{G}_{\mathcal{A}}$ is a connected undirected graph and for all vertices $v \in \mathcal{V} \setminus \mathcal{A}$, $\mathcal{G}_{\mathcal{A} \cup \{i\}}$ is no more a connected undirected graph. One denotes by $\mathcal{H}_{\mathcal{G}}$ the set containing all chain components in a mixed graph.

A u-shape is a set of 4 distinct vertices u, v, w and z of $\mathcal{V}$ such that (see $\mathcal{G}_9$ in figure 1.9):

- w and z are in the same chain component,
- w is a child of u,
- z is a child of v,
- u is not in the boundary of v and conversely.

An immorality in a mixed graph is an u-shape or a v-shape. On denotes by $\mathcal{I}_{\mathcal{G}}$ the set of immoralities in $\mathcal{G}$ defined as follows

$$\mathcal{I}_{\mathcal{G}} = \left\{(u,v,w) \in \mathcal{V}^3 \,\middle|\, [u \notin \mathrm{pa}\,(v)] \wedge [v \notin \mathrm{pa}\,(u)] \wedge [w \in ch\,(u)] \wedge [w \in ch\,(v)]\right\} \cup \left\{(u,v,w,z) \in \mathcal{V}^4 \,\middle|\, \begin{array}{l} [u \notin \mathrm{bd}\,(v)] \wedge [v \notin \mathrm{bd}\,(u)] \\ \wedge [w \in ch\,(u)] \wedge [z \in ch\,(v)] \\ \wedge [w \in cn\,(z)] \wedge [z \in cn\,(w)] \end{array}\right\}.$$

Remarkable graphs As for undirected graphs and directed graphs particular edge sets define important classes of mixed graphs (see figure 1.9).

In particular, the notion of DAG can also be extend to mixed graphs considering that a mixed graph is said to be a Mixed Acyclic Graph (MAG) if it does not contains any directed l-cycles. By opposition mixed cyclic graphs are mixed graphs which are not MAGs.

Moreover, the following two transformations of a mixed graph $\mathcal{G}$ into an undirected graph are generally considered:

- its undirected version noted $\mathcal{G}^u = (\mathcal{V}, \mathcal{E}^u)$, is the undirected graph obtained by dropping directed edge directions,

$$\mathcal{E}^u = \mathcal{E} \cup \mathcal{E}^r.$$

- ant its moral graph noted $\mathcal{G}^m = (\mathcal{V}, \mathcal{E}^m)$, is the undirected graph obtained by adding all edges corresponding to immoralities into its undirected version,

$$\mathcal{E}^m = \mathcal{E}^u \cup \left\{(u,v) \in \mathcal{P}\,(\mathcal{V}) \,\middle|\, \begin{array}{l} \left[\exists w \in \mathcal{V},\, (u,v,w) \in \mathcal{I}_{\mathcal{G}}\right] \\ \vee \left[\exists\,(w,z) \in \mathcal{V}^2,\, (u,v,w,z) \in \mathcal{I}_{\mathcal{G}}\right] \end{array}\right\}.$$

Separation property Let be three disjoint subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C} \subset \mathcal{V}$. The set $\mathcal{C}$ m-separates sets $\mathcal{A}$ and $\mathcal{B}$ if $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{G}^m_{an(\mathcal{A}\cup\mathcal{B}\cup\mathcal{C})}$[4],

$$\mathcal{A} \underset{m}{\overset{G}{\perp}} \mathcal{B}\,|\,\mathcal{C} \Rightarrow \mathcal{A} \overset{\mathcal{G}^m_{an(\mathcal{A}\cup\mathcal{B}\cup\mathcal{C})}}{\perp} \mathcal{B}\,|\,\mathcal{C}.$$

[4]The subgraph operator has the precedence over the moralization

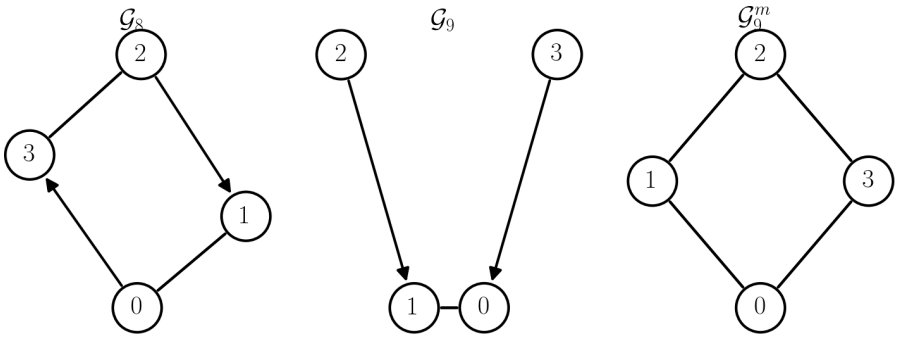


Figure 1.9 – Remarkable mixed graphs. Graph $\mathcal{G}_8$ is a mixed cyclic graph as there is the directed 4-cycle $0, 3, 2, 1, 0$. On graph $\mathcal{G}_9$ which is a MAG there is one u-shape implying the immorality $(2, 3)$. Graph $\mathcal{G}_9^m$, the moral graph of $\mathcal{G}_9$, is not chordal as the immorality $(2, 3)$ in $\mathcal{G}_9$ induced the addition of edge $(2, 3)$ in comparison to its undirected version.

When no confusion could arise the notations $\mathcal{A} \perp \mathcal{B}|\mathcal{C}$ if $\mathcal{C}$ is not the empty set, or $\mathcal{A} \perp \mathcal{B}$ otherwise, will be used.

Mixed graphs generalize both undirected graphs and directed graphs

$$\forall \mathcal{V} \subseteq \mathbb{N}^*, \mathcal{M}(\mathcal{V}) \supset \mathcal{U}(\mathcal{V}) \cup \mathcal{D}(\mathcal{V}),$$

and, as soon as there exists u-shapes in a mixed graph (see figure 1.10), it has no S-equivalent in $\mathcal{U}(\mathcal{V})$ nor $\mathcal{D}(\mathcal{V})$ which illustrates the interest of considering mixed graphs (see figure 1.11). As for the directed case, two mixed graphs $\mathcal{G}$ and $\mathcal{G}'$ sharing the same vertex set but not the same edge set are not necessarily different in terms of m-separations. In fact, similarly to the directed graph case, they are equivalent if and only if they have the same undirected version, the same set of v-shapes and u-shapes,

$$\forall (\mathcal{G}, \mathcal{G}') \in \mathcal{M}^2(\mathcal{V}), \ \mathcal{S}(\mathcal{G}) = \mathcal{S}(\mathcal{G}') \Leftrightarrow \begin{cases} \mathcal{E}^u = \mathcal{E}'^u, \\ \mathcal{I}_{\mathcal{G}} = \mathcal{I}_{\mathcal{G}'}. \end{cases}$$

1.2 Graphical model framework

Graphical models use a graph-based representation as the basis for compactly encoding a complex distribution. In this graph representation the vertices correspond to random variables and edges to direct probabilistic relationships between them.

1.2.1 Random vectors and independencies

Let X be a random variable defined on the probability space $(\Omega, \mathcal{F}, P)$ and x an outcome of X. If the observation space of X, noted $\mathcal{X}$:

- is $\mathbb{N}$ or a subset of $\mathbb{N}$, the random variable X is said to be a discrete random variable,

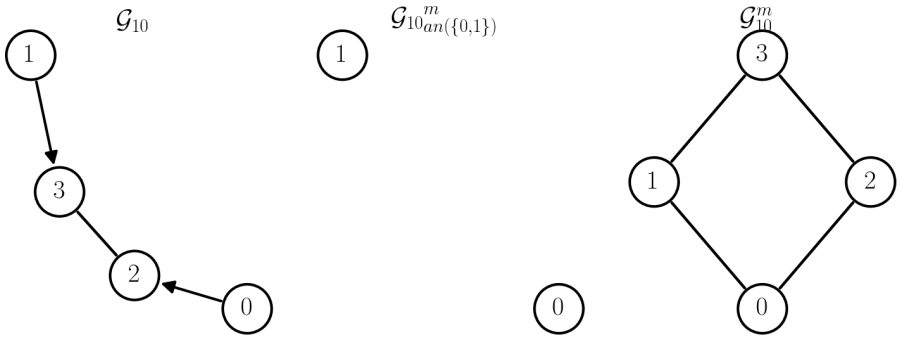


Figure 1.10 – The m-separation property. In the graph $\mathcal{G}_{10}$, $0 \perp 3|1,2$, $1 \perp 3|0,4$ *(see* $\mathcal{G}^m_{10}$*) and* $1 \perp 2$ *(see* $\mathcal{G}_{10^m_{an(\{0,1\})}}$*) hold but not* $1 \perp 2|3,4$ *as the edge* $(1,2)$ *is added in the moral graph* $\mathcal{G}^m_{10}$

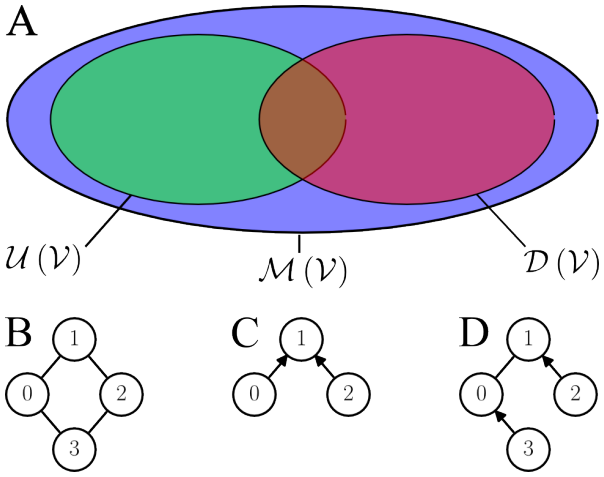


Figure 1.11 – Separation relations among the classes of graphs. (A) A representation of separation spaces for undirected, directed and mixed graphs. (B) A representation of a diamond shape, typical feature of undirected graphs. (C) A representation of a v-shape, typical feature of directed graphs. (D) A representation of an u-shape, typical feature of mixed graphs. The largest separation class is the one of mixed graphs which contains diamond shapes, v-shapes and/or u-shapes. The intersection of undirected graphs and directed graphs spaces is corresponding to chordal undirected graphs and directed graphs without immoralities sub-spaces. As soon as there is a diamond shape in a undirected graph, it cannot be represented by a S-equivalent directed graph. Conversely, as soon as a directed graph has a v-shape, it cannot be represented by a S-equivalent undirected graph. Note that the S-equivalent classes of directed graphs (i.e. directed graphs same undirected version and the same set of v-shapes) and mixed graphs (i.e. mixed graphs same undirected version, the same set of v-shapes and u-shapes) spaces are not represented. If it were the classes would be split in a lot of S-equivalent graphs classes. In particular, in $\mathcal{U}(\mathcal{V}) \cap \mathcal{D}(\mathcal{V})$ *each S-equivalent classes is represented by an chordal undirected graph.*

- is $\mathbb{R}$ or a subset of $\mathbb{R}$ but not one of $\mathbb{N}$, the random variable X is said to be a continuous random variable.

For a collection $(X_v)_{v\in\mathcal{V}}$ of random variables defined on $(\Omega_v, \mathcal{F}_v, P_v)_{v\in\mathcal{V}}$, $\boldsymbol{X} = (X_v)_{v\in\mathcal{V}}$ denotes the random vector defined on the probability space $(\Omega, \mathcal{F}, P)$, $\boldsymbol{x}$ an outcome of $\boldsymbol{X}$ and $\boldsymbol{\mathcal{X}}$ its observation space. For a subset $\mathcal{A}$ of $\mathcal{V}$, $\boldsymbol{X}_\mathcal{A}$ (resp. $\boldsymbol{x}_\mathcal{A}$ or $\boldsymbol{\mathcal{X}}_\mathcal{A}$) denotes the random vector $(X_v)_{v\in\mathcal{A}}$ (resp. an outcome or the observation space of the random vector $(X_v)_{v\in\mathcal{A}}$). In particular Δ and Γ denote the partition of $\mathcal{V}$ such as:

- $\boldsymbol{\mathcal{X}}_\Delta \subseteq \mathbb{N}^{|\Delta|}$, the random vector $\boldsymbol{X}_\Delta$ is said to be a discrete random vector,
- $\boldsymbol{\mathcal{X}}_\Gamma \subseteq \mathbb{R}^{|\Gamma|}$, the random vector $\boldsymbol{X}_\Gamma$ is said to a continuous random vector.

If both subsets are not empty, the random vector $\boldsymbol{X}$ is said to be a heterogeneous random vector.

Note that in the following $\boldsymbol{X}$ is considered to be a discrete random vector for convenience but the extension to continuous or mixed random vectors is straightforward. For three distinct subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ of $\mathcal{V}$, $\boldsymbol{X}_\mathcal{A}$ is independent of $\boldsymbol{X}_\mathcal{B}$ given $\boldsymbol{X}_\mathcal{C}$ under the joint distribution P if and only if

$\forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$,

$$P\left(\boldsymbol{X}_{\mathcal{A}\cup\mathcal{B}} = \boldsymbol{x}_{\mathcal{A}\cup\mathcal{B}} \,|\, \boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C}\right) = P\left(\boldsymbol{X}_\mathcal{A} = \boldsymbol{x}_\mathcal{A} \,|\, \boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C}\right) \cdot P\left(\boldsymbol{X}_\mathcal{B} = \boldsymbol{x}_\mathcal{B} \,|\, \boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C}\right),$$

whenever

$$P\left(\boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C}\right) > 0,$$

this conditional independence relationship noted

$$\boldsymbol{X}_\mathcal{A} \overset{P}{\perp\!\!\!\perp} \boldsymbol{X}_\mathcal{B} \,|\, \boldsymbol{X}_\mathcal{C},$$

will be simplified, when no confusion could arise, by the notations $\mathcal{A} \perp\!\!\!\perp \mathcal{B}|\mathcal{C}$ if $\mathcal{C}$ is not the empty set, or $\mathcal{A} \perp\!\!\!\perp \mathcal{B}$ otherwise. If $\mathcal{C} = \emptyset$, it is said to be a marginal independence.

When considering a fourth disjoint set $\mathcal{D}$ of $\mathcal{V}$, one can see that the same important properties hold in case of conditional independencies and graph separations. This intuitively introduces the reason of developing a formalism in order to encodes conditional independencies in graphs:

- The symmetry property,

$$\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{C} \Leftrightarrow \mathcal{B} \overset{P}{\perp\!\!\!\perp} \mathcal{A} \,|\, \mathcal{C},$$

- The decomposition property,

$$\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \cup \mathcal{C} \,|\, \mathcal{D} \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{D},$$

- The weak union property,

$$\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \cup \mathcal{C} \,|\, \mathcal{D} \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{D} \cup \mathcal{C},$$

- The contraction property,

$$\left[\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,\middle|\, \mathcal{D} \cup \mathcal{C}\right] \wedge \left[\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{C} \,\middle|\, \mathcal{D}\right] \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \cup \mathcal{C} \,|\, \mathcal{D},$$

Note that the intersection property,

$$\left[\mathcal{A} \overset{\mathcal{G}}{\perp} \mathcal{B} \,\middle|\, \mathcal{C}\right] \wedge \left[\mathcal{A} \overset{\mathcal{G}}{\perp} \mathcal{C} \,\middle|\, \mathcal{B}\right] \Rightarrow \mathcal{A} \overset{\mathcal{G}}{\perp} \mathcal{B} \cup \mathcal{C},$$

which is always true in graphs, holds only under some conditions for conditional independencies

$$\left[\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,\middle|\, \mathcal{C}\right] \wedge \left[\mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{C} \,\middle|\, \mathcal{B}\right] \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \cup \mathcal{C},$$

Actually, the intersection property hold for some interesting distributions called positive distributions. But, for instance, when considering that there exists some deterministic relationships among random variables it does not hold anymore. Let us consider the example presented in Lauritzen (1996) such as:

- $\mathcal{V} = \{0, 1, 2\}$,
- $X_0 = X_1 = X_2$,
- $P(X_0 = 0) = P(X_0 = 1) = \frac{1}{2}$.

If we have $0 \perp\!\!\!\perp 2 \,|\, 1$ and $0 \perp\!\!\!\perp 1 \,|\, 2$, we do not have $0 \perp\!\!\!\perp 1, 2$ since

$$P(\boldsymbol{X} = \{0, 0, 0\}) = \frac{1}{2} \neq \frac{1}{2} \cdot \frac{1}{2} = P\left(\boldsymbol{X}_{\{1,2\}} = \{0, 0\}\right) \cdot P(X_0 = 0).$$

In order to discuss the similarities between separations in a graph and conditional independencies in a distribution let $\mathcal{I}(P)$ be the set that contains all the independencies that holds in a distribution,

$$\mathcal{I}(P) = \left\{(\mathcal{A}, \mathcal{B}, \mathcal{C}) \in \mathcal{V}^3 \,\middle|\, \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{C}\right\}.$$

1.2.2 From graphs to distributions[5]

The formalism of graphical models relies on the definition of Markov and factorization properties ensuring that given a graph $\mathcal{G}$ and a distribution P,

$$\mathcal{I}(P) \subseteq \mathcal{S}(\mathcal{G}).$$

The interest of such a formalism is to enable the derivation of a rich set of independence assertions holding in distributions by simply examining graphs or to define relevant factorization of the distribution for graph structured random vectors.

[5]This section is largely based on Lauritzen (1996)

1.2.2.1 The undirected case

The distribution P is said to obey, relatively to the undirected graph $\mathcal{G}$:

- The Pairwise Markov property (PM), if for every pair of distinct vertices (u, v) that do not belong to set of edges, the random variables X_u and X_v are conditionally independent given the random vector $\boldsymbol{X}_{\mathcal{V}\setminus\{u,v\}}$,

$$\forall\,(u,v) \in \left\{(s,t) \in \mathcal{V}^2 \,\middle|\, [s \neq t] \wedge [(s,t) \notin \mathcal{E}]\right\}, \quad u \overset{P}{\perp\!\!\!\perp} v \,|\, \mathcal{V} \setminus \{u,v\}\,. \tag{PM}$$

- The Local Markov property (LM), if for every vertex v the random variable X_v is conditionally independent to the random vector $\boldsymbol{X}_{\mathcal{V}\setminus \mathrm{Ne}(v)}$ given the random vector $\boldsymbol{X}_{\mathrm{ne}(v)}$.

$$\forall v \in \mathcal{V},\; v \overset{P}{\perp\!\!\!\perp} \mathcal{V} \setminus \mathrm{Ne}\,(v) \;|\, \mathrm{ne}\,(v)\,. \tag{LM}$$

- The Global Markov property (GM), if for every triplets of disjoint subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ of vertices, the random vectors $\boldsymbol{X}_{\mathcal{A}}$ and $\boldsymbol{X}_{\mathcal{B}}$ are conditionally independent given the random vector $\boldsymbol{X}_{\mathcal{C}}$ if $\mathcal{C}$ separates them,

$$\forall \mathcal{A} \subseteq \mathcal{V}, \forall \mathcal{B} \subseteq \mathcal{V} \setminus \mathcal{A}, \forall \mathcal{C} \subseteq \mathcal{V} \setminus \{\mathcal{A} \cup \mathcal{B}\}\,, \quad \mathcal{A} \overset{\mathcal{G}}{\perp} \mathcal{B} \,|\, \mathcal{C} \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{C}. \tag{GM}$$

- The Factorization property (F), if there exists non-negative functions $\phi_{\mathcal{C}}\,(\boldsymbol{x}_{\mathcal{C}})$ called clique potentials such that

$$\forall \boldsymbol{x} \in \boldsymbol{X}\,(\boldsymbol{\Omega})\,, \quad P\,(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z} \prod_{\mathcal{C} \in \mathcal{K}_{\mathcal{G}}} \phi_{\mathcal{C}}\,(\boldsymbol{x}_{\mathcal{C}})\,, \tag{F}$$

  where the partition function Z is a renormalization quantity defined by,

$$Z = \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} \prod_{\mathcal{C} \in \mathcal{K}_{\mathcal{G}}} \phi_{\mathcal{C}}\,(\boldsymbol{x}_{\mathcal{C}})\,.$$

For distributions in which the intersection property holds,

$$\text{(F)} \Leftrightarrow \text{(GM)} \Leftrightarrow \text{(LM)} \Leftrightarrow \text{(PM)},$$

but in the general case,

$$\text{(F)} \Rightarrow \text{(GM)} \Rightarrow \text{(LM)} \Rightarrow \text{(PM)}.$$

Let consider the example introduced by Moussouris (1974) and discussed in Lauritzen (1996) in order to illustrate that (GM) may not imply (F). Let $\mathcal{V} = \{0, 1, 2, 3\}$,

$$\boldsymbol{\Omega} = \times_{v \in \mathcal{V}} \{0, 1\}\,,$$

and

$$\forall \boldsymbol{x} \in \left\{ \begin{array}{cccc} (0,0,0,0)\,, & (1,0,0,0)\,, & (1,1,0,0)\,, & (1,1,1,0)\,, \\ (0,0,0,1)\,, & (0,0,1,1)\,, & (0,1,1,1)\,, & (1,1,1,1) \end{array} \right\}, \quad P\left(\boldsymbol{X}=\boldsymbol{x}\right)=\frac{1}{8}. \quad (1.1)$$

The conditional distribution of X_0 given that $\boldsymbol{X}_{\{1,3\}}=(0,1)$ is degenerate,

$$P\left(X_0=0 \,\middle|\, \boldsymbol{X}_{\{1,3\}}=(0,1)\right)=1,$$

and therefore trivially independent of X_2. All other combinations of conditions on $\boldsymbol{X}_{\{1,3\}}$ give in a similar way degenerate distributions for one of the remaining variables and it can also been observed for given variables $\boldsymbol{X}_{\{0,2\}}$. Hence,

$$[0 \perp\!\!\!\perp 2 \,|\, 1,3] \wedge [1 \perp\!\!\!\perp 3 \,|\, 0,2]\,,$$

which is compatible with graph $\mathcal{G}_{12}$ (see figure 1.12) in term of (GM) but not in term of (F) as the probability distribution do not factorize according to it. Let us consider a *reductio ad absurdum*. If P factorizes according to $\mathcal{G}_{12}$:

$$P\left(\boldsymbol{X}=(0,0,0,0)\right)=\phi_{\{0,1\}}\left(0,0\right)\phi_{\{1,2\}}\left(0,0\right)\phi_{\{2,3\}}\left(0,0\right)\phi_{\{3,0\}}\left(0,0\right)=\frac{1}{8},$$

and

$$P\left(\boldsymbol{X}=(0,0,1,0)\right)=\phi_{\{0,1\}}\left(0,0\right)\phi_{\{1,2\}}\left(0,1\right)\phi_{\{2,3\}}\left(1,0\right)\phi_{\{3,0\}}\left(0,0\right)=0,$$

thus

$$\phi_{\{1,2\}}\left(0,1\right)\phi_{\{2,3\}}\left(1,0\right)=0.$$

Using

$$P\left(\boldsymbol{X}=(0,0,1,1)\right)=\phi_{\{0,1\}}\left(0,0\right)\phi_{\{1,2\}}\left(0,1\right)\phi_{\{2,3\}}\left(1,1\right)\phi_{\{3,0\}}\left(1,0\right)=\frac{1}{8},$$

is leading to

$$\phi_{\{2,3\}}\left(1,0\right)=0,$$

which contradicts

$$P\left(\boldsymbol{X}=(1,1,1,0)\right)=\phi_{\{0,1\}}\left(1,1\right)\phi_{\{1,2\}}\left(1,1\right)\phi_{\{2,3\}}\left(1,0\right)\phi_{\{3,0\}}\left(0,1\right)=\frac{1}{8}\neq 0.$$

Hence P does not factorize according to $\mathcal{G}_{12}$.

1.2.2.2 The directed acyclic case

The distribution P is said to obey with respect to the DAG $\mathcal{G}$:

- The Directed Pairwise Markov property (DPM) if for every pair of distinct vertices (u,v) that are not adjacent such that v is a non-descendant of u, the random variables X_u and X_v are conditionally independent given random vector $\boldsymbol{X}_{\mathrm{nd}(u)\setminus\{v\}}$,

$$\forall u \in \mathcal{V}, \forall v \in \mathrm{nd}\left(u\right),\ u \overset{P}{\perp\!\!\!\perp} v \,|\, \mathrm{nd}\left(u\right)\setminus\{v\}\,. \qquad \text{(DPM)}$$

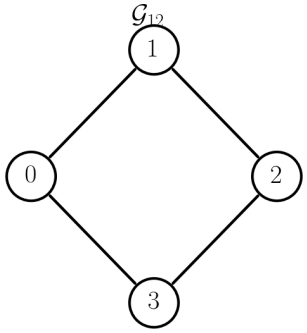


Figure 1.12 – Moussouris (1974) chordless four-cycle graph. Graph $\mathcal{G}_{12}$ represents P (GM) properties defined by (1.1) but P cannot be factorized according to (F). If (F) $\Rightarrow$ (GM) the reciprocal is not always true.

- The Directed Local Markov property (DLM) if for every vertex v, the random variable X_v is independent of the random vector $\boldsymbol{X}_{\mathrm{nd}(v)\setminus\mathrm{pa}(v)}$ given the random vector $\boldsymbol{X}_{\mathrm{pa}(v)}$,

$$\forall v \in \mathcal{V},\ v \overset{P}{\perp\!\!\!\perp} \mathrm{nd}\,(v) \setminus \mathrm{pa}\,(v) \mid \mathrm{pa}\,(v)\,. \tag{DLM}$$

- The Directed Global Markov property (DGM), if for every triplets of disjoint subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ of vertices, the random vectors $\boldsymbol{X}_{\mathcal{A}}$ and $\boldsymbol{X}_{\mathcal{B}}$ are conditionally independent given the random vector $\boldsymbol{X}_{\mathcal{C}}$ if $\mathcal{C}$ d-separates them,

$$\forall \mathcal{A} \subseteq \mathcal{V}, \forall \mathcal{B} \subseteq \mathcal{V} \setminus \mathcal{A}, \forall \mathcal{C} \subseteq \mathcal{V} \setminus \{\mathcal{A} \cup \mathcal{B}\}\,, \quad \mathcal{A} \overset{\mathcal{G}}{\underset{d}{\perp}} \mathcal{B} \,|\, \mathcal{C} \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{C}. \tag{DGM}$$

- The Directed Factorization property (DF) if the distribution can be factorized as follows,

$$\forall \boldsymbol{x} \in \boldsymbol{X}\,(\boldsymbol{\Omega})\,,\ P\,[\boldsymbol{X} = \boldsymbol{x}] = \prod_{v \in \mathcal{V}} P\left(X_v = X_v \,\middle|\, \boldsymbol{X}_{\mathrm{pa}(v)} = \boldsymbol{x}_{\mathrm{pa}(v)}\right). \tag{DF}$$

In the directed acyclic case for any distribution P, almost all these properties are equivalent,

$$\text{(DF)} \Leftrightarrow \text{(DGM)} \Leftrightarrow \text{(DLM)} \Leftarrow \text{(DPM)},$$

and for distributions in which the intersection property holds they are all equivalents,

$$\text{(DF)} \Leftrightarrow \text{(DGM)} \Leftrightarrow \text{(DLM)} \Leftrightarrow \text{(DPM)}.$$

1.2.2.3 The mixed acyclic case

The distribution P is said to obey with respect to the MAG $\mathcal{G}$:

- The Pairwise Chain Markov property (PCM) if for every pair of distinct vertices (u, v) that are not adjacent such that v is a non-descendant of u, the random variables X_u and X_v are conditionally independent given the random vector $\boldsymbol{X}_{\mathrm{nd}(u)\setminus\{v\}}$,

$$\forall u \in \mathcal{V},\, \forall v \in \mathrm{nd}\,(u)\,,\; u \overset{P}{\perp\!\!\!\perp} v \,|\, \mathrm{nd}\,(u) \setminus \{v\}\,. \tag{PCM}$$

- The Local Chain Markov property (LCM) if for every vertex v, the random variable X_v is independent of the random vector $\boldsymbol{X}_{\mathrm{nd}(v)\setminus\mathrm{bd}(v)}$ given the random vector $\boldsymbol{X}_{\mathrm{bd}(v)}$,

$$\forall v \in \mathcal{V},\; v \overset{P}{\perp\!\!\!\perp} \mathrm{nd}\,(v) \setminus \mathrm{bd}\,(v) \,|\, \mathrm{bd}\,(v)\,. \tag{LCM}$$

- The Global Chain Markov property (GCM) if for every triplets of disjoint subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ of vertices, the random vectors $\boldsymbol{X}_{\mathcal{A}}$ and $\boldsymbol{X}_{\mathcal{B}}$ are conditionally independent given the random vector $\boldsymbol{X}_{\mathcal{C}}$ if $\mathcal{C}$ m-separates them,

$$\forall \mathcal{A} \subseteq \mathcal{V}, \forall \mathcal{B} \subseteq \mathcal{V} \setminus \mathcal{A}, \forall \mathcal{C} \subseteq \mathcal{V} \setminus \{\mathcal{A} \cup \mathcal{B}\}\,, \quad \mathcal{A} \overset{\mathcal{G}}{\underset{m}{\perp}} \mathcal{B} \,|\, \mathcal{C} \Rightarrow \mathcal{A} \overset{P}{\perp\!\!\!\perp} \mathcal{B} \,|\, \mathcal{C}. \tag{GCM}$$

- The Factorization Chain property (FC) if the distribution can be factorized as follows,

$$P\,(\boldsymbol{X} = \boldsymbol{x}) = \prod_{\mathcal{C} \in \mathcal{H}_{\mathcal{G}}} P\left(\boldsymbol{X}_{\mathcal{C}} = \boldsymbol{x}_{\mathcal{C}} \,\middle|\, \boldsymbol{X}_{\mathrm{pa}(\mathcal{C})} = \boldsymbol{X}_{pa(\mathcal{C})}\right) \tag{FC}$$

  where for each chain component $\mathcal{C} \in \mathcal{H}_{\mathcal{G}}$, $P\left(\boldsymbol{X}_{\mathcal{C}} = \boldsymbol{x}_{\mathcal{C}} \,\middle|\, \boldsymbol{X}_{\mathrm{pa}(\mathcal{C})} = \boldsymbol{x}_{\mathrm{pa}(\mathcal{C})}\right)$ obeys to (F) in $\mathcal{G}^{m}_{\mathcal{A}\cup\mathrm{pa}(\mathcal{A})}$, the moral subgraph induced by $\mathcal{A} \cup \mathrm{pa}\,(\mathcal{A})$.

In the mixed acyclic case, with analogy with the undirected case, for any distribution P, we have

$$\text{(FC)} \Rightarrow \text{(GCM)} \Rightarrow \text{(LCM)} \Rightarrow \text{(PCM)}$$

in the general case but if the intersection property holds in P,

$$\text{(FC)} \Leftrightarrow \text{(GCM)} \Leftrightarrow \text{(LCM)} \Leftrightarrow \text{(PCM)}.$$

Note

In fact, four classes of Markov properties for chain graphs have been discussed in the literature (see Drton, 2009, and references therein). These four types arise by combining two different interpretations of directed edges with two different interpretations of undirected edges. Two of them are widely used to represent structured random vectors:

- The Lauritzen, Wermuth and Frydenberg property (LWF) or block concentration Markov property for mixed graphs (Lauritzen and Wermuth, 1989; Frydenberg, 1990).
- The Alternate Markov Property (AMP) or concentration regression Markov property for mixed graphs (Andersson et al., 1996).

We focused here on the LWF mixed graphs as they are known to be more easily interpretable in the Gaussian case (Cox and Wermuth, 1993) and to lead to smooth models even for discrete multivariate distribtuions (Drton, 2009).

1.2.3 From distributions to graphs[6]

Given a graph $\mathcal{G}$ and a distribution P, we hereinabove presented under which conditions $\mathcal{G}$ is an Independence map (I-map) for P, that is

$$\mathcal{I}(P) \subseteq \mathcal{S}(\mathcal{G}),$$

or how to define relevant factorizations of P given the graph $\mathcal{G}$. If the latter property is useful for modeling purposes, the derivation of independence assertions holding in distributions by simply examining graphs is of particular interest for interpretation purposes. To this end, the construction of $\mathcal{G}$ from a given P needs to be minimal in some sense: the complete graph is always an I-map but does not enable the derivation of independence assertions holding in P.

A graph $\mathcal{G}$ is a minimal I-map for P if it is an I-map and if the removal of any edge in $\mathcal{G}$ renders it not an I-map. $\mathcal{G}$ is a perfect I-map for P if the set of separations holding in $\mathcal{G}$ is equal to the set of independencies holding in P,

$$\mathcal{I}(P) = \mathcal{S}(\mathcal{G}).$$

1.3 Gaussian graphical models[7]

In the remainder of this chapter we focus on Gaussian graphical models as an illustration of the graphical model framework.

[6] This section is largely based on Koller and Friedman (2009)
[7] This section is largely based on Koller and Friedman (2009) and Lauritzen (1996)

1.3.1 Parametrizations

Gaussian distributions A continuous random variable X follows an univariate Gaussian distribution with mean μ and variance σ^2, denoted by $X \sim \mathcal{N}(\mu, \sigma^2)$, if it has the following density function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}.$$

Similarly, a continuous random vector $\boldsymbol{X} = (X_v)_{v\in\mathcal{V}}$ follows a multivariate Gaussian distribution with vector mean $\boldsymbol{\mu}$ and variance-covariance matrix Σ, denoted by $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, if it has the following density function

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{|\mathcal{V}|}\det(\Sigma)}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\},$$

where $\det(\Sigma)$ denotes the determinant of the variance-covariance matrix, and $\cdot^T$ the transpose of a vector or a matrix. For multivariate Gaussian distributions, independencies are easy to determine directly from the parameters of the distributions:

- marginal independencies can be determined in the variance-covariance matrix

$$\forall (u,v) \in \mathcal{P}(\mathcal{V}), \quad \Sigma_{u,v} = 0 \Leftrightarrow X_u \perp\!\!\!\perp X_v,$$

- conditional independencies can determined in the concentration matrix $\Theta = \Sigma^{-1}$

$$\forall (u,v) \in \mathcal{P}(\mathcal{V}), \quad \Theta_{u,v} = 0 \Leftrightarrow X_u \perp\!\!\!\perp X_v \,|\, \boldsymbol{X}_{\mathcal{V}\setminus\{u,v\}}.$$

Let $\Pi = \{\mathcal{A}, \mathcal{B}\}$ be a partition of $\mathcal{V}$ such as

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_{\mathcal{A}} \\ \boldsymbol{\mu}_{\mathcal{B}} \end{pmatrix},$$

and

$$\Sigma = \begin{pmatrix} \Sigma_{\mathcal{A},\mathcal{A}} & \Sigma_{\mathcal{A},\mathcal{B}} \\ \Sigma_{\mathcal{B},\mathcal{A}} & \Sigma_{\mathcal{B},\mathcal{B}} \end{pmatrix}.$$

The distribution of $\boldsymbol{X}_{\mathcal{A}} \,|\, \boldsymbol{X}_{\mathcal{B}} = \boldsymbol{x}_{\mathcal{B}}$ is a multivariate Gaussian distribution denoted $\mathcal{N}(\boldsymbol{\mu}', \Sigma')$ with

$$\boldsymbol{\mu}' = \boldsymbol{\mu}_{\mathcal{A}} + \Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}(\boldsymbol{x}_{\mathcal{B}} - \boldsymbol{\mu}_{\mathcal{B}}),$$

and

$$\Sigma' = \Sigma_{\mathcal{A},\mathcal{A}} - \Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}.$$

Variables indexed by $\mathcal{A}$ are called response variables, and these indexed by $\mathcal{B}$ are called explanatory variables.

Using block matrix decomposition note that

$$\begin{aligned}
\Theta_{\mathcal{B},\mathcal{A}} &= -\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\left(\Sigma_{\mathcal{A},\mathcal{A}} - \Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\right)^{-1} \\
&= -\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\Sigma'^{-1} \\
&= -\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\Theta',
\end{aligned}$$

with $\Theta' = \Sigma'^{-1}$. Therefore

$$\begin{aligned}\Theta_{\mathcal{A},\mathcal{B}} &= \Theta_{\mathcal{B},\mathcal{A}}^T \\ &= -\left(\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\Sigma'^{-1}\right)^T \\ &= -\Theta'\left(\Sigma_{\mathcal{B},\mathcal{B}}^{-1}\Sigma_{\mathcal{B},\mathcal{A}}\right)^T \\ &= -\Theta'\Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}.\end{aligned}$$

Hence, conditional independence relationships in conditional Gaussian distributions can be derived from null coefficients (Wermuth and Lauritzen, 1990):

- in the matrix $\Theta'\Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}$ considering a response and an explanatory variable given all other response and explanatory variables.
- in the matrix Θ' considering two response variables given all other response and explanatory variables.

The matrix $\Sigma_{\mathcal{A},\mathcal{B}}\Sigma_{\mathcal{B},\mathcal{B}}^{-1}$ is called the regression matrix and its elements regression coefficients.

From Gaussian distributions to graphs Since multivariate Gaussian densities are positive distributions, in the case of:

- undirected graphical models

$$(\text{F}) \Leftrightarrow (\text{GM}) \Leftrightarrow (\text{LM}) \Leftrightarrow (\text{PM}),$$

- directed graphical models

$$(\text{DF}) \Leftrightarrow (\text{DGM}) \Leftrightarrow (\text{DLM}) \Leftrightarrow (\text{DPM}),$$

- mixed graphical models

$$(\text{FC}) \Leftrightarrow (\text{GCM}) \Leftrightarrow (\text{LCM}) \Leftrightarrow (\text{PCM}).$$

As a consequence, with the conditional independence properties stated above, the matrix $\underline{\mathcal{G}}$ where the general element $\underline{\mathcal{G}}_{u,v}$ is obtained by binarization of the concentration matrix,

$$\forall\,(u,v) \in \mathcal{P}\left(\mathcal{V}\right), \quad \begin{cases} \underline{\mathcal{G}}_{u,v} = 1 & \text{if} \quad \Theta_{u,v} \neq 0, \\ \underline{\mathcal{G}}_{u,v} = 0 & \text{otherwise,} \end{cases}$$

is an adjacency matrix of an undirected graph $\mathcal{G}$ corresponding to a minimal I-map of the distribution. Gaussian directed graphical model are defined such that each variable associated to a vertex:

- without parent follows an univariate marginal Gaussian distribution,

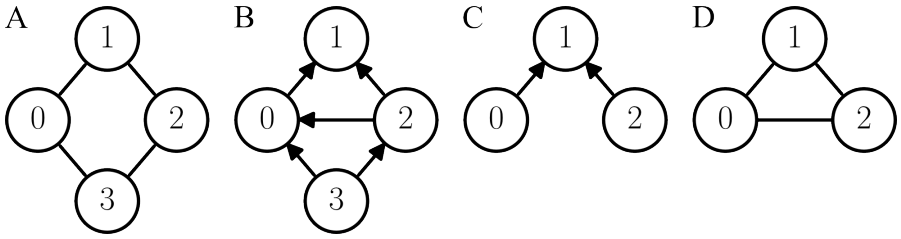


Figure 1.13 – Examples of Gaussian graphical models. (A) An undirected perfect I-map for a multivariate Gaussian distribution. (B) A directed minimal I-map of the multivariate Gaussian distribution defined in (A). (C) A directed perfect I-map for a multivariate Gaussian distribution. (D) An undirected minimal I-map of the multivariate Gaussian distribution defined in (C). In (A) the definition of the multivariate Gaussian distribution require 12 *parameters since there are* 4 *means and* 8 *non null coefficients in the concentration matrix whereas the same distribution in (B) is defined with* 13 *parameters since there are* 4 *means and* 4 *variances and* 5 *regression coefficients. Similarly, the multivariate Gaussian distribution in (C) is defined by* 8 *parameters whereas in (D)* 9 *parameters are used.*

- with parents follows an univariate conditional Gaussian distribution.

In fact, Wermuth (1980) demonstrated that any multivariate Gaussian distribution can also be represented by a Gaussian directed graphical model with any ordering $\sigma \in \mathfrak{S}(\mathcal{V})$ such as

$$\forall v \in \mathcal{V}^*, \quad \mathrm{pa}(\sigma(v)) \subseteq \left\{X_{\sigma(0)}, \ldots, X_{\sigma(v-1)}\right\}.$$

As soon as there are non-zero regression coefficients, the resulting graphs (i.e. for each ordering) are also I-maps. Similar results holds for Gaussian mixed graphs with partial ordering of vertices (Wermuth, 1992). Gaussian mixed graphical models are defined in a similar way as Gaussian directed graphical models. The only differences are that if a chain component has cardinality superior to 1, considered variables follow a multivariate marginal Gaussian distribution if they do not have parents or a multivariate conditional Gaussian distribution otherwise. While the three representations are equivalent in their expressive power, there is not a one-to-one correspondence between their parametrizations (see figure 1.13).

1.3.2 Inference

Parameter inference Given a directed graph, the Maximum Likelihood (ML) parameter inference of Gaussian multivariate distributions reduces to ML parameter inference of univariate marginal and conditional Gaussian distributions, which is standard in statistics.

In the case of undirected graph, the problem of ML parameter inference reduces to a convex optimization problem with the concentration matrix as variable. This problem was first studied by Dempster (1972) under the name of covariance selection. For an undirected graph, there are two cases:

1. The graph is chordal. In this case, the solution of the problem can expressed in closed form (see Wermuth (1980) or Lauritzen (1996) for details).

2. The graph is not chordal. There is no closed form in this case and the ML parameter inference has to be computed iteratively (see Dempster (1972) and Speed and Kiiveri (1986) for common algorithms).

These two cases have also been studied in details by Dahl et al. (2005) who designed clear and efficient algorithms for both cases, especially for non-chordal graph via chordal embedding.

The case of a mixed graph $\mathcal{G}$ can easily be tackled using undirected graph ML inferences in each moral subgraphs $\mathcal{G}^m_{\mathcal{A}\cup \mathrm{pa}(\mathcal{A})}$ induced by chain components $\mathcal{A} \in \mathcal{H}_\mathcal{G}$ in a first step and then a conditioning step with respect to $\mathrm{pa}(\mathcal{A})$, or directly using block recursive equations (Wermuth, 1992).

Structure inference Hereinabove we focused on parameter inference given a graph. Design of such graphs require expert knowledge but in many applications there are simply no experts with sufficient knowledge to be able to design these graphs. In such cases, given a sample to model, the joint inference of structure and parameters enable to infer these graphs, which can be *a posteriori* interpreted by experts.

If the sampling distribution is assumed to be faithful to a Gaussian undirected graphical model, the inference of structure involve finding the pattern of zeros in the concentration matrix. Traditionally, a greedy forward-backward search algorithm was used to determine the zero pattern (Lauritzen, 1996). More recently, another way to estimate the undirected graphical model has been introduced. This approach consist in finding the set of neighbors of each vertex in the graph by regressing that variable against the remaining variables. Meinshausen and Buhlmann (2006) studied this case with the use the Lasso of Tibshirani (1996) and showed that the resulting estimator is consistent, even for high-dimensional graphs. On this basis, exact or faster Lasso based algorithms has been developed to infer the undirected graph (see Banerjee et al. (2008) or the graphical lasso of Friedman et al. (2008) for example) in Gaussian case. Some extensions to discrete multivariate counts data or mixed data have also been proposed (Yang et al., 2012).

If the sampling distribution is assumed to be faithful to a Gaussian directed graphical model, the inference of structure involve finding a directed acyclic graph. There are two main approaches for structure inference in the case of directed acyclic graphs (see Gamez et al., 2011, and references therein):

- The use of greedy search algorithms. Given a consistent scoring function – as the Bayesian Information Criterion (BIC) for instance, see Yang and Chang (2002) for a review of different scores – a search heuristic among the DAG space is used to incrementally improve the considered graphical model. Greedy algorithms were widely studied in literature (Buntine, 1991, 1996; Heckerman et al., 1995; Chickering, 2002; De Campos and Puerta, 2001; Friedman and Goldszmidt, 1997; Friedman et al., 1999). One of the reason may be that they do not depend, in most cases (see Chickering, 2002, for a counter-example), on the distribution parametrization

while producing interpretable models due to their graphical representation. Therefore the directed acyclic graphical model representation was a available for lot of scientific field interested in complex multivariate distributions.

- The use of constraint-based methods involving test of hypothesis (Spirtes et al., 2000; Neapolitan et al., 2004).

The case of a sampling distribution assumed to be faithful to a Gaussian mixed graphical model has been considered less often in the literature. Proposed algorithms (Edwards, 2000; Ma et al., 2008; Drton and Perlman, 2008) mostly focused on test of hypothesis and can require an *a priori* the knowledge of the chain components. Moreover, they are highly related to the Gaussian distribution, another approach could be to propose a greedy search algorithm extension to enable search among the MAG space in order to generalize graph inference algorithms for mixed acyclic graphical models to discrete, categorical and continuous variables.

References

P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905, 1996. URL http://epubs.siam.org/doi/abs/10.1137/S0895479894278952. 20

S. Andersson, D. Madigan, and M. Perlman. An alternative Markov property for chain graphs. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 40–48. Morgan Kaufmann Publishers Inc., 1996. URL http://dl.acm.org/citation.cfm?id=2074289. 35

O. Banerjee, L. El Ghaoui, and A. d Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008. URL http://dl.acm.org/citation.cfm?id=1390696. 39

A. Berry, J. R. Blair, P. Heggernes, and B. W. Peyton. Maximum cardinality search for computing minimal triangulations of graphs. *Algorithmica*, 39(4):287–298, 2004. URL http://link.springer.com/article/10.1007/s00453-004-1084-3. 20

C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973. URL http://dl.acm.org/citation.cfm?id=362367. 19

W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991. URL http://dl.acm.org/citation.cfm?id=2100669. 39

W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(2):195–210, 1996. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=494161. 39

F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1):564–568, 2008. 19

D. Chickering. Learning equivalence classes of Bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002. URL http://dl.acm.org/citation.cfm?id=944800. 39

D. Cox and N. Wermuth. Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218, 1993. URL http://projecteuclid.org/euclid.ss/1177010887. 35

J. Dahl, V. Roychowdhury, and L. Vandenberghe. Maximum likelihood estimation of Gaussian graphical models: numerical implementation and topology selection. *Preprint*, 2005. 39

L. M. De Campos and J. M. Puerta. Stochastic local algorithms for learning belief networks: Searching in the space of the orderings. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 228–239. Springer, 2001. URL http://link.springer.com/chapter/10.1007/3-540-44652-4_21. 39

A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972. URL http://www.jstor.org/stable/10.2307/2528966. 38, 39

M. Drton. Discrete chain graph models. *Bernoulli*, 15(3):736–753, 2009. URL http://projecteuclid.org/euclid.bj/1251463279. 35

M. Drton and M. D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008. URL http://www.sciencedirect.com/science/article/pii/S0378375807002303. 40

P. Eades. A heuristics for graph drawing. *Congressus Numerantium*, 42:146–160, 1984. URL http://ci.nii.ac.jp/naid/10000075358/en/. 11, 12, 13, 14

D. Edwards. *Introduction to graphical modelling*. Springer, 2000. 40

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3):432–441, 2008. URL http://biostatistics.oxfordjournals.org/content/9/3/432.short. 39

N. Friedman and M. Goldszmidt. Sequential update of Bayesian network structure. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 165–174. Morgan Kaufmann Publishers Inc., 1997. URL http://dl.acm.org/citation.cfm?id=2074246. 39

N. Friedman, I. Nachman, and D. Peer. Learning Bayesian network structure from massive datasets: the sparse candidate algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc., 1999. URL http://dl.acm.org/citation.cfm?id=2073820. 39

T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. URL http://onlinelibrary.wiley.com/doi/10.1002/spe.4380211102/abstract. 12, 13, 14

M. Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17:333–353, 1990. URL http://www.jstor.org/stable/4616181. 35

J. A. Gamez, J. L. Mateo, and J. M. Puerta. Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1-2):106–148, 2011. URL http://link.springer.com/article/10.1007/s10618-010-0178-6. 39

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995. URL http://link.springer.com/article/10.1007/BF00994016. 39

J. D. Hunter. Matplotlib: A 2D graphics environment. *IEEE Transactions on Computing in Science & Engineering*, 9(3):0090–95, 2007. URL http://doi.ieeecomputersociety.org/10.1109/mcse.2007.55. 15, 18

S. Kirkpatrick, D. G. Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. URL http://link.springer.com/article/10.1007/BF01009452. 12

S. G. Kobourov. *Spring embedders and force directed graph drawing algorithms*, chapter 12, pages 383–408. Tamassia, 2012. URL http://arxiv.org/abs/1201.3011. 11, 12

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 9, 35

S. Lauritzen. *Graphical models*, volume 17. Oxford University Press, 1996. 9, 19, 30, 31, 35, 39

S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989. URL http://www.jstor.org/stable/2241503. 35

Z. Ma, X. Xie, and Z. Geng. Structural learning of chain graphs via decomposition. *Journal of machine learning research: JMLR*, 9:2847, 2008. 40

N. Meinshausen and P. Buhlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006. URL http://projecteuclid.org/euclid.aos/1152540754. 39

J. Moussouris. Gibbs and Markov random systems with constraints. *Journal of statistical physics*, 10(1):11–33, 1974. URL http://link.springer.com/article/10.1007/BF01011714. x, 31, 33

R. E. Neapolitan et al. *Learning Bayesian networks*, volume 38. Prentice Hall Upper Saddle River, 2004. 40

D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976. URL http://epubs.siam.org/doi/abs/10.1137/0205021. 20

T. Speed and H. Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986. URL http://projecteuclid.org/euclid.aos/1176349846. 39

P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2000. 40

K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995. URL http://www.sciencedirect.com/science/article/pii/S1045926X85710130. 13, 15, 17

R. Tamassia. *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007. ISBN 1584884126. 11, 13

R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972. URL http://epubs.siam.org/doi/abs/10.1137/0201010. 17, 18

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. URL http://www.jstor.org/stable/10.2307/2346178. 39

E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006. 19

N. Wermuth. Linear recursive equations, covariance selection, and path analysis. *Journal of the American Statistical Association*, 75(372):963–972, 1980. doi: 10.1080/01621459.1980.10477580. URL http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1980.10477580. 38, 39

N. Wermuth. On block-recursive linear regression equations. *Reuista Brasileim de Probabilidade Estattstic*, 6:1–56, 1992. 38, 39

N. Wermuth and S. L. Lauritzen. On substantive research hypotheses, conditional independence graphs and graphical chain models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(1):21–50, 1990. URL http://www.jstor.org/stable/2345650. 37

E. Yang, P. Ravikumar, G. Allen, and Z. Liu. Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems 25*, pages 1367–1375, 2012. 39

S. Yang and K.-C. Chang. Comparison of score metrics for Bayesian network learning. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 32(3):419–428, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1046072. 39

M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981. URL http://epubs.siam.org/doi/pdf/10.1137/0602010. 20

Chapter 2

Tree-indexed data and Markov Tree (MT) models

Abstract In this chapter graphs and graphical models introduced in the previous chapter are used in the context of directed tree-graphs.

Firstly directed tree-graphs are used to define tree-indexed data that are the structured data of interest in this thesis. In the context of statistical modeling for tree-indexed data, visualization of these data is critical for hypothesis formulation and model validation. Some drawing algorithms are therefore introduced.

Secondly, examples of tree-indexed data used in the remainder of this thesis are presented in details. Two examples taken from plant developmental biology are considered and associated modeling issues introduced. The first one at a microscopic scale where tree-indexed data are used to represent cell lineage trees. The second one at a macroscopic scale where tree-indexed data are used to represent plant architecture.

Finally, since presented examples raise the issue of motif detection in tree-indexed data, state-of-the-art methods that address such problems are reviewed. In particuler the ability of such models to detect motifs of interest in these applications are discussed. This enables to target the hypotheses that need to be relaxed in the remainder of this thesis.

Keywords cell lineage; Markovian models; plant architecture; quotient tree graph; tree graph drawing; tree-indexed data; tree pattern

Contents

2.1 Introduction to tree-indexed data 47
2.1.1 Definitions 47
2.1.2 Drawing tree-indexed data 48
2.2 Tree-indexed data and plants 51
2.2.1 Tree-indexed data at cellular scale 52
2.2.2 Tree-indexed data at whole plant scale 55
2.3 Markov models for tree indexed-data 59
2.3.1 Markov models 60
2.3.2 Hidden Markov Tree (HMT) models 61
References 62

2.1 Introduction to tree-indexed data

2.1.1 Definitions

Tree-indexed data Data of interest are tree-indexed data $\bar{x} = (x_t)_{t\in\mathcal{T}}$ where $\mathcal{T} \subset \mathbb{N}$ is the set of vertices of a directed tree graph $\mathcal{T} = (\mathcal{T}, \mathcal{E})$, $\mathcal{E} \subset \mathcal{T} \times \mathcal{T} \setminus \mathcal{R}$ the set of directed edges representing lineage relationships between vertices and $\mathcal{R}$ the set of roots.

Note

Sensu stricto $\mathcal{T}$ is a directed tree graph but *sensu lato*, $\mathcal{T}$ is a forest of directed tree graphs.

Let $\mathcal{A}$ be a subset of $\mathcal{T}$ and $\bar{x}_\mathcal{A}$ be the subset of $\bar{x}$ obtained by only considering vertices in $\mathcal{A}$,

$$\forall \mathcal{A} \subseteq \mathcal{T},\ \bar{x}_\mathcal{A} = (x_t)_{t\in\mathcal{A}}\,.$$

Topological notions Since $\mathcal{T}$ is a tree – or a forest – topological notions of directed graphs directly apply to tree-indexed data.

For example child ($\mathrm{ch}\,(.)$), descendant ($\mathrm{de}\,(.)$), ancestor ($\mathrm{an}\,(.)$) sets of a vertex – or set of vertices – and their closures (capitalized notations) allow the characterization of relations between vertices in the data. In particular, for any vertex $t \in \mathcal{T}$, $\bar{x}_{\mathrm{De}(t)}$ is the subset indexed by the subtree $\tau_t = \mathit{sub}\,[\tau, \mathrm{De}\,(t)]$ rooted at vertex t.

It is worth noting that in trees, the parents set ($pa\,(.)$) of a vertex has cardinal 0 or 1. Therefore the notation of parenthood defined for directed graphs is slightly altered when working with trees. The parent of a vertex is only defined for non-root vertices and is another vertex

$$\forall\,(v,u) \in \mathcal{T} \times \mathcal{T} \setminus \mathcal{R}, \quad (v,u) \in \mathcal{E} \Leftrightarrow \mathrm{pa}\,(u) = v.$$

Vertices sharing the same parent are called sibling vertices.

Roots, noted $\mathcal{R}$, and leaves, noted $\mathcal{L}$, play a central role in trees. The length of the directed path from a root to a vertex t is named depth of a vertex and denoted d_t ,

$$\forall t \in \mathcal{T}, \quad d_t = \begin{cases} 0 & \text{if} \quad t \in \mathcal{R}, \\ d_{\mathrm{pa}(t)} + 1 & \text{otherwise.} \end{cases}.$$

Similarly, the length of longest directed path from a vertex to its leaves is named height and denoted h_t,

$$\forall t \in \mathcal{T}, \quad h_t = \begin{cases} 0 & \text{if} \quad t \in \mathcal{L}, \\ \max\limits_{s\in\mathrm{ch}(t)} \{h_s\} + 1 & \text{otherwise.} \end{cases}.$$

Quotient tree graphs Quotient tree graphs are quotient graphs of tree graphs, noted τ_Π, that are tree graphs. As presented in Godin and Caraglio (1998), a sufficient condition to obtain quotient tree graphs (or similarly quotient forest graphs obtained from forest graphs) is that the subgraphs induced by quotients in Π must be *sensu stricto*

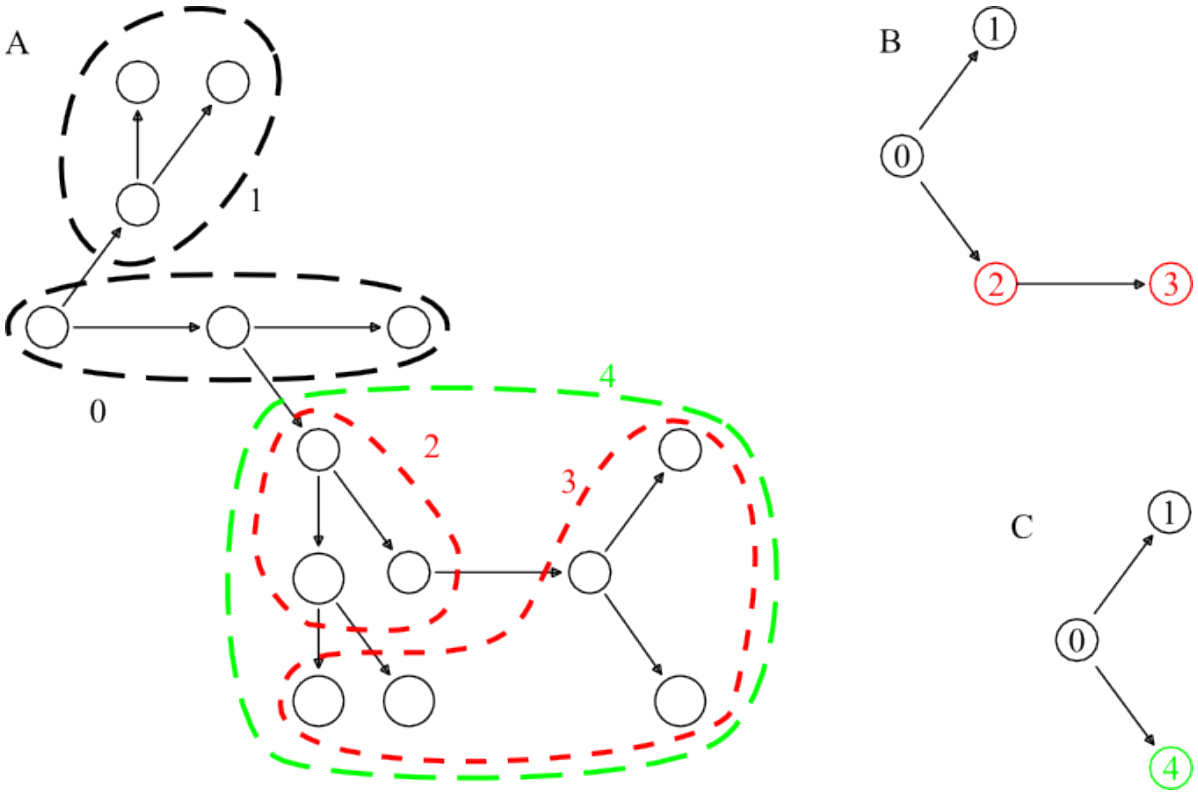


Figure 2.1 – Quotient tree graphs. (A) A tree graph with 5 *quotients represent by the dashed hulls including vertices quotiented. (B) The quotient tree graph obtained by considering the* 2 *black quotients and the* 2 *red ones. (C) The quotient tree graph obtained by considering the* 2 *black quotients and the green one. In (B) and (C) vertices are colored according to the color of quotients in (A). The quotient tree (C) is obtained from a partition where each subgraph induced by quotients are tree graphs. This a sufficient condition but not neccessary as illustrated on (CB where the quotient tree graph has one red quotient which induce a forest tree graph (see Godin and Caraglio, 1998, for more details).*

tree graphs (see figure 2.1). These quotient tree graphs are particularly interesting in the context of tree-indexed data. In fact, this quotienting operations can be seen as the production of structure at coarser scales enabling data inquiries at larger scales.

2.1.2 Drawing tree-indexed data[1]

The objective of tree-indexed data drawing is to automatically produce informative geometric representations for visualization purposes. Since tree-indexed data can be viewed as directed graphs, they could benefits from the drawing conventions and algorithms previously presented (see sub-section 1.1.2 page 10). Nevertheless, it is common to modify the directed graph drawing standards for trees in order to take into account

[1]This section is largely based on Tamassia (2007, chapter 5)

their topological particularities.

Aesthetics of tree drawings Trees and forests are by definition sparse graphs – same number of edges than number of vertices minus the number of roots – therefore node and link diagrams are preferred to adjacency plots. Usually, node and link diagram for drawing trees are compared considering qualitative and quantitative aesthetic criteria:

- the area, defined as the surface of the enclosing rectangle of vertex drawing.
- the ratio, defined as the ratio of the length of the shortest side to the length of the longest side of the enclosing rectangle of vertex drawing.
- the subtree separation property. A drawing of $\mathcal{T}$ satisfies the subtree separation property defined by Chan et al. (1997) if, for any two distinct vertices u and v of $\mathcal{T}$, the enclosing rectangles of the drawing of $\mathcal{T}_u$ and $\mathcal{T}_v$ do not overlap with each other.

Moreover, considering the simplicity of tree topology compared to general directed graphs, two non-exclusive types of drawing are of marked interest:

- Directional drawing. Considering a directed axis in the coordinate system used to draw the trees, no child is placed before its parent. With such a convention and for clarity purposes, one can switch from arrows to lines in order to represent a directed edge, since there is no confusion about the edge direction.
- Planar drawing. A planar drawing is a drawing in which edges do not intersect in the drawing. Planar drawings are normally easier to understand than non-planar drawings (i.e. with edge-crossings). Since any tree admits a planar drawing, it is desirable to obtain planar drawings for trees.

Both Eades (1991) and Fruchterman and Reingold (1991) reported the difficulty of drawing trees without edges crossing by force-directed algorithms. Even when using the magnetic extension of Sugiyama and Misue (1995) if the drawing tend to be directional, the result is not guaranteed to be planar. Hereafter we present only two classes of tree layout algorithms among many others (see Tamassia, 2007, chapter 5 and reference therein). We focused on these two classes since they are relatively easy to understand and to implement while producing high quality layouts.

Level-based layouts The level-based approach to draw trees is characterized by the fact that vertices at the same depth are aligned on the same straight line (i.e. the level) and for two given depths these straight lines are parallel (Bloesch, 1993; Reingold and Tilford, 1981; Buchheim et al., 2002; Walker, 1990). Algorithms based on this approach produce intuitive drawings that exhibit clear display of symmetries and respect both planarity and directionality conventions (see algorithm 2 and its results in figure 2.2).

Algorithm 2 Computing vertex positions of trees for a level drawing

Require: σ a reversed Depth-First Search (DFS) ordering

```
1 function LevelLayout(𝒯)
2     r̄ ← (0, d_t)_{t∈𝒯}                ▷ Use the depth of vertex as second coordinate
3     l ← 0                                ▷ Initialize leaf index
4     for v ∈ 𝒯 do                         ▷ Compute vertex first coordinates
5         if σ(v) ∈ ℒ then
6             r_{σ(v),0} ← l               ▷ Assign leaf index as first coordinate
7             l ← l + 1                    ▷ Increment the leaf index
8         else
9             r_{σ(v),0} ← (Σ_{u∈ch(v)} r_{u,0}) / |ch(v)|   ▷ Assign non-leaf first coordinate
      return r̄
```

Radial layouts While drawing using the level-based layouts respects, in most cases, usual tree drawing conventions and the subtree separation property (it is not the case for the algorithms proposed by Buchheim et al. (2002) and Walker (1990)) quantitative aesthetics criteria are not satisfactory. For instance let $\mathcal{T}$ be a perfect binary tree of depth d. In a perfect binary tree every non-leaf vertex has two children. Therefore there are 2^d leaves. The drawing of $\mathcal{T}$ produced using algorithm 2 has:

- an area of $d \cdot 2^d$ units,
- a ratio of 2^{-d}.

As presented in Tamassia (2007, Chapter 5), by considering a geometric transformation from Cartesian coordinates to polar ones, a level-based layout yields a radial one (see algorithm 3 and its results in figure 2.2).

Algorithm 3 Computing vertex positions of trees for a radial drawing

```
1 function RadialLayout(𝒯)
2     r̄ ← LevelLayout(𝒯)              ▷ Compute vertex r̄ Cartesian coordinates
3     θ ← max_{t∈𝒯} {r_{t,0}} − min_{t∈𝒯} {r_{t,0}} + 1
4     for t ∈ 𝒯 do
5         r_{t,0} ← 2π · r_{t,0}/θ    ▷ Transform vertex Cartesian coordinates into polar ones
      return r̄
```

Compared to the level drawing of $\mathcal{T}$, the radial drawing produced by algorithm 3 has:

- an area of d^2 units,
- a ratio of 1.

Such values are far more satisfactory for these criteria. Given a level-based layout respecting planarity and directionality conventions such as those produced by algorithm 2, the geometric transformation from Cartesian coordinates to polar ones preserves these

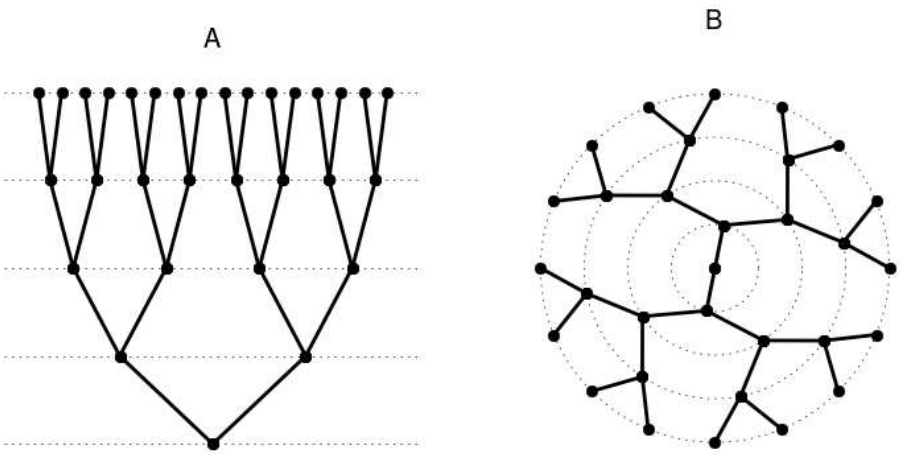


Figure 2.2 – Drawing of trees. (A) The level-based drawing produced by algorithm 2. (B) The radial drawing produced by algorithm 3. Each edge is represented by a straight line as directions are evident as drawings are directional. Each vertex is represented with a black filled dot. Both drawings are directional since no child vertex is before its parent considering the second unit vector. Roots of trees are situated at the bottom using algorithm 2 whereas there are situated in the inner-most concentric circle using algorithm 3. In this thesis tree drawings are achieved using algorithms 2 or 3 for node positioning and a customized interface to the matplotlib package for the rest (Barrett et al., 2005; Hunter, 2007).

conventions. Moreover, although algorithm 3 drawings do not respect *sensu stricto* the subtree separation property, if one changes the enclosing rectangle by an enclosing triangle, the subtree separation property is respected.

2.2 Tree-indexed data and plants

Tree-indexed data are particularly used in signal processing (Crouse et al., 1998; Dasgupta et al., 2001) or 2D and 3D images (Choi and Baraniuk, 2001) as multi-scale representations of path or grid-indexed data. We hereafter do not present a large spectrum of tree-indexed data use but we focus on those which can be collected in the case of plant development. In particular, we present the usefulness of tree-indexed data representation at two scales:

- at a microscopic one, where tree-indexed data represents cell lineages in tissues all along their development (Olariu et al., 2009),
- at a macroscopic one, where tree-indexed data represents the whole plant architecture (Durand et al., 2005).

2.2.1 Tree-indexed data at cellular scale

The study of morphogenesis A major challenge in developmental biology is to understand how multi-cellular tissues can give rise to complex shapes in animals or plants. It is therefore crucial to be able to quantify and explain the cellular and tissular patterns taking place during morphogenesis[2]. Although several studies have provided profound insight into the molecular regulatory networks that act during development, the effects of such networks on shape transformations are often only described qualitatively. Describing size and shape changes as a geometrical output of gene activity requires the quantification of growth patterns at a cellular resolution. Obtaining accurate geometric information about cell position and shape is essential to develop quantitative models of morphogenesis. It is also fundamental to identify groups of cells, not only based on their differentiation state, but on the outcome of the mechanical, genetic and hormonal events that drive morphogenesis.

Meristems and tree representation of tissues[3] In plants, meristems are driving morphogenesis phenomenon. A meristem is a set of embryonic cells that organizes the construction of the plant. It creates new tissues by successive divisions of its stem cells. This division process is coupled with:

- a phenomenon maintaining certain cells obtained by division into a totipotency[4] state,
- a phenomenon enrolling certain cells obtained by division into a differentiation genetic program.

Divisions occur in such a way that cell entering the differentiating process will be part of new tissues and organs while the meristem does not disappear, since there is a constant regeneration of the totipotent cells composing it. Considering tissues and organs produced and location within plant three main types of meristems can be considered:

- The Shoot Apical Meristem (SAM) located at the apex of stems (see figure 2.3). It is responsible for the genesis of the aerial part of the plant, that is leaves, stems and inflorescences[5]. The inflorescence set up is the result of the transformation of a Shoot Apical Meristem (SAM) into a floral meristem.
- The Root Apical Meristem (RAM) located at the tip of the root and responsible for the genesis of the below ground part of the plant, roots.
- The secondary meristems, which are responsible – when located inside stems – for the thickening of the stems, or of roots – when located inside roots.

[2]The morphogenesis is the process of organisms' shapes acquisition

[3]This section is largely based on Campbell and Reece (1984) and Nultsch (1998)

[4]Totipotency is a cellular property reflecting the cell capacity to differentiate into any specialized cell

[5]An inflorescence is a group of flowers on a stem or complicated arrangement of branches issued from a stem.

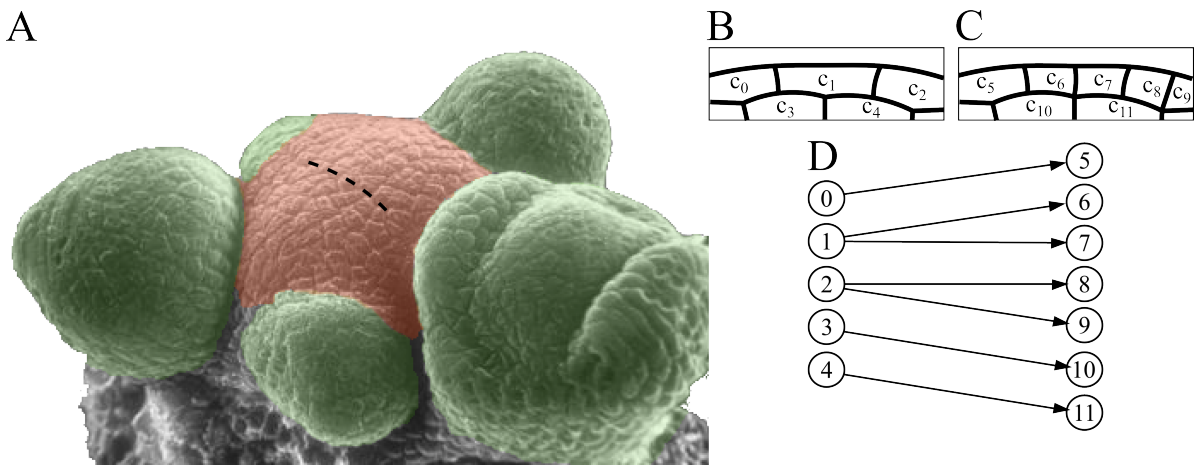


Figure 2.3 – Mersitem and example of tree-indexed data at microscopic scale (Fernandez, 2010; Legrand, 2014). (A) SAM of Arabidopsis thaliana *photography and associated floral meristems. At some point, cells not maintained as embryonic cells in the SAM enter into the differentiation process of floral meristems. (B, C) Schematic cross-sections of a same meristem at different times are represented with identified cells* $(c_t)_{t\in[\![0,11]\!]}$*. The possible location of this cross-section is located on (A) with a dotted black line. Only the first two layers of cells L1 and L2 are represented in this scheme. (C) Tree representation of the follow up from (A) to (B) where each cell is represented by the corresponding labeled vertex and lineages are represented by directed edges. Cells of the first observation (A) are considered as roots of the trees and at each time, a cell is connected to itself at the precedent time if no division occurred and to its mother otherwise. Only two divisions occurred here, from cell* 1 *to cells* 6 *and* 7 *and from cell* 2 *to cells* 8 *and* 9*.*

The idea of meristem development representation into a tree-structure, is to follow up in time meristem cells represented by vertices and use edges to represent lineage relationships (see figure 2.3).

Tree-indexed data collection by 3D + t **meristem imaging** Fernandez et al. (2010) presented a method to generate 3D digitized tissues at cell resolution with automatic tracking of cell lineage during growth. To create a digitized tissue that can be used to quantitatively analyze growth in four dimensions, they developed an experimental pipeline comprising two key steps:

- Multi-angle Acquisition, 3 dimensional Reconstruction and Segmentation (MARS). The multi-angle acquisition produces stacks of 2D images that are transformed into 3D images. In each voxel of the images the intensity of the signal associated to cell walls – or other sources, for instance genetic markers – is stored. At the end of the MARS step, the segmentation associates each voxel to a given cell of the meristem or to the background (see figure 2.4).

- Automated Lineage Tracking (ALT). Once the cells have been identified in the MARS step, the goal of the ALT step is to perform cell tracking all along the

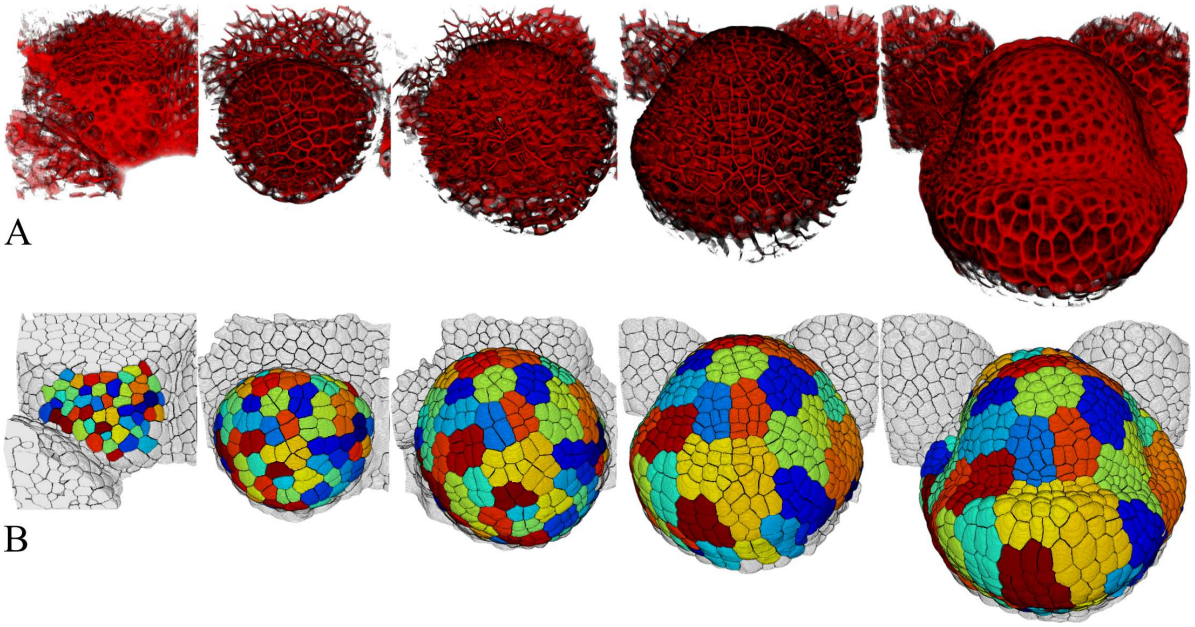


Figure 2.4 – Example of a 3D + t *reconstruction and segmentation. (A)* 3D *images reconstructed after the multi-angle acquisition are displayed according to time. Images, from left to right, were respectively taken at* 0h, 26h, 44h, 56h *and* 69h *after the beginning of the experiment according to Das et al. (2009) experimental design. The red color is due to a marker of cell walls. (B) The corresponding segmented images are presented. Cells are identified and their color is given according to their root identifier in lineage trees (see Legrand, 2014, for more details). Since cells in the same lineage tree share the same color, it can be seen that even though there is no explicit spatial information in trees, since cells cannot migrate in plants, there is an implicit spatial information.*

experiment. At the end of the ALT step lineage trees of cells are obtained.

Available data In this thesis we focused on the joint work concerning flower morphogenesis in *Arabidopsis thaliana* done with J. Legrand, another Ph.D. student of the team (Legrand, 2014). We were interested in SAMs of *Arabidopsis thaliana* transformed into floral meristems. In contrast to the original SAM, a floral meristem follows a determinate growth process[6]. This transformation is controlled by the expression of particular genes, called identity genes, specifying floral organs and causing determinate growth. This work is focused on the L1 cell layer[7] (see figure 2.4) and the lineage trees was produced by Fernandez et al. (2010) MARS-ALT method (see figure 2.3).

Under the assumption that the differentiation process of cells in floral meristems can be assimilated to succession of finite unobservable cell identities, we aim at recovering

[6]The determinate growth process is induced by the termination of stem cells production in the mersitem

[7]The L1 cell layer is the one at the surface of the meristem

these identities on the basis of genetic and geometrical cell characteristics (Legrand, 2014) such as:

- volume,
- surfaces (internal L1/L2 and external L1),
- inertia values (according to three axes),
- principal and secondary curvatures,
- AHP6[8] concentration.

Moreover, in order to understand early mechanisms at stake during flower morphogenesis we aimed at identifying and characterizing cell identity motifs.

2.2.2 Tree-indexed data at whole plant scale[9]

Plant architecture analysis The importance of the topological structure for understanding and analyzing the development of plants was underlined by Hallé et al. (1978) and Gatsuk et al. (1980) who introduced the analysis of plant architecture. Architectural analysis was at first essentially developed as a qualitative method for describing plants (Barthélémy et al., 1989). Afterward, a large research effort was devoted on the one hand to validating and refining architectural concepts and, on the other hand, to studying their application to agronomic contexts. These two preoccupations have led researchers to study progressively how to quantify plant architecture and to develop corresponding concepts and tools (see Godin and Caraglio, 1998, and references therein). The quantitative approach was rapidly faced with the problem of obtaining computational representations of plants that are consistent with field observations. This problem raises the question of measuring plant topological structures and formally representing them.

SAM activity, plant modularity and tree representation of plants The notion of plant topological structure is based on the idea of decomposing a plant into elementary constituents and describing their connections. To obtain natural decompositions, it is possible to take advantage of the fact that the outcome of the plant growth process is modular: a stem is a succession of metamers constituted by an internode, the upper node, leaves and axillary buds attached to the node (see figure 2.5)

The topological structure stemming from a modular organism such as plants consists of a description of the connections between its elementary constituents. Considering only one SAM activity leads to consider stems that can be viewed as sequences: a metamer is connected to an anterior metamer – called the predecessor metamer – and possibly to a posterior metamer – called the successor metamer. But as a SAM produces buds containing other SAMs as soon as an axillary meristem of a stem develops into a lateral

[8]The AHP6 is a marker for a hormonal signal (cytokinins) present in floral meristem

[9]This section is largely based on Godin and Caraglio (1998); Barthélémy and Caraglio (2007)

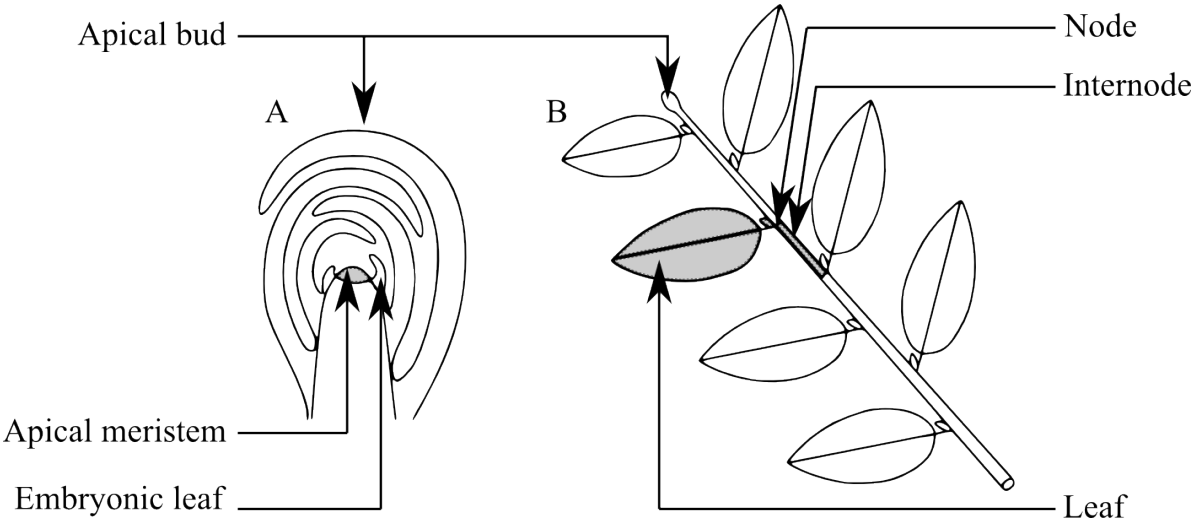


Figure 2.5 – (A) Shoot apical meristem and (B) stem organization (Barthélémy and Caraglio, 2007). Each leafy axis (B) ends in an apical meristem frequently protected by an apical bud (A). Each stem comprises a succession of metamers (in gray on (B) constituted by an internode, the upper node, leaves and axillary buds attached to the node).

axis, a metamer may have more than one child counting the successor and the lateral(s) metamer(s) but only has one predecessor. The whole plant can thus be viewed as a tree-like structure (see figure 2.6).

Tree-indexed data collection by retrospective measurements Plant growth is often a cyclic phenomenon: the setting up of metamers may be interrupted by resting phases corresponding for instance to winter for temperate species. It is thus interesting to consider the meristematic activity at different scales according to the growth strategy of the plant when collecting the plant topological structure. Indeed, if the metamer is the basic unit of the architecture of the plant, according to the plant growth cycles the tree-indexation of data can be considered at different scales (see figure 2.7):

- At the Growth Unit (GU) scale. The GU is composed of the metamers established in a ininterrupted phase of growth.
- At the annual shoot scale. The annual shoot corresponds to the GUs established over a year.
- At the axis scale. The axis corresponds to the succession of annual shoots or GUs produced by the same meristem.

When studying the architecture of a plant, the selection of the botanical entity – elementary constituent at the considered scale – is therefore primordial in order to describe the plant growth strategy (see figure 2.6):

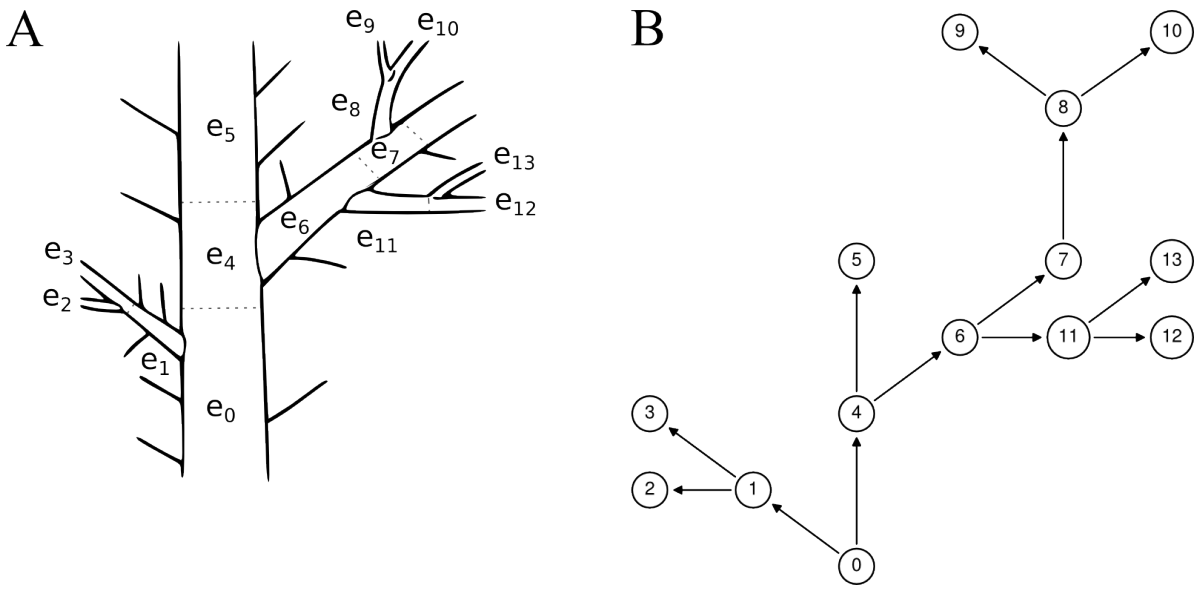


Figure 2.6 – Tree-indexed data representation of plants (Durand et al., 2005). (A) The plant is represented at the Growth Unit (GU) scale where each GU is denoted by e_v with $v \in [\![0, 14[\![$. (B) The formal tree graph representation of the same plant is drawn: each GU e_v is represented by a vertex v. Part of the topological information is not encoded in the graph but can be stored as a property (the three shoots borne by e_1). Few other vertices properties can be defined such the lengths of GUs, their top and bottom diameters... depending on the conducted experiment.

- For the common walnut (Sabatier et al., 1998), we can find two types of annual shoots. Monocyclic annual shoots are preformed in the winter bud. The annual shoot and GU have thus the same meaning. Bicyclic annual shoots are for their part constituted of two GUs. Considering the objectives of the analysis the botanical entity chosen could be the GUs or the annual shoots.
- For some tropical plants, growth can be almost continuous. As a consequence GU are therefore no more relevant. A reasonable choice could be therefore to consider the metamers or the axis as the botanical entity.

Moreover, it is worth noting that if the axis scales are defined for all plants, the GU and the annual shoot scales are mainly defined for temperate species.

Morphological markers, which reflect the past meristem activity, enable the botanist to reconstruct the life of a plant by identifying *a posteriori* growth periods. The tree graph $\mathcal{T}$ is therefore constructed with respect to the plant growth strategy (see figure 2.6). During the same time the univariate set $\bar{x}$ or more generally the multivariate set $\bar{\boldsymbol{x}}$ is collected considering characteristics – depending on the experiment – of botanical entities such as length, diameter, number (or presence) of flowers, number (or presence) of fruits.

We here only consider the collection of a tree-indexed data. Nevertheless since on the same plant there are more than one pertinent scale, data is actually collected using

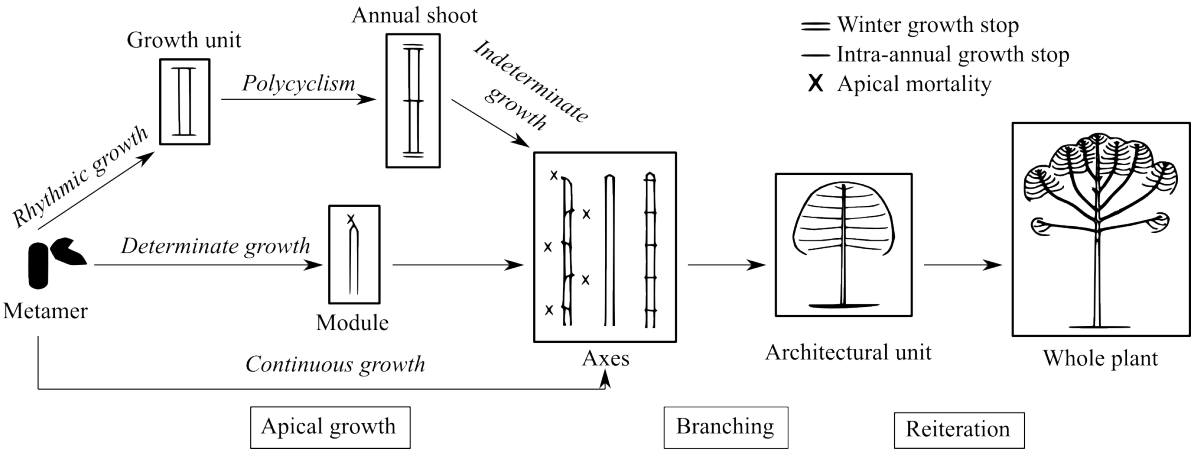


Figure 2.7 – Plant modularity (Barthélémy and Caraglio, 2007). This diagram represents main scales of organization (botanical entity) and repetition phenomena (terms in italics or in boxes) in seed plants.

the Multiscale Tree Graph (MTG) data structure defined by Godin and Caraglio (1998). This MTG data structure can be seen as tree-indexed data where scales are represented by a recursive quotienting of the tree at finer scale (see Godin and Caraglio (1998) for more details). The choice of the scale is therefore made *a posteriori* in order to produce a tree-indexed data and can depend on the studied plant growth aspect.

Available data In this thesis we focused on the joint work concerning mango tree phenology done with Annaëlle Dambreville, Pierre-Eric Lauri et Frédéric Normand. Using mango MTGs containing 15 trees belonging to 5 cultivars collected during the thesis of Dambreville (2012), we aimed at highlighting and characterizing the mango trees patchiness phenomenon. As other tropical trees, mango tree is characterized by strong phenological asynchronisms, between and within trees, entailing patchiness (Chacko, 1986, see figure 2.8). Patchiness is characterized by clumps of either vegetative or reproductive GUs within the canopy: while some parts of the tree canopy develop vegetative GUs, other parts may remain in rest or produce inflorescences at the same time. These asynchronisms concern more or less large branching systems (Ramírez and Davenport, 2010). They entail various agronomic problems, such as the repeated use of pesticides to protect recurrent susceptible phenological stages from pests, or a too extended period of fruit maturity, which may lead to difficulties to organize fruit harvesting.

Previous studies by Dambreville et al. (2013) showed that the fate and burst date of a daughter GU are strongly affected by these of ancestor GUs, indicating that patchiness pattern formation could be studied using spatio-temporal analysis. Our objective here was unfold as follows:

- Characterizing tree patchiness. As stated above, patchiness corresponds to more or less large branching systems sharing almost similar GUs fates. We therefore aimed at recovering a quotient tree of tree-indexed data at GU scale in which quotients were roughly homogeneous in terms of GU fates.



Figure 2.8 – Illustration of mango tree patchiness (Dambreville, 2012). This mango tree is separated into two parts. The left one in dark green is a clump of old GUs wherein fruits can be found. In contrary the right one in light green is a clump of new vegetative GUs.

- Identifying the mechanisms responsible for the set-up of tree patchiness. An inquiry of fate alternations along paths within the tree or successions of homogeneous zones in mango trees could reveal mechanisms at stake in this set-up. To this end, we therefore aimed at highlighting particular fate motifs in mango trees at GU scale.

2.3 Markov models for tree indexed-data

We here assume that the indexed set, $\bar{x} = (x_t)_{t\in\mathcal{T}}$, or more generally $\bar{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t\in\mathcal{T}}$, are the outcomes of a random process.

Note

We here consider that τ is *sensu stricto* a tree. The only root of the tree is noted r. In a forest, trees are considered as independent and indentically distributed.

2.3.1 Markov models

Let us first consider the simple case where $\bar{x}$ is the realization of a $\mathcal{X}$-valued stochastic process $\bar{X} = (X_t)_{t\in\mathcal{T}}$ such that $\mathcal{X} \subset \mathbb{N}$ is called the state space. We are here interested in modeling the distribution of the random process

$$P\left(\bar{X} = \bar{x}\right). \tag{2.1}$$

When considering the case of tree-indexed data, the simplest dependent model that can be constructed is the one that directly consider the tree-graph of the data as a graphical model combined with an usual homogeneity assumption. Combining both hypotheses leads to the following factorization of (2.1):

$$P\left(\bar{X} = \bar{x}\right) = P\left(X_r = x_r\right) \prod_{t\in\mathcal{T}\setminus\{r\}} P\left(X_t = x_t \,\middle|\, X_{\mathrm{pa}(t)} = x_{\mathrm{pa}(t)}\right). \tag{2.2}$$

Given factorization (2.2), classical Markovian models for path-indexed data have been easily adapted to tree-indexed data. These models are called Independent Markov Out-Tree (IMOT) where independent stands for the fact that for such models siblings are assumed independent given their parent. Considering the mango tree application we aimed at highlighting GU fate motifs assuming that at some point, there is a switch from a homogeneous tree to a heterogeneous patchy tree. In order to detect such patterns, we assumed that for a given parent fate:

- and a given growth period, only few different state combinations could be observed for children,
- and for a generation, all children states could be observed.

Under these assumptions we wanted to model dependencies among children fates in order to obtain such inclusion/exclusion patterns. Since in (2.2) children fates are assumed independent given their parent fate, we had to consider other models (see Durand et al., 2005, for a discussion of available models):

- Markov In-Tree (MIT) models. Instead of modeling siblings given their parent as in IMOT, the parent is modeled given its children, introducing the following factorization of (2.1),

$$P\left(\bar{X} = \bar{x}\right) = \prod_{l\in\mathcal{L}} P\left(X_l = x_l\right) \prod_{t\in\mathcal{T}\setminus\mathcal{L}} P\left(X_t = x_t \,\middle|\, \boldsymbol{X}_{\mathrm{ch}(t)} = \boldsymbol{x}_{\mathrm{ch}(t)}\right), \tag{2.3}$$

  where siblings are marginally independent but conditionally dependent to their parent.

- Multi-Type Branching Process (MTBP). Under a permutation invariance property (see Haccou et al., 2005; Kimmel and Axelrod, 2002, for more details), an extension of Markov Out-Tree (MOT) models considering dependencies between children and where the tree topology is partially represented trought the parametrization

of vertex out-degree combinatorics. The following factorization of (2.1) is therefore introduced

$$P\left(\bar{X}=\bar{x}\right) \propto P\left(X_r = x_r\right) \prod_{t\in\mathcal{T}} P\left(\boldsymbol{N}_t = \boldsymbol{n}_t \,|\, X_t = x_t\right), \tag{2.4}$$

where $\boldsymbol{N}_t$ is the discrete random vector of the number of children of vertex t in each state.

In the context of mango tree analysis, the assumption of unordered children and the combinatorics induced by the variable and high number of child vertices in each state induces an inflation in the number of the model parameters. We therefore focused on parametric versions of these models. Since parametric MIT models are not suitable for left-right cases (see chapter 5.4), we thus focused on MTBP models. The issue of specifying parametric MTBPs reduces to the problem of defining parametric models for discrete multivariate counts. The classical discrete multivariate distributions catalog (Johnson et al., 1997) only propose rigid dependence and covariance patterns, thus the next step towards modeling mango tree patchiness was to derive flexible discrete multivariate distributions with complex dependency patterns. This has been dealt with the introduction of mixed graphical models for multivariate discrete random vectors.

2.3.2 Hidden Markov Tree (HMT) models

When confronted to tree-indexed data that do not contain few discrete outcomes as the mango tree case but multidimensional heterogeneous outcomes as the floral meristem case, the MIT and MTBP models cannot be considered as they stand. A widespread extension of Markov Tree (MT) models in such cases is to consider Hidden Markov Tree (HMT) models. HMT models introduced by Crouse et al. (1998) are for MT models what Hidden Markov Chain (HMC) models are to Markov Chain (MC) models. As for HMC models (see Ephraim and Merhav, 2002, for more details), HMT models are no more restricted to categorical variables but deal with any types of random variables or vectors at low cost in term of parameters.

A HMT model can be viewed as a pair of stochastic processes $(S_t, \boldsymbol{X}_t)_{t\in\mathcal{T}}$ where $\bar{S} = (S_t)_{t\in\mathcal{T}}$ is a $\mathcal{S}$-valued MT process called state process and the output or observed process $\bar{\boldsymbol{X}}$ is related to $\bar{S}$ by a probabilistic mapping. Thus, for HMT models, the distribution (2.1) is rewritten as follows

$$\begin{aligned} P\left(\bar{\boldsymbol{X}}=\bar{\boldsymbol{x}}\right) &= \sum_{\bar{s}\in\mathcal{S}^{|\mathcal{T}|}} P\left(\bar{S}=\bar{s}, \bar{\boldsymbol{X}}=\bar{\boldsymbol{x}}\right) \\ &= \sum_{\bar{s}\in\mathcal{S}^{|\mathcal{T}|}} P\left(\bar{S}=\bar{s}\right) \prod_{t\in\mathcal{T}} f_{s_t}\left(\boldsymbol{x}_t\right), \end{aligned} \tag{2.5}$$

where $f_{s_t}(.)$ denotes the density function of the multivariate random vector $\boldsymbol{X}_t$ given the vertex state $S_t = s_t$. Parametrization of HMT models are therefore only depending on that of the state process and of observation densities. The assumption of the output process at vertex t depending only on the underlying state process at vertex t is relevant for the floral meristem application since these states can be interpreted as cell identities

(see Olariu et al., 2009, for an example with human cells). The use of HMT models relies on two main algorithms:

- The smoothing algorithm. Quantities computed during the smoothing algorithm enable an efficient implementation of the E-step of the Expectation-Maximization (EM) algorithm for model parameters inference. Moreover by computing the probabilities of being in each states for each vertex given all the observed data, it constitutes a relevant diagnosis tool (see Durand et al. (2004) in the context of binary trees).

- The dynamic programming restoration algorithm. The goal of this algorithm is to reveal the most probable state tree given all observed data. In our case of floral meristem study, since hidden states are assumed to be corresponding to cell identities, this algorithm provides a direct interpretation of the data.

Because of application contexts, literature on HMT models focused on models defined by (2.5) where the vertex out-degree combinatorics is not represented in the parametrization, in particular:

- Crouse et al. (1998) and Durand et al. (2004) developed efficient EM algorithms and restoration algorithms for Independent Hidden Markov Out-Tree (HIMOT) models where state processes were modeled by IMOT.

- Bacciu et al. (2010) developed the EM algorithm and restoration algorithm for parametric Hidden Markov In-Tree (HMIT) models.

Nevertheless, it was expected that there were high differences between cell identity division patterns and that this phenomena could lead to better discrimination and interpretation of cell identities. The next step towards modeling cell lineage trees was therefore to derive EM and restoration algorithms where the state process was modeled by a MTBP.

References

D. Bacciu, A. Micheli, and A. Sperduti. Bottom-up generative modeling of tree-structured data. In K. Wong, B. Mendis, and A. Bouzerdoum, editors, *Neural Information Processing. Theory and Algorithms*, volume 6443 of *Lecture Notes in Computer Science*, pages 660–668. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17536-7. doi: 10.1007/978-3-642-17537-4_80. URL http://dx.doi.org/10.1007/978-3-642-17537-4_80. 62

P. Barrett, J. Hunter, J. T. Miller, J.-C. Hsu, and P. Greenfield. Matplotlib–a portable python plotting package. In *Astronomical Data Analysis Software and Systems XIV*, volume 347, page 91. P. Shopbell and M. Britton and R. Ebert, 2005. 51

D. Barthélémy and Y. Caraglio. Plant architecture: A dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. *Annals of Botany*, 99(3):

375–407, 2007. doi: 10.1093/aob/mcl260. URL http://aob.oxfordjournals.org/content/99/3/375.abstract. 55, 56, 58

D. Barthélémy, C. Edelin, and F. Hallé. Architectural concepts for tropical trees. In *Tropical forests: botanical dynamics, speciation and diversity*, pages 89–100. Academic Press London, 1989. 55

A. Bloesch. Aesthetic layout of generalized trees. *Software: Practice and Experience*, 23(8):817–827, 1993. URL http://onlinelibrary.wiley.com/doi/10.1002/spe.4380230802/abstract. 49

C. Buchheim, M. Jünger, and S. Leipert. Improving Walker's algorithm to run in linear time. In *Graph Drawing*, pages 344–353. Springer, 2002. URL http://link.springer.com/chapter/10.1007/3-540-36151-0_32. 49, 50

N. A. Campbell and J. B. Reece. *Biology, (2008)*, volume 98. Benjamin Cumming's Publishing Company, 1984. 52

E. Chacko. Physiology of vegetative and reproductive growth in mango (*Mangifera indica* L.) trees. In *Proceedings of the First Australian Mango Research Workshop*, volume 1, pages 54–70. CSIRO Australia, Melbourne, 1986. 58

T. Chan, S. R. Kosaraju, M. T. Goodrich, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In *Graph Drawing*, pages 63–75. Springer, 1997. URL http://link.springer.com/chapter/10.1007/3-540-62495-3_38. 49

H. Choi and R. G. Baraniuk. Multiscale image segmentation using wavelet-domain hidden Markov models. *IEEE Transactions on Image Processing*, 10(9):1309–1321, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=941855. 51

M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668544. 51, 61, 62

A. Dambreville. *Croissance et développement du manguier (*Mangifera indica *L.) in natura: approche expérimentale et modélisation de l'influence d'un facteur exogène, la température, et de facteurs endogènes architecturaux.* PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2012. URL http://hal.archives-ouvertes.fr/tel-00860484/. 58, 59

A. Dambreville, P.-É. Lauri, C. Trottier, Y. Guédon, and F. Normand. Deciphering structural and temporal interplays during the architectural development of mango trees. *Journal of experimental botany*, 64(8):2467–2480, 2013. URL http://jxb.oxfordjournals.org/content/64/8/2467.short. 58

P. Das, T. Ito, F. Wellmer, T. Vernoux, A. Dedieu, J. Traas, and E. M. Meyerowitz. Floral stem cell termination involves the direct regulation of agamous by perianthia. *Development*, 136(10):1605–1611, 2009. URL http://dev.biologists.org/content/136/10/1605.short. 54

N. Dasgupta, P. Runkle, L. Couchman, and L. Carin. Dual hidden Markov model for characterizing wavelet coefficients from multi-aspect scattering data. *Signal Processing*, 81(6):1303–1316, 2001. URL http://www.sciencedirect.com/science/article/pii/S0165168400002620. 51

J.-B. Durand, P. Goncalvès, and Y. Guédon. Computational methods for hidden Markov tree models–An application to wavelet trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1323262. 62

J.-B. Durand, Y. Guédon, Y. Caraglio, and E. Costes. Analysis of the plant architecture via tree–structured statistical models: the hidden Markov tree models. *New Phytologist*, 166(3):813–825, 2005. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.2005.01405.x/full. 51, 57, 60

P. Eades. *Drawing free trees*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991. 49

Y. Ephraim and N. Merhav. hidden Markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1003838. 61

R. Fernandez. *Reconstruction tridimensionnelle et suivi de lignées cellulaires à partir d'images de microscopie laser: application à des tissus végétaux*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2010. 53

R. Fernandez, P. Das, V. Mirabet, E. Moscardi, J. Traas, J.-L. Verdeil, G. Malandain, and C. Godin. Imaging plant growth in 4D: robust tissue reconstruction and lineaging at cell resolution. *Nature Methods*, 7(7):547–553, 2010. URL http://www.nature.com/nmeth/journal/v7/n7/abs/nmeth.1472.html. 53, 54

T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. URL http://onlinelibrary.wiley.com/doi/10.1002/spe.4380211102/abstract. 49

L. E. Gatsuk, O. V. Smirnova, L. I. Vorontzova, L. B. Zaugolnova, and L. A. Zhukova. Age states of plants of various growth forms: A review. *Journal of Ecology*, 68(2):pp. 675–696, 1980. ISSN 00220477. URL http://www.jstor.org/stable/2259429. 55

C. Godin and Y. Caraglio. A multiscale model of plant topological structures. *Journal of Theoretical Biology*, 191(1):1–46, 1998. URL http://www.sciencedirect.com/science/article/pii/S0022519397905610. 47, 48, 55, 58

P. Haccou, P. Jagers, and V. A. Vatutin. *Branching processes: variation, growth, and extinction of populations*. Cambridge University Press, 2005. 60

F. Hallé, R. A. Oldeman, P. B. Tomlinson, et al. *Tropical trees and forests: an architectural analysis*. Springer-Verlag., 1978. 55

J. D. Hunter. Matplotlib: A 2D graphics environment. *IEEE Transactions on Computing in Science & Engineering*, 9(3):0090–95, 2007. URL http://doi.ieeecomputersociety.org/10.1109/mcse.2007.55. 51

N. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*. Wiley New York, 1997. 61

M. Kimmel and D. E. Axelrod. Branching processes in biology. interdisciplinary applied mathematics 19, 2002. 60

J. Legrand. *Vers une compréhension multi-échelle du development floral: des réseaux auxiniques aux patrons de la dynamique cellulaire*. PhD thesis, École Normale Supérieure de Lyon, 2014. 53, 54, 55

W. Nultsch. *Botanique générale*. De Boeck Supérieur, 1998. 52

V. Olariu, D. Coca, S. A. Billings, P. Tonge, P. Gokhale, P. W. Andrews, and V. Kadirkamanathan. Modified variational bayes EM estimation of hidden Markov tree model of cell lineages. *Bioinformatics*, 25(21):2824–2830, 2009. URL http://bioinformatics.oxfordjournals.org/content/25/21/2824.short. 51, 62

F. Ramírez and T. L. Davenport. Mango (*Mangifera indica* L.) flowering physiology. *Scientia Horticulturae*, 126(2):65–72, 2010. URL http://www.sciencedirect.com/science/article/pii/S0304423810002992. 58

E. M. Reingold and J. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, SE-7(2):223–228, 1981. doi: 10.1109/TSE.1981.234519. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1702828. 49

S. Sabatier, D. Barthélémy, I. Ducousso, and É. Germain. Modalités d'allongement et morphologie des pousses annuelles chez le noyer commun, Juglans regia l. 'lara'; (Juglandaceae). *Canadian Journal of Botany*, 76(7):1253–1264, 1998. doi: 10.1139/b98-055. URL http://dx.doi.org/10.1139/b98-055. 57

K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995. URL http://www.sciencedirect.com/science/article/pii/S1045926X85710130. 49

R. Tamassia. *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007. ISBN 1584884126. 48, 49, 50

J. Q. Walker. A node-positioning algorithm for general trees. *Software: Practice and Experience*, 20(7):685–705, 1990. URL http://onlinelibrary.wiley.com/doi/10.1002/spe.4380200705/abstract. 49, 50

Chapter 3

Semi-parametric Hidden Markov Out-Tree (HMOT) models for cell lineage analysis

Abstract An enlarged family of Hidden Markov Out-Tree (HMOT) models is introduced. Contrarily of state-of-the-art HMOT models, in theses models child vertices are not independent given their parent vertex and the number of children per parent is random. The upward-downward smoothing algorithm which in particular enables to implement efficiently the E-step of the Expectation-Maximization (EM) algorithm and the dynamic programming algorithm for the restoration of the most probable state tree are derived for this family of models. The interest of such models is illustrated on cell lineages in floral meristems where non-parametric generation distributions are coupled with parametric observation models in order to define semi-parametric HMOT models

Keywords cell lineage; dynamic programming algorithm; hidden Markov out-tree model; smoothing algorithm; upward-downward algorithm; viterbi-algorithm;

Contents

3.1 Introduction **67**
3.2 Definitions **68**
3.2.1 Markov Out-Tree (MOT) models 68
3.2.2 Hidden Markov Tree (HMT) models 72
3.3 Computational methods for Hidden Markov Out-Tree (HMOT) models **73**
3.3.1 Upward-downward smoothing algorithm 73
3.3.2 Application of the EM algorithm 76
3.3.3 Dynamic programming restoration algorithm 78
3.4 Application to cell lineage trees **79**
3.4.1 Results 79
3.4.2 Discussions 83
References **86**

3.1 Introduction

Cell division is the key mechanism responsible for the development of plant organ (leaf, stem, flower, root). Data of interest typically consist of time series of growing organs at a cellular resolution obtained by live imaging. Considering that various geometrical and morphological characteristics of cells can be extracted from these images , we assume that organogenesis can be described using a small number of cell categories referred to as cell identities in the following. In order to study the emergence of new cell identities during organogenesis, we choose to focus tree-indexed data corresponding to lineage forests extracted from the time series of tissues corresponding to growing organs. The only spatial structuring taken into account thus corresponds to siblings with respect to a given parent cell. This is supported by the fact that cell topology is only affected by division, which makes the local spatial information taken into account in this approach highly relevant.

The Hidden Markov Out-Tree (HMOT) models were introduced by Crouse et al. (1998) as a direct generalization of Hidden Markov Chain (HMC) models to tree-indexed data. In a HMOT model, the non-observable states, assimilated to the cell identities in our application context, are arranged as a directed tree, whose topology duplicates those of the observed data (i.e. the cell characteristics). These initially proposed models have the same parametrization as first-order HMC models. This is the consequence of two main assumptions: the tree topology and in particular the vertex out-degree combinatorics is not represented in the model parametrization and the children are independent given their parent. These two strong assumptions enable efficient algorithms to be designed both for estimation of an HMOT model and for the restoration of the most probable state out-tree (see Durand et al., 2004). We here introduce an enlarged family of HMOT models that relax these two assumptions. Since we focus here on binary trees this family is presented in its semi-parametric version where semi-parametric stands for parametric modeling of the observation process and non-parametric modeling of the state process. For this family of models, the application of the Expectation-Maximization (EM) algorithm is straightforward and technical difficulties are concentrated in the design of the upward-downward smoothing algorithm that enables the implementation of the E-step of the EM algorithm, and the dynamic programming algorithm for the restoration of the most probable state tree. The upward-downward algorithm was initially proposed by Ronen et al. (1995) for the estimation of Markov Out-Tree (MOT) models with missing data and then adapted to the case of HMOT models by Crouse et al. (1998). This initially proposed algorithm is the analogous of the forward-backward algorithm proposed by Baum et al. (1970) for HMC models (see Ephraim and Merhav, 2002, for an overview of Hidden Markov Model (HMM) and associated statistical methods). Like the forward and backward recursions, the upward and downward recursions are not numerically stable. To overcome this problem, Durand et al. (2004) proposed an upward-downward algorithm which is a true smoothing algorithm and therefore numerically stable.

The remainder of this chapter is organized as follows. HMOT for ordered and unordered tree-indexed data are defined in section 3.2. The dedicated upward-downward algorithm which is a true smoothing algorithm, the EM algorithm and the dynamic

programming restoration algorithm are derived in section 3.3. These models are applied and discussed in the case of early stages of flower development in section 3.4.

3.2 Definitions

Data of interest are univariate tree-indexed data $\bar{x} = (x_t)_{t\in\mathcal{T}}$ – or more generally multivariate tree-indexed data noted $\bar{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t\in\mathcal{T}}$ – where $\mathcal{T} \subset \mathbb{N}$ is the set of vertices of a directed tree-graph $\tau = (\mathcal{T}, \mathcal{E})$ and $\mathcal{E} \subset \mathcal{T} \times \mathcal{T} \setminus \mathcal{R}$ the set of directed edges representing lineage relationships between vertices. $\mathcal{R}$ represent the set of roots and $\mathcal{L}$ the set of leaves of τ. Let $\mathrm{pa}(.)$ denote the parent, $\mathrm{ch}(.)$ the child set, $\mathrm{de}(.)$ the descendant set and $\mathrm{nd}(.)$ the non-descendant set of a vertex. These notations also apply to set of vertices (see Lauritzen, 1996, for graph terminology). Capitalized versions indicate the closure of the corresponding notation,

$$\forall t \in \mathcal{T},\ \mathrm{De}(t) = \mathrm{de}(t) \cup \{t\}.$$

For any set $\mathcal{A} \subseteq \mathcal{T}$, $\bar{x}_{\mathcal{A}}$ denote the subset of $\bar{x}$ obtained by only considering the vertices in $\mathcal{A}$,

$$\forall \mathcal{A} \subseteq \mathcal{T},\ \bar{x}_{\mathcal{A}} = (x_t)_{t\in\mathcal{A}}.$$

The notation $\bar{n} = (n_t)_{t\in\mathcal{T}}$ designates the univariate tree-indexed data indexed by the same tree as $\bar{x}$ and corresponding to the number of children of each vertex,

$$\forall t \in \mathcal{T},\ n_t = |\mathrm{ch}(t)|.$$

3.2.1 Markov Out-Tree (MOT) models

We here assume that $\bar{x} = (x_t)_{t\in\mathcal{T}}$, or more generally $\bar{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t\in\mathcal{T}}$, and $\bar{n} = (n_t)_{t\in\mathcal{T}}$ are the outcomes of a random process. Note that in the following τ is considered *sensu stricto* as a tree and that in a forest, trees are considered as independent and identically distributed. Let r denote the only root of τ. Let us first consider the simple case where:

- $\bar{x}$ is the realization of a $\mathcal{X}$-valued stochastic process $\bar{X} = (X_t)_{t\in\mathcal{T}}$ such that $\mathcal{X} \subset \mathbb{N}$ is called the state space and each value $x \in \mathcal{X}$ is called state. This process is therefore called this state process.
- $\bar{n}$ is the realization of a $\mathcal{N}$-valued stochastic process $\bar{N} = (N_t)_{t\in\mathcal{T}}$ with $\mathcal{N} \subset \mathbb{N}$. This process is called generation process.

These considerations raise the question of modeling the joint distribution

$$P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right). \tag{3.1}$$

Markov Tree (MT) models are parsimonious models relying on local dependence assumptions in $\bar{X}$ with respect to the tree topology. The induced conditional independence hypotheses are called Markov properties. The order of a MT model is related to these Markov properties and refers to the number of ancestors or predecessors considered to

model siblings. Contrarily to sequences where the structure is unchanged whichever the chosen direction, directed trees are non-symmetrical structures. In fact, as presented by Durand et al. (2005), one can distinguish two types of MT models of order 1:

- Markov In-Trees (MITs) studied by Bacciu et al. (2010) where a vertex is modeled given its children,
- MOTs introduced by Ronen et al. (1995) where children are modeled given their parent.

We here focus on the case of MOT models of order 1, modeling tree-indexed data considering only child-parent local dependencies. The key difference with the models of Crouse et al. (1998) and Durand et al. (2004) is that we now assume that the children vertices are not independent given their parent vertex and that the tree topology is partially represented through the generation process.

3.2.1.1 Markov Ordered Out-Tree (MOOT) models

The usual Markov property of order 1 is expressed in trees as the assumption that the state process at vertex t is independent of its non-descendants processes given its parent state process

$$\forall t \in \mathcal{T},\ X_t \perp\!\!\!\perp N_{nd(t)\setminus pa(t)}, \bar{X}_{\mathrm{nd}(t)\setminus\{\mathrm{pa}(t)\}} \,\big|\, X_{\mathrm{pa}(t)}\ .$$

Similarly, it is assumed that the generation process at vertex t is independent of its non-descendants processes given its state process

$$\forall t \in \mathcal{T},\ N_t \perp\!\!\!\perp \bar{N}_{\mathrm{nd}(t)}, \bar{X}_{\mathrm{nd}(t)} \,|\, X_t.$$

The preceding assumptions induce the following factorization of the joint distribution (3.1)

$$P\left(\bar{X}=\bar{x},\bar{N}=\bar{n}\right) = P\left(X_r = x_r\right) \prod_{t\in\mathcal{T}} \left\{ P\left(\bar{X}_{\mathrm{ch}(t)} = \bar{x}_{\mathrm{ch}(t)} \,\middle|\, X_t = x_t, N_t = n_t\right) \times P\left(N_t = n_t \,|\, X_t = x_t\right)\right\}. \tag{3.2}$$

where the child set, $\mathrm{ch}\left(.\right)$, is considered as an ordered set. Considering (3.2), a Markov Ordered Out-Tree (MOOT) model is specified by:

- one initial distribution for the root vertex

  $$\pi_{x_r} = P\left(X_r = x_r\right),$$

  with $\sum_{x\in\mathcal{X}} \pi_x = 1$.
- as many composition distributions as states

  $$\forall t \in \mathcal{T},\ \Pi_{x_t}\left(\bar{x}_{\mathrm{ch}(t)}\right) = P\left(\bar{X}_{\mathrm{ch}(t)} = \bar{x}_{\mathrm{ch}(t)} \,\middle|\, X_t = x_t\right),$$

  with

  $$\forall x \in \mathcal{X}, \forall n \in \mathcal{N},\ \sum_{\bar{x}\in\mathcal{X}^n} \Pi_x^{(n)}\left(\bar{x}\right) = 1.$$

Number	Maximal degree		
of states	2	3	4
2	**13**	**29**	**61**
3	**38**	**119**	**362**
4	**83**	**339**	**1363**

Table 3.1 – Number of parameters of Markov ordered out-tree models as a function of the number of states and the maximal degree.

- as many generation distributions as states

$$\forall t \in \mathcal{T},\, \Gamma_{x_t}\left(n_t\right) = P\left(N_t = n_t \,|\, X_t = x_t\right)$$

with

$$\forall x \in \mathcal{X},\, \sum_{n \in \mathcal{N}} \Gamma_x\left(n\right) = 1.$$

Without any further hypotheses, there is a total of

$$|\mathcal{X}| - 1 + |\mathcal{X}|\left(|\mathcal{N}| - 1 + \sum_{n \in \mathcal{N}} \left\{|\mathcal{X}|^n - 1\right\}\right),$$

independent parameters to define (see table 3.1). But in practice, such models can be parsimoniously parametrized using:

- Markov Chain (MC) models and variants (Ephraim and Merhav, 2002) for each composition distribution. Ordered children can be viewed as a sequence for which local dependencies can be assumed leading to the factorization of the composition distribution.

- Parametric discrete univariate distributions (Johnson et al., 1993) for each generation distribution.

3.2.1.2 Markov Unordered Out-Tree (MUOT) models

Depending on the application context, one can either consider ordered or unordered trees. In our context focusing on cell division, the latter case is more relevant since both children appear at the same time. Considering no order structure among siblings is equivalent of assuming that composition probabilities are invariant under every permutation of children vertices,

$$\forall t \in \mathcal{T},\, \forall \sigma \in \mathfrak{S}\left[\mathrm{ch}(t)\right],\, \Pi_{x_t}^{(n_t)}\left(\bar{x}_{\mathrm{ch}(t)}\right) = \Pi_{x_t}^{(n_t)}\left(\bar{x}_{\sigma \cdot \mathrm{ch}(t)}\right).$$

Let $\boldsymbol{N}_t = (N_{t,x})_{x\in\mathcal{X}}$ denote the random vector of the number of children of vertex t in the different states $\mathcal{X}$, $\mathrm{I}(\cdot)$ the indicator function and $\boldsymbol{n}_t = (n_{t,x})_{x\in\mathcal{X}}$ an outcome of $\boldsymbol{N}_t$. As a consequence of the latter assumption

$$P\left(\boldsymbol{N}_t = \boldsymbol{n}_t \,|\, X_t = x_t\right) = \Gamma_{x_t}\left(n_t\right)\binom{n_t}{n_{t,0},\cdots,n_{t,|\mathcal{X}|-1}}\Pi_{x_t}\left(\bar{x}_{\mathrm{ch}(t)}\right),$$

where $\binom{\cdot}{\cdot,\cdots,\cdot}$ denote the multinomial coefficient. Adaptations of algorithms for MOOT models to the Markov Unordered Out-Tree (MUOT) models case are therefore straightforward and only require combinatorics arguments.

For MUOT models the marginal distribution (3.1) is factorized as follows

$$P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right) \propto P\left(X_r = x_r\right)\prod_{t\in\mathcal{T}} P\left(\boldsymbol{N}_t = \boldsymbol{n}_t | X_t = x_t,\right). \tag{3.3}$$

This factorization corresponds to the family of Multi-Type Branching Processes (MTBPs) introduced as a generalization of Watson and Galton (1875) processes (see Harris, 2002). In such MOT models, composition and generation distributions are replaced by the corresponding generation distributions

$$\forall t \in \mathcal{T},\, \Gamma_{x_t}\left(\boldsymbol{n}_t\right) = P\left(\boldsymbol{N}_t = \boldsymbol{n}_t | X_t = x_t\right),$$

with

$$\forall x \in \mathcal{X},\, \sum_{\boldsymbol{n}\in\mathcal{N}^{|\mathcal{X}|}} \Gamma_x\left(\boldsymbol{n}\right) = 1.$$

Hence, the total number of independent parameters drops to

$$|\mathcal{X}| - 1 + |\mathcal{X}|\left[\sum_{n\in\mathcal{N}}\left\{\binom{|\mathcal{X}|+n-1}{n}\right\} - 1\right],$$

where $\binom{\cdot}{\cdot}$ denote the binomial coefficient (see table 3.2). Their parametrization is highly related to the parametrization of discrete multivariate count models. In practice, two different cases can be considered:

Simple trees. In cell lineage trees $\mathcal{N} = \{1, 2\}$. Value 0 corresponds to the censuring at the end of the experiment since no cell death is observed. The combinatorics of child states is therefore reasonable for models with few states and non-parametric models can be considered.

General trees. The combinatorics induced by the variable and high number of child vertices in each state induces an inflation in the number of model parameters. In such cases, models can be parsimoniously parametrized using different parametric discrete multivariate distributions (Johnson et al., 1997) for each generation distribution.

Number of states	Maximal degree 2	3	4
2	**11**	**19**	**29**
3	**29**	**59**	**104**
4	**59**	**139**	**279**

Table 3.2 – Number of parameters of Markov unordered out-tree models as a function of the number of states and the maximal degree.

3.2.2 Hidden Markov Tree (HMT) models

Hidden Markov Tree (HMT) models introduced by Crouse et al. (1998) have the same parametrization of standard HMC models for sequences. As for HMC models (see Ephraim and Merhav, 2002, for more details), HMT models are not restricted to categorical variables but allow any types of random variables or vectors to be considered, at low cost in term of parameters.

In our case, a HMOT model can be viewed as a triplet of stochastic processes $(S_t, N_t, X_t)_{t\in\mathcal{T}}$ where:

- $\bar{S} = (S_t)_{t\in\mathcal{T}}$ is a $\mathcal{S}$-valued state process.
- $\bar{N} = (N_t)_{t\in\mathcal{T}}$ is a $\mathcal{N}$-valued generation process.
- $\bar{X} = (X_t)_{t\in\mathcal{T}}$ is a $\mathcal{X}$-valued process corresponding to the output or observation process. This process is related to $\bar{S}$ by a probabilistic function $f_{s_t}(x_t)$.

To simplify notations we will consider an univariate discrete output process in the following

$$\forall t \in \mathcal{T},\ f_{s_t}(x_t) = P(X_t = x_t \,|\, S_t = s_t),$$

with

$$\forall s \in \mathcal{S},\ \sum_{x\in\mathcal{X}} f_s(x) = 1 \text{ and } \mathcal{X} \subseteq \mathbb{N}.$$

It is assumed that the output process at vertex t only depends on the underlying state process at vertex t

$$\forall t \in \mathcal{T},\ X_t \perp\!\!\!\perp \bar{S}_{\mathcal{T}\setminus t}, \bar{X}_{\mathcal{T}\setminus t}, \bar{N}_{\mathcal{T}} \,|\, S_t. \tag{3.4}$$

Thus, for HMOT models, the marginal distribution (3.1) can be factorized as follows

$$\begin{aligned} P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right) &= \sum_{\bar{s}\in\mathcal{S}^{|\mathcal{T}|}} P\left(\bar{S} = \bar{s}, \bar{N} = \bar{n}, \bar{X} = \bar{x}\right) \\ &= \sum_{\bar{s}\in\mathcal{S}^{|\mathcal{T}|}} P\left(\bar{S} = \bar{s}, \bar{N} = \bar{n}\right) \prod_{t\in\mathcal{T}} f_{s_t}(x_t). \end{aligned} \tag{3.5}$$

Parametrization of HMOT models therefore only depends on that of the state process and the observation probabilities $f_{s_t}(x_t)$. Thus, with appropriate parametrization, the extension to the continuous or mixed multivariate case is straightforward.

We define Hidden Markov Ordered Out-Tree (HMOOT) models as HMT models where the state process is a MOOT model. Similarly, for Hidden Markov Unordered Out-Tree (HMUOT) models the corresponding state process is a MUOT.

3.3 Computational methods for Hidden Markov Out-Tree (HMOT) models

Since it is assumed *a priori* that any outcome value could be observed in any states, the state process $\bar{S}$ is not directly observable but only indirectly through the observation process $\bar{X}$. Therefore, the use of HMT models relies on two main algorithms:

- the smoothing algorithm, which computes the probabilities of being in each state for each vertex given the observed tree,
- the restoration algorithm, which compute the most probable state tree given the observed tree.

Parameter Maximum Likelihood (ML) inference can be done using the EM algorithm or its variants (see McLachlan and Peel, 2004, for more details) based on quantities computed in the smoothing algorithm.

In the remainder of this section, we derive algorithms for HMOOT and HMUOT models. Therefore we introduce the notion of transition distributions, noted

$$\forall t \in \mathcal{T},\ \Delta_{s_t}\left(\bar{s}_{ch(t)}\right) = P\left(N_t = n_t, \bar{S}_{ch(t)} = \bar{s}_{ch(t)} \middle| S_t = s_t\right),$$

with:

- For MOOT,
$$\forall t \in \mathcal{T},\ \Delta_{s_t}\left(\bar{s}_{ch(t)}\right) = \Gamma_{s_t}(n_t)\,\Pi_{s_t}^{(n_t)}\left(\bar{s}_{ch(t)}\right).$$
- For MUOT,
$$\forall t \in \mathcal{T},\ \Delta_{s_t}\left(\bar{s}_{ch(t)}\right) = \frac{\Gamma_{s_t}(\boldsymbol{n}_t)}{\binom{n_t}{n_{t,0},\cdots,n_{t,|\mathcal{S}|-1}}}.$$

In particular, $\Delta_{s_t}(\emptyset)$ denote the probability of having 0 children for a vertex in state s_t.

3.3.1 Upward-downward smoothing algorithm

The aim of the smoothing algorithm is to compute the smoothed probabilities,

$$\xi_t(s) = P\left(S_t = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}\right),$$

for each vertex t and each state s. Such probabilities can be recursively computed using a downward pass (i.e. vertices are taken successively from root to leaves) requiring the upward probabilities,

$$\beta_t(s) = P\left(S_t = s \,\middle|\, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right),$$

which are computed in an upward pass (i.e. vertices are taken successively from leaves to root).

Preprocessing As for Independent Hidden Markov Out-Tree (HIMOT) models discussed by Durand et al. (2004), the upward recursion requires the preliminary knowledge of the marginal state distributions for each vertices and each states, which can be computed in an initial downward recursion (Durand et al., 2004). For an observed tree, this preprocessing is initialized at the root vertex with,

$$\forall s \in \mathcal{S},\ P(S_r = s) = \pi_s, \tag{3.6}$$

Afterward, the computation is done for all parent vertices taken from the root to the leaf vertices,

$$\begin{aligned} \forall t \in \mathcal{T}, \forall \bar{s} \in \mathcal{S}^{n_t},\ P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s}\right) &= \sum_{s \in \mathcal{S}} \Delta_s(\bar{s})\, P(S_t = s), \\ \forall c \in \mathrm{ch}(t), \forall s \in \mathcal{S},\ P(S_c = s) &= \sum_{\bar{s} \in \mathcal{S}^{n_t - 1}} P\left(S_{\mathrm{ch}(t)\setminus\{c\}} = \bar{s}, S_c = s\right). \end{aligned} \tag{3.7}$$

Upward recursion The upward recursion is initialized for each leaf vertex by

$$\begin{aligned} \forall l \in \mathcal{L}, \forall s \in \mathcal{S},\ \beta_l(s) &= P(S_l = s \mid X_l = x_l, N_l = 0) \\ &\propto P(X_l = x_l, N_l = 0 \mid S_l = s)\, P(S_l = s) \qquad (\star) \\ &\propto \Delta_s(\emptyset)\, f_s(x_l)\, P(S_l = s), \end{aligned}$$

where $(\star)$ indicates the use of the Bayes' rule.

Then, for each of the remaining vertices taken upwards, we have the following recur-

sion

$$
\begin{aligned}
&\forall t \in \mathcal{T} \setminus \mathcal{L}, \forall s \in \mathcal{S}, \\
&\quad \beta_t(s) = P\left(S_t = s \,\middle|\, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right) \\
&\qquad \propto \sum_{\bar{s} \in \mathcal{S}^{n_t}} P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s}, S_t = s, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right) \qquad (\star) \\
&\qquad \propto \sum_{\bar{s} \in \mathcal{S}^{n_t}} \Big\{ P(S_t = s)\, P\left(N_t = n_t, \bar{S}_{\mathrm{ch}(t)} = \bar{s} \,\middle|\, S_t = s\right) P(X_t = x_t \,|\, S_t = s) \\
&\qquad\qquad \times \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} P\left(\bar{X}_{\mathrm{De}(c)} = \bar{x}_{\mathrm{De}(c)}, \bar{N}_{\mathrm{De}(c)} = \bar{n}_{\mathrm{De}(c)} \,\middle|\, S_c = s_c\right) \Big\} \\
&\qquad \propto \sum_{\bar{s} \in \mathcal{S}^{n_t}} \Big\{ P(S_t = s)\, \Delta_s(\bar{s})\, f_s(x_t) \\
&\qquad\qquad \times \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} \frac{P\left(S_c = s_c \,\middle|\, \bar{X}_{\mathrm{De}(c)} = \bar{x}_{\mathrm{De}(c)}, \bar{N}_{\mathrm{De}(c)} = \bar{n}_{\mathrm{De}(c)}\right)}{P(S_c = s_c)} \Big\} \qquad (\star) \\
&\qquad \propto P(S_t = s)\, f_s(x_t) \sum_{\bar{s} \in \mathcal{S}^{n_t}} \Delta_s(\bar{s}) \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} \frac{\beta_c(s_c)}{P(S_c = s_c)}, \qquad (3.8)
\end{aligned}
$$

Let ϕ_t be the normalization constant for each vertex upward probability distribution, the different $(\star)$ give

$$
\begin{aligned}
\forall l \in \mathcal{L},\ \phi_l &= P(X_l = x_l, N_l = 0) \\
&= P\left(\bar{X}_{\mathrm{De}(l)} = \bar{x}_{\mathrm{De}(l)}, \bar{N}_{\mathrm{De}(l)} = \bar{n}_{\mathrm{De}(l)}\right),
\end{aligned}
$$

for each leaf vertices, and

$$
\forall t \in \mathcal{T} \setminus \mathcal{L},\ \phi_t = \frac{P\left(\bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right)}{\prod_{c \in \mathrm{ch}(t)} P\left(\bar{X}_{\mathrm{De}(c)} = \bar{x}_{\mathrm{De}(c)}, \bar{N}_{\mathrm{De}(c)} = \bar{n}_{\mathrm{De}(c)}\right)}
$$

for the internal vertices. Since,

$$
\begin{aligned}
\prod_{t \in \mathcal{T}} \phi_u &= \frac{\prod_{t \in \mathcal{T}} P\left(\bar{X}_{\mathrm{De}(t)} = x_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = n_{\mathrm{De}(t)}\right)}{\prod_{t \in \mathcal{T} \setminus \{r\}} P\left(\bar{X}_{\mathrm{De}(t)} = x_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = N_{\mathrm{De}(t)}\right)} \\
&= P\left(\bar{X}_{\mathrm{De}(r)} = x_{\mathrm{De}(r)}, \bar{N}_{\mathrm{De}(r)} = \bar{n}_{\mathrm{De}(r)}\right) \\
&= P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right),
\end{aligned}
$$

the log-likelihood can be computed as a byproduct of the upward recursion. Among other potential applications, this computation allows the monitoring of the EM algorithm convergence (McLachlan and Krishnan, 2007) and model selection (Claeskens and Hjort, 2008).

Downward recursion The downward recursion of the smoothing algorithm is initialized at the root vertex by,

$$\forall s \in \mathcal{S},\, \xi_r(s) = P\left(S_r = s \,\middle|\, \bar{X} = \bar{x}, \bar{N} = \bar{n}\right) = \beta_r(s). \tag{3.9}$$

For all remaining vertices let us remark first that

$$\begin{aligned}
&\forall t \in \mathcal{T} \setminus \mathcal{L}, \forall s \in \mathcal{S}, \forall \bar{s} \in \mathcal{S}^{n_t},\\
P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s} \,\middle|\, S_t = s, \bar{X} = \bar{x}, \bar{N} = \bar{n}\right) &= P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s} \,\middle|\, S_t = s, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right)\\
&= \frac{P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s}, S_t = s, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right)}{P\left(S_t = s, \bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}\right)}\\
&= \frac{P(S_t = s)\, f_s(x_t)\, \Delta_s(\bar{s})}{\beta_t(s)\, \phi_t} \prod_{\substack{s_c \in \bar{s}\\ c \in ch(t)}} \frac{\beta_c(s_c)}{P(S_c = s_c)},
\end{aligned} \tag{3.10}$$

using (3.4) and previous calculations in (3.8). We therefore obtain directly the following downward recursion,

$$\begin{aligned}
&\forall t \in \mathcal{T}, \forall \bar{s} \in \mathcal{S}^{n_t},\\
&P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s} \,\middle|\, \bar{X} = \bar{x}, \bar{N} = \bar{n}\right) = \sum_{s \in \mathcal{S}} \xi_t(s) \frac{P(S_t = s)\, f_s(x_t)\, \Delta_s(\bar{s})}{\beta_t(s)\, \phi_t} \prod_{\substack{s_c \in \bar{s}\\ c \in ch(t)}} \frac{\beta_c(s_c)}{P(S_c = s_c)},\\
&\forall c \in \mathrm{ch}(t), \forall s \in \mathcal{S},\, \xi_c(s) = \sum_{\bar{s} \in \mathcal{S}^{n_t - 1}} P\left(S_{\mathrm{ch}(t)\setminus\{c\}} = \bar{s}, S_c = s \,\middle|\, \bar{X} = \bar{x}, \bar{N} = \bar{n}\right).
\end{aligned} \tag{3.11}$$

3.3.2 Application of the Expectation-Maximization algorithm

With reference to HIMOT models (see Crouse et al., 1998; Durand et al., 2004), the adaptation of the EM algorithm is straightforward. Let us consider the complete data where both the outputs $\bar{x}$ and the states $\bar{s}$ of the underlying MT model are observed. Note that in this section $\bar{x}$ is considered as a forest. The EM algorithm iteratively modifying model parameters in order to increase the likelihood, let θ be the vector of model parameters and $\theta^{(k)}$ denote the current value of θ at iteration k. The conditional expectation of the complete-data log-likelihood is given by

$$Q\left(\theta \,\middle|\, \theta^{(k)}\right) = E\left\{\log L\left(\bar{S}, \bar{X}; \theta\right) \,\middle|\, \bar{X} = \bar{x}, \bar{N} = \bar{n}; \theta^{(k)}\right\}.$$

Let $\theta = \theta_O \uplus \theta_L \biguplus_{s \in \mathcal{S}} \theta_s$ with $\theta_O = \{f_s(x)\}_{x \in \mathcal{X}, s \in \mathcal{S}}$, $\theta_R = \{\pi_s\}_{s \in \mathcal{S}}$, and

$$\forall s \in \mathcal{S},\, \theta_s = \{\Delta_s(\bar{s})\}_{\bar{s} \in \bar{\mathcal{S}}},$$

where

$$\bar{\mathcal{S}} = \bigcup_{n \in \mathcal{N}} \mathcal{S}^n.$$

Using (3.2) and (3.5), this conditional expectation can be rewritten as a sum of terms, each term depending on a given subset of parameters

$$Q\left(\theta \middle| \theta^{(k)}\right) = Q_O\left(\theta_O \middle| \theta^{(k)}\right) + Q_R\left(\theta_R \middle| \theta^{(k)}\right) + \sum_{s\in\mathcal{S}} Q_s\left(\theta_s \middle| \theta^{(k)}\right),$$

with

$$Q_O\left(\theta_O \middle| \theta^{(k)}\right) = \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}} P\left(X_t = x_t, S_t = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}; \theta^{(k)}\right) \log f_s\left(x_t\right). \tag{3.12}$$

$$Q_R\left(\theta_R \middle| \theta^{(k)}\right) = \sum_{r\in\mathcal{R}}\sum_{s\in\mathcal{S}} P\left(S_r = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}; \theta^{(k)}\right) \log \pi_s \tag{3.13}$$

$\forall s \in \mathcal{S}$,

$$Q_s\left(\theta_s \middle| \theta^{(k)}\right) = \sum_{t\in\mathcal{T}}\sum_{\bar{s}\in\mathcal{S}^{n_t}} P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s}, S_t = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}; \theta^{(k)}\right) \log \Delta_s\left(\bar{s}\right) \tag{3.14}$$

During the M-step the initial and transition probabilities are directly obtained respectively from the maximization of (3.13),

$$\begin{aligned}\forall s \in \mathcal{S}, \pi_s^{(k+1)} &\propto \sum_{r\in\mathcal{R}} P\left(S_r = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}\right) \\ &\propto \sum_{r\in\mathcal{R}} \xi_r\left(s\right),\end{aligned}$$

given by (3.9), and the maximization of (3.14),

$$\begin{aligned}\forall n \in \mathcal{N}, \forall s \in \mathcal{S}, \forall \bar{s} \in \mathcal{S}^n, \Delta_s^{(k+1)}\left(\bar{s}\right) &\propto \sum_{\substack{t\in\mathcal{T}\\ n_t = n}} P\left(\bar{S}_{\mathrm{ch}(t)} = \bar{s}, S_t = s \middle| \bar{X} = \bar{x}, \bar{N} = \bar{n}\right) \\ &\propto \sum_{\substack{t\in\mathcal{T}\\ n_t = n}} \xi_t\left(s\right) \frac{P\left(S_t = s\right) f_s\left(x_t\right) \Delta_s\left(\bar{s}\right)}{\beta_t\left(s\right)\phi_t} \prod_{\substack{s_c\in\bar{s}\\ c\in ch(t)}} \frac{\beta_c\left(s_c\right)}{P\left(S_c = s_c\right)}.\end{aligned}$$

given by (3.7), (3.8), and (3.11). Note that for algorithmic aspects the computation of the transient quantity

$$\forall t \in \mathcal{T}\setminus\mathcal{L}, \forall s \in \mathcal{S},\ \varphi_t\left(s\right) = \frac{\beta_t\left(s\right)\phi_t}{f_s\left(x_t\right) P\left(S_t = s\right)} = \sum_{\bar{s}\in\mathcal{S}^{n_t}} \Delta_s\left(\bar{s}\right) \prod_{\substack{s_c\in\bar{s}\\ c\in ch(t)}} \frac{\beta_c\left(s_c\right)}{P\left(S_c = s_c\right)},$$

is done during the upward recursion of the smoothing algorithm for all non-leaf vertices.

In the case of the Monte Carlo EM (MCEM) (Wei and Tanner, 1990) algorithm, the E-step of the EM algorithm is replaced by an approximation of the completed log-likelihood. In the case of HMOT models this approximation is computed using a downward simulation of the state process given the observation and generation processes initialized at root vertices with (3.9) and proceeding with vertices taken downwards considering (3.10). The M-step of the MCEM algorithm is based on count data extracted from the simulated states.

3.3.3 Dynamic programming restoration algorithm

The objective of this Viterbi-like algorithm is to restore the most probable state tree $\bar{s}$ associated with the observed tree $\bar{x}$. This is a major diagnostic tool in many applications of hidden Markovian models as in most applications, the knowledge of the hidden states provides an interpretation of the data.

The Viterbi upward recursion for a HMOT is initialized for each leaf vertex by

$$\begin{aligned}\forall l \in \mathcal{L},\, \forall s \in \mathcal{S},\, \delta_t(s) &= P\left(X_t = x_t N_t = 0 \,|\, S_t = s\right) \\ &= \Delta_s(\emptyset)\, f_s(x_t)\,.\end{aligned}$$

Then, for each of the internal vertices taken upward, we have the following recursion

$$\begin{aligned}&\forall t \in \mathcal{T} \setminus \mathcal{L}, \forall s \in \mathcal{S}, \\ &\quad \delta_t(s) = \max_{\bar{s}_{\mathrm{de}(t)}} \left\{ P\left(\bar{X}_{\mathrm{De}(t)} = \bar{x}_{\mathrm{De}(t)}, \bar{N}_{\mathrm{De}(t)} = \bar{n}_{\mathrm{De}(t)}, \bar{S}_{\mathrm{de}(t)} = \bar{s}_{\mathrm{de}(t)} \,\middle|\, S_t = s\right)\right\} \\ &\quad = \max_{\bar{s}_{\mathrm{de}(t)}} \left\{ P\left(\bar{X}_{\mathrm{de}(t)} = \bar{x}_{\mathrm{de}(t)}, \bar{N}_{\mathrm{de}(t)} = \bar{n}_{\mathrm{de}(t)}, \bar{S}_{\mathrm{de}(\mathrm{ch}(t))} = \bar{s}_{\mathrm{de}(\mathrm{ch}(t))} \,\middle|\, \bar{S}_{\mathrm{ch}(t)} = \bar{s}_{\mathrm{ch}(t)}\right)\right. \\ &\qquad \left. \times\, P\left(X_t = x_t \,|\, S_t = s\right) P\left(N_t = n_t, \bar{S}_{\mathrm{ch}(t)} = \bar{s}_{\mathrm{ch}(t)} \,\middle|\, S_t = s\right)\right\} \\ &\quad = \max_{\bar{s}_{\mathrm{de}(t)}} \left\{ \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} P\left(\bar{X}_{\mathrm{De}(c)} = \bar{x}_{\mathrm{De}(c)}, \bar{N}_{\mathrm{De}(c)} = \bar{n}_{\mathrm{De}(c)}, \bar{S}_{\mathrm{de}(c)} = \bar{s}_{\mathrm{de}(c)} \,\middle|\, S_c = s_c\right)\right. \\ &\qquad \left. \times \Delta_s\left(\bar{s}_{\mathrm{ch}(t)}\right)\right\} f_s(x_t) \\ &\quad = \max_{\bar{s}_{\mathrm{ch}(t)}} \left\{ \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} \max_{\bar{s}_{\mathrm{de}(c)}} \left\{ P\left(\bar{X}_{\mathrm{De}(c)} = \bar{x}_{\mathrm{De}(c)}, \bar{N}_{\mathrm{De}(c)} = \bar{n}_{\mathrm{De}(c)}, \bar{S}_{\mathrm{de}(c)} = \bar{s}_{\mathrm{de}(c)} \,\middle|\, S_c = s_c\right)\right\}\right. \\ &\qquad \left. \times \Delta_s\left(\bar{s}_{\mathrm{ch}(t)}\right)\right\} f_s(x_t) \\ &\quad = \max_{\bar{s}_{\mathrm{ch}(t)}} \left\{ \Delta_s\left(\bar{s}_{\mathrm{ch}(t)}\right) \prod_{\substack{s_c \in \bar{s} \\ c \in ch(t)}} \delta_c(s_c) \right\} f_s(x_t)\,. \end{aligned} \tag{3.15}$$

The probability of the observed tree $\bar{x}$ jointly with the most probable state tree is $\prod_{r \in \mathcal{R}} \max_s \{\delta_r(s)\, \pi_s\}$. The recursion (3.15) is equivalent to the upward recursion (3.8) where the summation on the states are replaced by maximization. To retrieve the most probable state tree, it is necessary to store for each vertex t and each state s the optimal states corresponding to each of the children. The backtracking procedure consists in tracing downward along the backpointers from the optimal root state to the optimal leaf states.

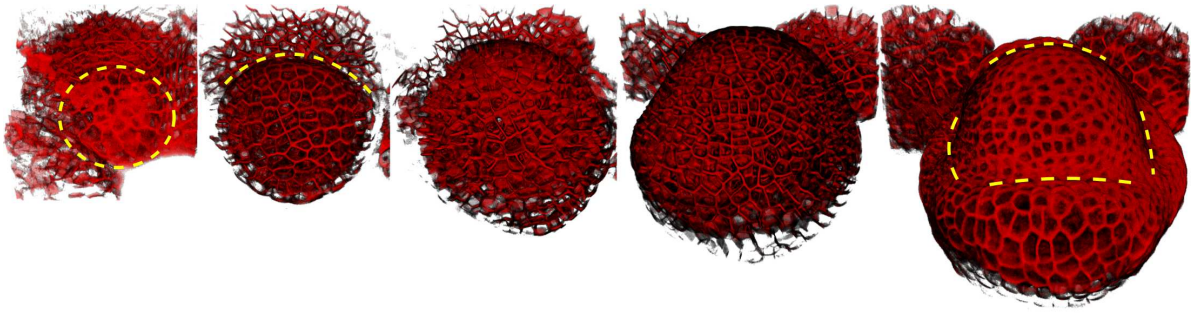


Figure 3.1 – Example of a $3D + t$ *images and meristem early stages. The* 3D *images reconstructed after the multi-angle acquisition are displayed according to time. Images, from left to right, were respectively taken at* 0h, 26h, 44h, 56h *and* 69h *after the beginning of the experiment according to Das et al. (2009) experimental design. The red color is due to a marker of cell walls (vital dye* $FM4-64$*). We here focus on the first three stages of floral mersitem development. The stage* 1 *is observed at* 0h *when there is no clear distinction between floral meristem (confined in the dotted circle) and the inflorescence meristem. The stage* 2 *is observed at* 26h *with clear boundary (represented by the dotted line) between the inflorescence meristem and the floral meristem. The early stage* 3 *is observed at* 69h *when sepals start emerging at the sides of the floral meristem (represented by the dotted line).*

3.4 Application to cell lineage trees

3.4.1 Results

The use of HMOT models is illustrated by the analysis of early stages of flower development which is usually described as a series of morphological events (Smyth et al., 1990). Only the first three stages were observed during the experiment (see figures 3.1, 3.2):

Stage 1 corresponds to the floral meristem development from the initiation as a small bulge on the flank of the inflorescence meristem.

Stage 2 starts when the floral bud is separated from the inflorescence meristem by a small crease between the two meristems.

Stage 3 is characterized by the emergence of the sepals from the sides of the floral meristem, growing to overlie the primordium.

In a first step, cell identities were inferred on the basis of the following cell characteristics:

- volume,
- epidermal surface,
- external surface,

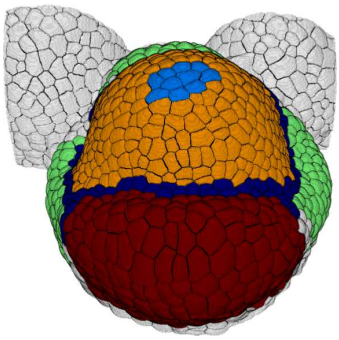


Figure 3.2 – Spatial regions on the floral meristem at stage 3*. At stage* 3 *spatial regions are relatively well defined and can be assigned manually. The most advanced sepal corresponds to red cells and the latest ones to green cells. Orange cells correspond to the central dome and light blue ones to the central zone. Boundary cells are in dark blue.*

- inertia values (according to three axes),
- principal and secondary curvatures.

These characteristics have been modeled independently using univariate Gaussian or Gamma distributions (depending on the observation space). Concerning the generation distributions we used a non-parametric dependence models corresponding to the saturated model.

The number of states was selected using the Bayesian Information Criterion (BIC) (Schwarz, 1978). Although BIC properties have not been established in this context, it is used frequently (Durand et al., 2005). This penalized likelihood criterion makes a trade-off between fit of model to the data and model parsimony and favored a 4-state model (model $\mathcal{M}_0$). Epidermal surface, internal surface, volume and curvatures of the cells are structuring observed variables in this model since the estimated observation distributions for the different states are well separated for this five characteristics (see figure 3.3). These observation distributions allowed us to characterize the different states:

State 0 **and** 3 correspond to large cells and are mostly differentiated by their curvatures (both negative for state 0 and positive for state 3).

State 1 corresponds to small cells with both curvatures almost of the same norm and mostly negative, this being typical characteristics of saddle forms.

State 2 is in-between considering size but with clearly positive curvatures corresponding to the dome area.

In contrast, states do not have marked differences with respect to anisotropies.

Using the restoration algorithm, the spatial regions that emerged from the cell identity labeling were then characterized (see figures 3.2,3.4):

Central dome and zone have been assigned to state 2.

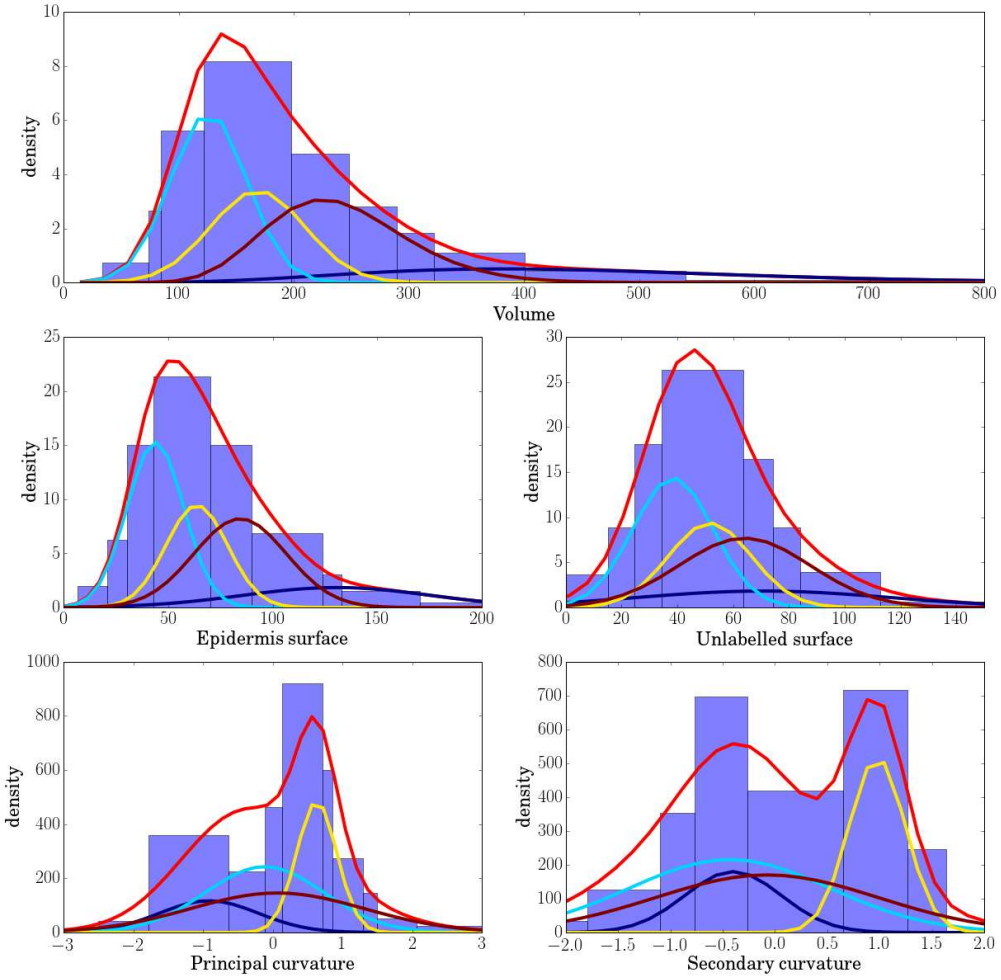


Figure 3.3 – Observation distributions of the hidden Markov unordered out-tree model. Observed histogram and mixture of observed distributions for each structuring characteristic. State 0 *is in dark blue, state* 1 *in light blue, state* 2 *in yellow and state* 3 *in dark red. Surfaces and volumes are modeled by Gamma distributions and curvatures by Gaussian distribution. Combining separations induced by surfaces and volume in one hand and curvatures in the other hand indicates that states are well separated using only these characteristics.*

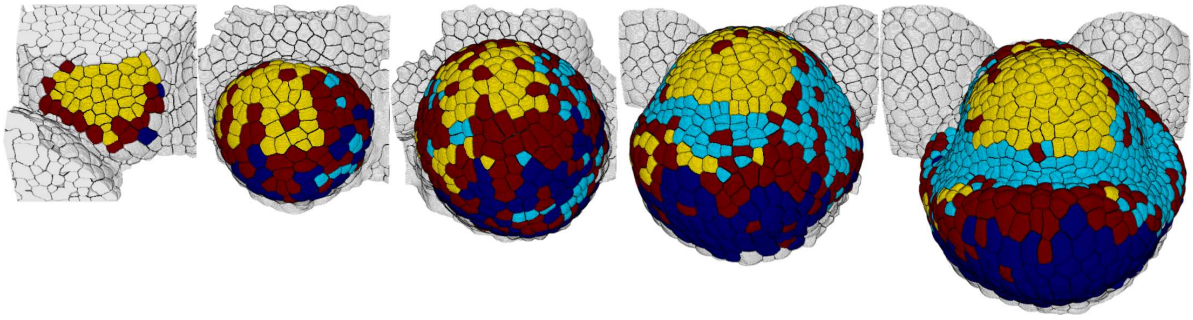


Figure 3.4 – Restoration of hidden states for using the restoration algorithm for Hidden Markov Unordered Out-Tree (HMUOT) models. Images, from left to right, were respectively taken at 0h, 26h, 44h, 56h *and* 69h *after the beginning of the experiment. Spatial projection of the four states obtained using for the HMUOT estimated using epidermis surface, internal surface, volume, curvatures and inertia as cell characteristics. State* 0 *is in dark blue, state* 1 *in light blue, state* 2 *in yellow and state* 3 *in dark red. Sepals are mostly identified by considering state* 0 *and* 3*, the dome by state* 2 *and boundary cells by state* 1*.*

Sepals have been split into states 0 and 3.

Boundary zone has been assigned to state 1.

State 2 is the main state of the first time point and presents a high spatio-temporal coherence from 0h to 69h. Despite an early stage of meristem differentiation at 0h, few cells are already assigned to putative sepals. At subsequent time points, the multiplication of sepal and dome cells with the apparition of boundary cells, delimiting the frontier between sepals and the dome zone, is observed. The apparition of boundary cells is unobtrusive until 44h but significant as the continuous border is clearly identified starting from 56h. In fact, at this time point and the next, state 2 clearly corresponds to the dome, states 0 and 3 to the sepals and state 1 to the boundary. Temporal changes in cell identity are highlighted by the highest estimated probabilities in each generation distribution

$$\begin{aligned}
&\Gamma_0(0,0,0,2) = 0.07,\\
&\Gamma_0(1,0,0,0) = 0.29,\\
&\Gamma_0(1,0,0,1) = 0.42,\\
&\Gamma_0(2,0,0,0) = 0.20.
\end{aligned}
\qquad
\begin{aligned}
&\Gamma_1(0,0,0,1) = 0.13,\\
&\Gamma_1(0,1,0,0) = 0.45,\\
&\Gamma_1(0,2,0,0) = 0.35.
\end{aligned}$$

$$\begin{aligned}
&\Gamma_2(0,0,0,1) = 0.13,\\
&\Gamma_2(0,0,0,2) = 0.10,\\
&\Gamma_2(0,0,1,0) = 0.18,\\
&\Gamma_2(0,0,1,1) = 0.28,\\
&\Gamma_2(0,0,2,0) = 0.31.
\end{aligned}
\qquad
\begin{aligned}
&\Gamma_3(0,0,0,1) = 0.14,\\
&\Gamma_3(0,0,0,2) = 0.09,\\
&\Gamma_3(0,0,1,1) = 0.11,\\
&\Gamma_3(0,1,0,1) = 0.35,\\
&\Gamma_3(0,2,0,0) = 0.17,\\
&\Gamma_3(1,0,0,0) = 0.05.
\end{aligned}$$

Recall that

$$\forall s \in \{0,1,2,3\}\,, \quad \Gamma_s\left(n_0, n_1, n_2, n_3\right),$$

denotes the probability of having jointly n_0, n_1, n_2 and n_3 children in state 0, 1, 2 and 3 considering a parent cell in state s. The reproduction and emergence of cell identities underlined by generation distributions are consistent with biological beliefs. State 3 is a hub for transitions from state 2 at 0h to other states at times superior to 44h. Transition from state 3 to state 0 corresponds to transition from early cells to late ones in sepals. Transition from state 3 to state 1 corresponds to emergence of boundary cells induced by sepal formation, it seems to be a passive phenomena more than an active one.

3.4.2 Discussions

Data limitation Regarding the biological interpretations drawn from the outputs of the HMOT models, we would like to stress that they have been partly limited by the number of available successive time points and data quality. Indeed, the time intervals between successive images were too large, thus a few divisions were not observed. For biological purposes these missing divisions were interpolated but this resulted into the presence of a large number of predicted cells without observed characteristics (almost 50%). In addition, the number of time points (5) also limited the investigation in details of cell division patterns. To solve this, Yassin Rehafi – former Ph.D. student of the team – used an enhanced version of the experimental protocol to acquire more time points (up to 15) and with a smaller time interval between successive acquisitions. In addition, the obtained raw images are of better quality. This will help to obtain more accurate segmentations and thus more reliable cell characteristics. Since some algorithms tend to systematically overestimate or underestimate the cell characteristics values, there are still some improvements to be done in the cell characteristic computation from raw or segmented images. This is in particular true for the curvature characteristics computed with an unadaptive algorithm, which could enhanced by an adaptive one (Tong and Tang, 2005).

Dependence hypothesis Let $\mathcal{M}_1$ be the inferred HMUOT model in which the number of children in each state are assumed to be independent

$$\forall s \in \mathcal{S}, \forall t \in \mathcal{T},\ P\left(\boldsymbol{N}_t = \boldsymbol{n}_t \,|\, S_t = s\right) = \prod_{i \in \mathcal{S}} P\left(N_{t,i} = n_{t,i} \,|\, S_t = s\right),$$

Comparing $\mathcal{M}_0$ to $\mathcal{M}_1$ enable to discuss the benefit of introducing dependencies in HMUOT models (see figure 3.5).

It is not surprising to that $\mathcal{M}_0$ has a higher log-likelihood than $\mathcal{M}_1$ since the state processes are nested and have theoretically respectively 55 and 36 independent parameters. Nevertheless, the estimated models have a number of non-zero parameters relatively close: 22 for $\mathcal{M}_0$ against 18 for $\mathcal{M}_1$. As a consequence, the BIC is higher for $\mathcal{M}_0$ (-18164) than for $\mathcal{M}_1$ (-18668). This fact indicates that there is a clear benefit of taking into account dependencies in this application.

According to the confusion table 3.3, the robustness of the restoration algorithm with respect to model misspecifications induced little changes concerning the most probable

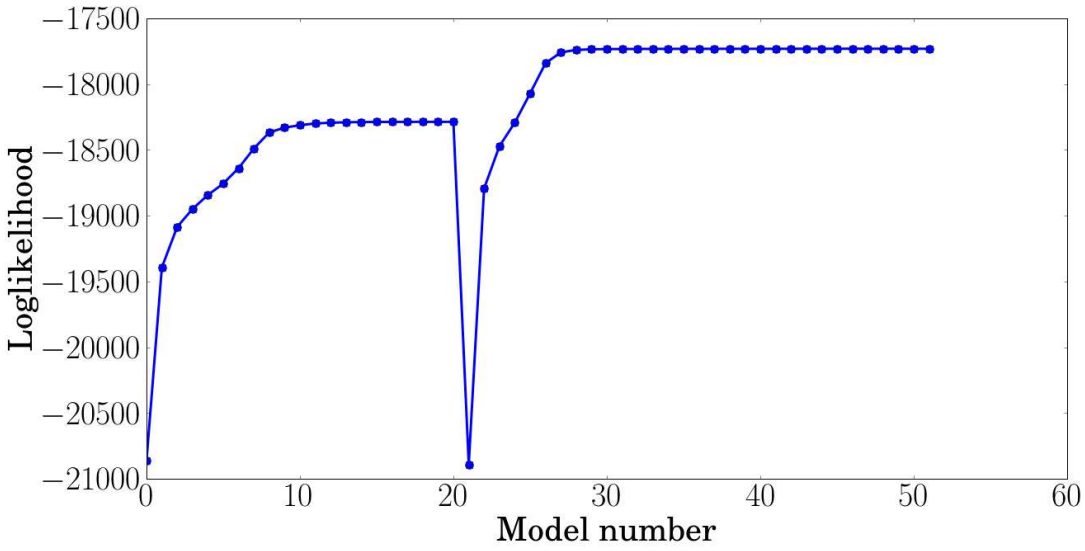


Figure 3.5 – Iterations of the Expectation-Maximization (EM) algorithm for the independent and dependent Markov unordered out-tree models. The successive log-likelihood improvements during the EM algorithm for the independent model are numbered from 0 *up to* 20*. The successive log-likelihood improvements during the EM algorithm for the dependent model are numbered from* 21 *up to* 56*. The maximum log-likelihood estimates reach* -18286.48 *for* $\mathcal{M}_1$ *and* -17730.57 *for* $\mathcal{M}_0$*.*

cell identities with models $\mathcal{M}_0$ and $\mathcal{M}_1$. Nevertheless, most of assignment mismatches concern the state 3. If the limited quality and amount of data did not allow to observe clear left-right models –in left-right models there are successions of transient states and final absorbing states such that states are partially ordered – the generation distributions for state 3 estimated in $\mathcal{M}_0$ emphasizes the main differences concerning left-right HMUOT models. If the number of children in each state are assumed to be:

Independent. In such models, a left-right model is induced by forbidden states transitions whatever the fact that division occurred or not.

Dependent. In such models, a left-right model can take into account the division phenomenon. In $\mathcal{M}_1$, the state 3 cannot change without dividing except for state 0 which corresponds to sepal cells aging. However, when a cell in state 3 divides, it can give mostly a cell in the same state and in states 1 or 2, which corresponds to the transient period when the boundary zone is set up. The dependent model enable to detect via its generation distributions patterns that are of marked interest for the biological application.

Link to Hidden Independent Markov Out-Tree (HIMOT) The HIMOT proposed by Crouse et al. (1998) have the same parametrization of standard hidden first-

$\mathcal{M}_1$	$\mathcal{M}_0$ states			
states	0	1	2	3
0	**197**	**18**	**0**	**98**
1	**0**	**533**	**20**	**69**
2	**0**	**5**	**358**	**24**
3	**28**	**15**	**19**	**280**

Table 3.3 – Confusion table regarding the most probable state tree for the model $\mathcal{M}_1$ (number of children in each state independent) against the model $\mathcal{M}_0$ (number of children in each state dependent). The matching between the restorations are high (more than 82%) since the restoration can be considered as robust relatively to model misspecifications (Durand et al., 2005).

order Markov chain models. This is the consequence of a strong conditional independence assumption within the state process where the child vertices are independent given the state of the parent vertex. Given this assumption the following transition distributions are obtained for MOOT models

$$\begin{aligned}\forall t \in \mathcal{T},\, \Pi_{x_t}^{(n_t)}\left(\bar{x}_{\mathrm{ch}(t)}\right) &= \prod_{\substack{x_c \in \bar{x}_{\mathrm{ch}(t)} \\ c \in ch(t)}} P\left(X_c = x_c \,|\, X_t = x_t, N_t = n_t\right) \\ &= \prod_{x \in \mathcal{X}} \left(\pi_{x_t}(x)\right)^{n_{t,x}},\end{aligned}$$

with

$$\forall x \in \mathcal{X},\, \sum_{x' \in \mathcal{X}} \pi_x(x') = 1.$$

Hence, for MUOT models

$$\forall t \in \mathcal{T},\, \Gamma_{x_t}\left(\bar{x}_{\mathrm{ch}(t)}\right) = \Gamma_{x_t}(n_t) \binom{n_t}{n_{t,0}, \cdots, n_{t,|\mathcal{X}|-1}} \prod_{x \in \mathcal{X}} \left(\pi_{x_t}(x)\right)^{n_{t,x}}.$$

This corresponds to parametric generation distributions with sum compound multinomial parametrization (see Johnson et al., 1997).

Parametric generation distributions We only considered semi-parametric HMUOT in this application since we were dealing with simple trees. For general trees, the combinatorics induced by the variable and high number of child vertices in each state induces a rapid inflation in the number of model parameters (see table 3.2). In such cases, since the data used to infer each generation distribution is of limited size, the inference of parametric HMUOT models is required in order to obtained reliable generation distributions. As it has been presented, this issue reduces to the inference of parametric discrete multivariate distributions (Johnson et al., 1997) and will be discussed in the next chapter.

References

D. Bacciu, A. Micheli, and A. Sperduti. Bottom-up generative modeling of tree-structured data. In K. Wong, B. Mendis, and A. Bouzerdoum, editors, *Neural Information Processing. Theory and Algorithms*, volume 6443 of *Lecture Notes in Computer Science*, pages 660–668. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17536-7. doi: 10.1007/978-3-642-17537-4_80. URL http://dx.doi.org/10.1007/978-3-642-17537-4_80. 69

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of Mathematical Statistics*, 41(1):164–171, 1970. 67

G. Claeskens and N. L. Hjort. *Model selection and model averaging*, volume 330. Cambridge University Press, 2008. 75

M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4): 886–902, 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668544. 67, 69, 72, 76, 84

P. Das, T. Ito, F. Wellmer, T. Vernoux, A. Dedieu, J. Traas, and E. M. Meyerowitz. Floral stem cell termination involves the direct regulation of agamous by perianthia. *Development*, 136(10):1605–1611, 2009. URL http://dev.biologists.org/content/136/10/1605.short. 79

J.-B. Durand, P. Goncalvès, and Y. Guédon. Computational methods for hidden Markov tree models–An application to wavelet trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1323262. 67, 69, 74, 76

J.-B. Durand, Y. Guédon, Y. Caraglio, and E. Costes. Analysis of the plant architecture via tree–structured statistical models: the hidden Markov tree models. *New Phytologist*, 166(3):813–825, 2005. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.2005.01405.x/full. 69, 80, 85

Y. Ephraim and N. Merhav. hidden Markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1003838. 67, 70, 72

T. E. Harris. *The theory of branching processes*. Courier Dover Publications, 2002. 71

N. Johnson, A. Kemp, and S. Kotz. *Univariate discrete distributions*. Wiley-Interscience, 1993. 70

N. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*. Wiley New York, 1997. 71, 85

S. Lauritzen. *Graphical models*, volume 17. Oxford University Press, 1996. 68

G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, Hoboken, NJ, 2007. 75

G. McLachlan and D. Peel. *Finite mixture models*. Wiley New York, 2004. 73

O. Ronen, J. Rohlicek, and M. Ostendorf. Parameter estimation of dependence tree models using the EM algorithm. *IEEE Transactions on Signal Processing Letters*, 2(8): 157–159, 1995. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=404132. 67, 69

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL http://dx.doi.org/10.1214/aos/1176344136. 80

D. R. Smyth, J. L. Bowman, and E. M. Meyerowitz. Early flower development in Arabidopsis. *The Plant Cell Online*, 2(8):755–767, 1990. URL http://www.plantcell.org/content/2/8/755.short. 79

W.-S. Tong and C.-K. Tang. Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3): 434–449, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1388268. 83

H. W. Watson and F. Galton. On the probability of the extinction of families. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 4:138–144, 1875. 71

G. C. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10474930. 77

Chapter 4

Inference of Mixed Acyclic Graphical Models (MAGMs) in Multi-Type Branching Processes (MTBPs)

Abstract We address the inference of discrete state-space models for tree-structured data. Our aim is to introduce parametric Multi-Type Branching Processes (MTBPs) that can be efficiently estimated on the basis of data of limited size. Each generation distribution within this macroscopic model is modeled by a Discrete Parametric Mixed Acyclic Graphical Model (DPMAGM). Special attention is first given to the case of Poisson Mixed Acyclic Graphical Model (PMAGM) in order to introduce constraints imposed in such DPMAGM. Then, the model is generalized to other distributions than Poisson. The algorithm presented for the estimation of each graphical model relies on a greedy algorithm.The proposed modeling approach is illustrated on plant datasets.

Keywords Branching process; discrete multivariate distribution; graph selection; mixed acyclic graphical model; multivariate count data; plant architecture; tree pattern

Contents

4.1 Introduction **89**
4.2 Definitions **90**
4.2.1 Multi-Type Branching Processes (MTBPs) 90
4.2.2 Poisson Mixed Acyclic Graphical Models (PMAGMs) 91
4.2.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) 95
4.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) inference **95**
4.3.1 Parameter inference 96
4.3.2 Structure inference 96
4.4 Application to Multi-Type Branching Processes (MTBPs): the case of mango tree asynchronisms **99**
4.5 Concluding remarks **102**
References **104**

4.1 Introduction

We consider discrete state-space stochastic processes indexed by a directed 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 Multi-Type Branching Processes (MTBPs).

In a practical setting of general tree 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. Inference of MTBPs mostly reduces to inference of their generation distributions, which are discrete multivariate distributions. In order to have interpretable results, we propose to focus on a family of discrete multivariate generation distributions that fulfills the following criteria:

1. Multivariate parametric distributions have to be used since the direct estimation of probability masses on the basis of multivariate counts is unreliable except for very large data sets.

2. These multivariate parametric distributions can have zero-inflated, right-skewed and natural number valued marginals, so that discretized multivariate Gaussian distributions are not appropriate.

3. These multivariate parametric distributions can easily be simulated and probability masses can easily be computed in order to investigate motifs induced by generation distributions and long-range patterns stemming from these generation distributions as trees develop.

4. Child states that tend to appear simultaneously or on the contrary to be incompatible can be identified.

To achieve this goal, we introduce parametric MTBPs incorporating probabilistic graphical models (Koller and Friedman, 2009) to represent each generation distributions. In this framework conditional independence relationships between number of children in each state can be easily represented. In particular, graph identification is one way to consider the above-mentioned criterion (4). Three kinds of graphical models are usual: undirected graphical models, Directed Acyclic Graphical Models (DAGMs) and Mixed Acyclic Graphical Models (MAGMs). Methods for graph identification were proposed for:

Undirected graphical models. Using log-linear models or a multivariate Gaussian distribution assumption, approaches based on a L_1 penalization (Lasso) were proposed (see Friedman et al. (2008) for the Gaussian graphical Lasso). To some extent, these approaches were extended to discrete multivariate distributions using Poisson regression models and more generally Generalized Linear Models (GLMs) (Yang et al., 2012).

DAGMs. Most methods rely on an exploration of the Directed Acyclic Graph (DAG) space using search heuristics (e.g. hill climbing, see Chickering (2002) for instance)

and consistent scores (e.g. Bayesian Information Criterion (BIC), see Yang and Chang (2002) for details). At each step of the greedy algorithm a list of graphs is proposed using graph edition (i.e. edges are removed, reversed or added), then the graph with highest score is selected. This procedure is repeated until the score reaches a local optimum (see for Koller and Friedman (2009) a review).

MAGMs. If MAGMs generalize both undirected graphical models and DAGMs, they have been considered less often in the literature. Proposed algorithms (Edwards, 2000; Ma et al., 2008; Drton and Perlman, 2008) mostly focused on tests of hypothesis and require an *a priori* the knowledge of the chain components. Moreover, most of them rely on Gaussian distributions or log-linear models that are not adapted to criteria 1 or 2.

We choose here to use discrete parametric MAGMs for generation distribution in MTBPs. The remainder of this chapter is organized as follows. MTBPs with Poisson Mixed Acyclic Graphical Models (PMAGMs) and their generalization to Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) used to model generation distributions are presented in section 4.2. A dedicated algorithm for inference of MAGMs is introduced in section 4.3. Then, the interest of such models in the case of MTBPs is illustrated in section 4.4 using the example of the mango tree asynchronism analysis. Finally, in section 4.5, work in progress and possible extensions of considered MAGMs to multivariate continuous or mixed distributions are discussed.

4.2 Definitions

Data of interest are categorical tree-indexed data $\bar{x} = (x_t)_{t\in\mathcal{T}}$ where $\mathcal{T} \subset \mathbb{N}$ is the set of vertices of a directed tree-graph $\tau = (\mathcal{T}, \mathcal{E})$ and $\mathcal{E} \subset \mathcal{T} \times \mathcal{T} \setminus \mathcal{R}$ the set of directed edges representing lineage relationships between vertices. $\mathcal{R}$ represent the set of roots and $\mathcal{L}$ the set of leaves of τ. Let $\mathrm{pa}(.)$ denote the parent, $\mathrm{ch}(.)$ the child set, $\mathrm{de}(.)$ the descendant set and $\mathrm{nd}(.)$ the non-descendant set of a vertex. These notations also apply to sets of vertices (see Koller and Friedman, 2009, for graph terminology). For any set $\mathcal{A} \subseteq \mathcal{T}$, let $\bar{x}_\mathcal{A}$ denote the subset of $\bar{x}$ obtained by only considering the vertices in $\mathcal{A}$,

$$\forall \mathcal{A} \subseteq \mathcal{T},\ \bar{x}_\mathcal{A} = (x_t)_{t\in\mathcal{A}}.$$

The notation $\bar{n} = (n_t)_{t\in\mathcal{T}}$ denotes the univariate tree-indexed data indexed by the same tree as $\bar{x}$ and corresponding to the number of children of each vertex,

$$\forall t \in \mathcal{T},\ n_t = |\mathrm{ch}(t)|.$$

4.2.1 Multi-Type Branching Processes (MTBPs)

We here assume that $\bar{x} = (x_t)_{t\in\mathcal{T}}$ and $\bar{n} = (n_t)_{t\in\mathcal{T}}$ are the outcomes of a random process. In the following, τ is considered *sensu stricto* as a tree and in a forest, trees are considered as independent and identically distributed. Let r denote the only root of τ. We here consider the simple case where:

- $\bar{x}$ is the realization of a $\mathcal{X}$-valued stochastic process $\bar{X} = (X_t)_{t\in\mathcal{T}}$ such that $\mathcal{X} \subset \mathbb{N}$ is called the state space and each value $x \in \mathcal{X}$ is called state. This process is therefore called the state process.

- $\bar{n}$ is the realization of a $\mathcal{N}$-valued stochastic process $\bar{N} = (N_t)_{t\in\mathcal{T}}$ with $\mathcal{N} \subset \mathbb{N}$. This process is called generation process.

These considerations raise the question of modeling the joint distribution

$$P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right). \tag{4.1}$$

MTBPs are parsimonious models relying on local dependence assumptions in $\bar{X}$ with respect to tree topology. More precisely, the following Markov property is considered

$$\forall t \in \mathcal{T},\ X_t \perp\!\!\!\perp N_{\mathrm{nd}(t)\setminus\{\mathrm{pa}(t)\}}, \bar{X}_{\mathrm{nd}(t)\setminus\{\mathrm{pa}(t)\}} \,\middle|\, X_{\mathrm{pa}(t)}\ .$$

This assumption that the state variable at vertex t is independent of its non-descendants variables given its parent state variable combined with the assumption that the generation process at vertex t is independent of its non-descendants variables given its state variable

$$\forall t \in \mathcal{T},\ N_t \perp\!\!\!\perp \bar{N}_{\mathrm{nd}(t)}, \bar{X}_{\mathrm{nd}(t)} \,|\, X_t.$$

Adding a permutation invariance property (see Haccou et al., 2005, for details) lead to parsimonious model, in which the distribution (4.1) is factorized as follows

$$P\left(\bar{X} = \bar{x}, \bar{N} = \bar{n}\right) \propto P\left(X_r = x_r\right) \prod_{t\in\mathcal{T}\setminus\{r\}} P\left(\boldsymbol{N}_t = \boldsymbol{n}_t \,|\, X_t = x_t\right), \tag{4.2}$$

where $\boldsymbol{N}_t$ is the discrete random vector of the number of children of vertex t in each state. Therefore, the outcomes to model are the realizations, $(\boldsymbol{n}_t)_{t\in\mathcal{T}}$, of the discrete random vector $\boldsymbol{N}_t$ for each vertex

$$\begin{aligned}
\forall t \in \mathcal{T},\ \boldsymbol{n}_t &= (n_{t,x})_{x\in\mathcal{X}} \\
&= \left(|\{c \in ch\left(t\right) | X_c = x\}|\right)_{x\in\mathcal{X}}, \\
n_t &= \sum_{x\in\mathcal{X}} n_{t,x}.
\end{aligned}$$

Under a homogeneity hypothesis, the considered MTBPs are thus specified by $|\mathcal{X}|$ discrete multivariate distributions – one by state – called generation distributions and an initial distribution for the root vertex that will not be considered hereafter.

4.2.2 Poisson Mixed Acyclic Graphical Models (PMAGMs)

We here propose to model these generation distributions by MAGMs. Since, in the following, we focus on a single generation distribution, the vertex indexing and parent state conditioning will be omitted in notations.

Parametrization A MAGM is a bipartite model composed of a graph $\mathcal{G}$ and a distribution P. The graph $\mathcal{G}$ is a Mixed Acyclic Graph (MAG) and P is said to satisfy the Factorization Chain property (FC) with respect to $\mathcal{G}$,

$$P\left(\boldsymbol{N}=\boldsymbol{n}\right)=\prod_{\mathcal{C}\in\mathcal{H}_{\mathcal{G}}}P\left(\boldsymbol{N}_{\mathcal{C}}=\boldsymbol{n}_{\mathcal{C}}\,\middle|\,\boldsymbol{N}_{\mathrm{pa}(\mathcal{C})}=\boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}\right), \tag{4.3}$$

where $\mathcal{H}_{\mathcal{G}}$ denotes the set of chain components of $\mathcal{G}$ induced by undirected edges and $\mathrm{pa}\left(.\right)$ the set of parents of a chain component induced by directed edges. Considering multivariate counts, we define a PMAGM as a MAGM where:

- For a chain component $\mathcal{C}$ that is a singleton and has no parent, $N_{\mathcal{C}}$ follows an univariate marginal Poisson distribution,

$$\forall\mathcal{C}\in\mathcal{H}_{\mathcal{G}},\ \{|\mathcal{C}|=1\}\wedge\{|\mathrm{pa}\left(\mathcal{C}\right)|=0\}\Rightarrow N_{\mathcal{C}}\sim\mathcal{P}\left(\theta_{\mathcal{C}}\right).$$

- For a chain component $\mathcal{C}$ that is a singleton and has at least one parent, $N_{\mathcal{C}}|\boldsymbol{N}_{\mathrm{pa}(\mathcal{C})}=\boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}$ follows a conditional Poisson distribution,

$$\forall\mathcal{C}\in\mathcal{H}_{\mathcal{G}},\ \{|\mathcal{C}|=1\}\wedge\{|\mathrm{pa}\left(\mathcal{C}\right)|>0\}\Rightarrow N_{\mathcal{C}}|\boldsymbol{N}_{\mathrm{pa}(\mathcal{C})}=\boldsymbol{n}_{\mathrm{pa}(c)}\sim\mathcal{P}\left(f_{\mathcal{C}}\left(\boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}\right)\right).$$

- For a chain component $\mathcal{C}$ that is not a singleton and has no parent, $\boldsymbol{N}_{\mathcal{C}}$ follows a multivariate marginal Poisson distribution,

$$\forall\mathcal{C}\in\mathcal{H}_{\mathcal{G}},\ \{|\mathcal{C}|>1\}\wedge\{|\mathrm{pa}\left(\mathcal{C}\right)|=0\}\Rightarrow\boldsymbol{N}_{\mathcal{C}}\sim\mathcal{P}_{|\mathcal{C}|}\left(\boldsymbol{\theta}_{\mathcal{C}}\right).$$

- For a chain component $\mathcal{C}$ that is not a singleton and has at least one parent, $\boldsymbol{N}_{\mathcal{C}}|\boldsymbol{N}_{\mathrm{pa}(\mathcal{C})}=\boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}$ follows a multivariate conditional Poisson distribution,

$$\forall\mathcal{C}\in\mathcal{H}_{\mathcal{G}},\ \{|\mathcal{C}|>1\}\wedge\{|\mathrm{pa}\left(\mathcal{C}\right)|>0\}\Rightarrow\boldsymbol{N}_{\mathcal{C}}|\boldsymbol{N}_{\mathrm{pa}(\mathcal{C})}=\boldsymbol{n}_{\mathrm{pa}(c)}\sim\mathcal{P}_{|\mathcal{C}|}\left(\boldsymbol{f}_{\mathcal{C}}\left(\boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}\right)\right).$$

We will here only present the basic multivariate marginal and conditional Poisson distributions. The reader can refer to Johnson et al. (1993, 1997), Karlis (2003) and Karlis and Meligkotsidou (2005) for further details. The derivation of the multivariate Poisson distribution considered here is the result of multivariate reduction (see Mardia (1970) for further examples). The idea is to start with some independent random variables and to create new ones by considering some functions of the original variables. Since each new variable is a function of the original ones, a dependence structure is imposed creating multivariate models. Here, for a chain component $\mathcal{C}\in\mathcal{H}_{\mathcal{G}}$, the multivariate marginal Poisson distributions (Karlis, 2003) are constructed considering $|\mathcal{C}|+1$ independent univariate marginal Poisson variables denoted $Y_0,\ldots,Y_{|\mathcal{C}|}$ and

$$\forall c\in\mathcal{C},\ N_c=Y_0+Y_{(c)+1},$$

where (c) denotes the rank of c in $\mathcal{C}$. Similarly, multivariate conditional Poisson distributions (Karlis and Meligkotsidou, 2005) are constructed considering $|\mathcal{C}|$ independent

Number of states	Number of parameters for the non-parametric case	Poisson worst case
2	**19**	**11**
3	**59**	**29**
4	**139**	**59**

Table 4.1 – Number of parameters in non-parametric and worst case Poisson multi-type branching processes as a function of the number of states given trees with $\mathcal{N} = \{0, 1, 2, 3\}$*. For binary trees, there is at worst the same number of parameters in both cases. Note that the number of parameters in the non-parametric models is also a function of the cardinality of* $\mathcal{N}$ *but it is not true for the Poisson case.*

univariate marginal Poisson variables and one common univariate conditional Poisson variable.

Under this parametrization, the number of parameters, noted $|\theta|$, is bounded in the worst case by:

$$\begin{aligned} |\theta| &\leq |\mathcal{X}| - 1 + |\mathcal{X}| \sum_{\mathcal{C} \in \mathcal{H}_{\mathcal{G}}} \left(|\mathcal{C}| + 1 + |\mathrm{pa}\left(\mathcal{C}\right)| \right) \\ &\leq |\mathcal{X}| - 1 + \frac{|\mathcal{X}|^2 \left(|\mathcal{X}| + 3 \right)}{2}, \end{aligned}$$

where the worst case is obtained by considering a complete DAG. In contrast to the number of parameters for non-parametric MTBPs,

$$|\mathcal{X}| - 1 + |\mathcal{X}| \left[\sum_{n \in \mathcal{N}} \left\{ \binom{|\mathcal{X}| + n - 1}{n} \right\} - 1 \right],$$

far more parsimonious models for generation distributions are obtained even for relatively low observed vertex out-degrees when considering PMAGMs (see table 4.1).

Complete chain components Usually, for each $\mathcal{C} \in \mathcal{H}_{\mathcal{G}}$, the conditional distributions $P\left(\boldsymbol{N}_{\mathcal{C}} = \boldsymbol{n}_{\mathcal{C}} \,\middle|\, \boldsymbol{N}_{\mathrm{pa}(\mathcal{C})} = \boldsymbol{n}_{\mathrm{pa}(\mathcal{C})}\right)$ are factorized as products of clique factors (Lauritzen, 1996). The issue of defining discrete parametric models for these distributions is related to the definition of parametric discrete undirected graphical models. In such undirected graphical models P satisfies the Factorization property (F)

$$P\left(\boldsymbol{N} = \boldsymbol{n}\right) = \prod_{\mathcal{C} \in \mathcal{H}_{\mathcal{G}}} \phi_{\mathcal{C}}\left(\boldsymbol{n}_{\mathcal{C}}\right), \tag{4.4}$$

with respect to the undirected graph $\mathcal{G}$.

The case of univariate and multivariate Poisson distributions lead to undirected graphical models where each chain component is complete. In order to obtain parametric undirected graphical models with sparse graphs, the recent framework of Yang et al.

(2012) is of marked interest. In such models, each variable follows a conditional distribution conditioned on its neighbors in the graph. The undirected edges of the graph are in fact considered as bi-directed edges. Yang et al. (2012, 2014) considered the GLMs framework (McCullagh and Nelder, 1989) to define these conditional distributions and in particular Poisson regressions (Allen and Liu, 2012). Nevertheless, if criteria (1), (2) and (4) raised in the introduction are verified, the third one is problematic for two reasons:

- In this framework, the partition function is not computationally tractable even for a small number of vertices. Moreover, the joint distribution may even not be defined (coefficients of the Poisson regression models must all be negative). As a consequence masses cannot be computed exactly and/or rapidly.

- Simulations are conducted via Monte-Carlo Markov Chain (MCMC) methods such as the Gibbs sampler (Gilks, 2005). This algorithms introduce important mixing time issue such as the minimum chain length in order to reach the stationary distribution and the sampling interval in order to have independent events.

For these reasons we therefore chose to consider MAGMs such that chain components are complete.

Chain component vertices with same parents In addition to the complete chain component constraint, vertices of a same chain component have the same parents in the defined PMAGMs. The direct consequence of these constraints is that the set of independences represented by the PMAGM class is the same as the one represented by Poisson Directed Acyclic Graphical Models (PDAGMs). The PDAGMs are defined as DAGMs such that P satisfies the Directed Factorization property (DF)

$$P\left(\boldsymbol{N}=\boldsymbol{n}\right)=\prod_{x\in\mathcal{X}}P\left(N_x=n_x\,\middle|\,\boldsymbol{N}_{\mathrm{pa}(x)}=\boldsymbol{n}_{\mathrm{pa}(x)}\right), \tag{4.5}$$

with respect to the directed acyclic graph $\mathcal{G}$ and such that:

- For a vertex v that has no parent, N_v follows an univariate marginal Poisson distribution.

- For a vertex v that has at least one parent, $N_v|\boldsymbol{N}_{\mathrm{pa}(v)}=\boldsymbol{n}_{\mathrm{pa}(v)}$ follows a conditional Poisson distribution.

Nevertheless, the particular interest of PMAGMs with respect to PDAGMs can be illustrated by two major examples:

The parametric catalog. Although multivariate Poisson distributions are highly related to univariate Poisson distributions, a multivariate Poisson distribution cannot be expressed as a succession of Poisson regressions. The distributions of PMAGMs are therefore more general than the PDAGMs ones.

Controlled variance. Let us consider a weakly connected component in a PDAGM such that covariances between variables are positive. For simulation studies, an undesirable effect is that if the first random variable – in term of topology – produced a relatively rare event (high values), the succession of exponential links for Poisson regressions tends to produce at the end a multivariate count with a huge count sum. In contrast, in a chain component with positive covariances, this effect is damped.

4.2.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs)

In the case of DPMAGMs, the objective is to relax the Poissonian hypothesis. As presented by Johnson et al. (1997), the spectrum of discrete multivariate parametric distributions is not so large and mostly rely on generalization of usual discrete univariate parametric distributions (Johnson et al., 1993):

- the binomial distribution is generalized into the multinomial distribution,
- the negative binomial distribution is generalized into the negative multinomial distribution,
- and the multivariate Poisson distribution has already been introduced.

If, as in the Poisson case, multinomial and negative multinomial distributions and/or regressions can easily be introduced in MAGM, the minimal Independence map (I-map) chain components of resulting DPMAGMs are subject to the two same constraints:

1. all chain components are complete,
2. all vertices of a chain component have the same parents.

4.3 Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) inference

In the following, we address inference of DPMAGM in two different cases:

- A graph is given. In such case, the Maximum Likelihood (ML) parameter inference of a DPMAGM for which the given graph is an I-map is presented.
- No graph is given. In such case, the inference of the graph is performed using a search heuristic and graph scoring via ML inference.

4.3.1 Parameter inference

Let us consider a graph $\mathcal{G}$ that fulfills constraints (1) and (2). In such case parameter inference reduces to multinomial, negative multinomial, multivariate Poisson marginal or conditional distributions for chain components of cardinality superior to 1 and binomial, negative binomial and Poisson marginal or conditional distributions otherwise. Such inferences are standard in statistics and the ML estimates have closed forms (Johnson et al., 1997) or can be estimated via iterative algorithms such as the Expectation-Maximization (EM) algorithm for multivariate Poisson marginal (Karlis, 2003) or conditional distributions (Karlis and Meligkotsidou, 2005).

In the case where a graph $\mathcal{G}$ that does not fulfill the imposed constraints is given as an I-map, it cannot be a minimal I-map. The solution is therefore to degrade the given graph – by removing undirected and/or directed edges – until reaching a graph, which fulfills constraints (1) and (2).

4.3.2 Structure inference

If the graph is not given, inference of graph structure and distribution parameters has to be performed. As in many discrete optimization problems, graph identification for DAGMs and MAGMs do not appear to admit tractable solutions. In such cases heuristic methods have to be considered, although they do not guarantee to find the optimal solution. We here consider an extension of the standard method proposed for DAGMs, the local search.

The local search method operates over a search space (set of graphs in our case). This search space can be represented as an undirected graph where:

The vertex set is the set of candidate solutions, each being associated with a score. In the case of DAGMs, a vertex $\mathcal{G}$ represents a DAG and the score, noted $\text{score}(\mathcal{G})$ corresponds to log-likelihood, BIC or Akaike Information Criterion (AIC) obtained after ML parameter estimation (see Yang and Chang (2002) for a review of consistent scores in this case).

The edge set is defined using search operators. In the case of DAGMs, these operators correspond to edit operations: adding, removing or reversing an edge (see Koller and Friedman (2009) for a review). As a consequence, the neighbor set of a graph $\mathcal{G}$ is defined as the DAG subspace such that there is only one edge that is added, removed or reversed in $\mathcal{G}$:

$$\forall \mathcal{G} \in \mathcal{D}_a(\mathcal{X}), \text{ne}(\mathcal{G}) = \left\{ \mathcal{G}' \in \mathcal{D}_a(\mathcal{X}) \;\middle|\; \exists! (u,v) \in \mathcal{P}(\mathcal{X}), \begin{array}{r} [\mathcal{E}' \cup (u,v) = \mathcal{E} \cup (v,u)] \\ \vee\, [\mathcal{E}' = \mathcal{E} \cup (u,v)] \\ \vee\, [\mathcal{E}' \cup (u,v) = \mathcal{E}] \end{array} \right\},$$

with $\mathcal{D}_a(\mathcal{X})$ the set of directed acyclic graphs with $\mathcal{X}$ as vertex set.

Given this search space and an initial candidate $\mathcal{G}^{(0)}$, the local search consists in iteratively selecting among the neighbors of $\mathcal{G}^{(t-1)}$ the candidate $\mathcal{G}^{(t)}$ with highest score,

$$\forall \mathcal{G}^{(0)} \in \mathcal{D}_a(\mathcal{X}), \forall t \in \mathbb{N}^*, \; \mathcal{G}^{(t)} = \underset{\mathcal{G} \in \text{ne}\left(\mathcal{G}^{(t-1)}\right)}{\arg\max} \{\text{score}(\mathcal{G})\},$$

until the score reaches a local optimum.

The efficiency of this heuristic relies on a connected state space and this space interconnectivity definition:

- If each candidate has few neighbors, then the search procedure has to consider only a few options at each iteration, which can be evaluated exhaustively. However, due to the small number of neighbors, the path to an optimal solution can be long and the probability of being stuck in a local optimum is high.
- If each candidate has many neighbors, the path to the optimal is shorter and the probability of being stuck in a local optimum is lower. Nevertheless, each step can be computationally intensive or even prohibitive.

Local search in Mixed Acyclic Graph (MAG) space The local search in the MAG search space combines edit operations in the DAG and the undirected search space (adding or removing undirected edges). To improve the search space connectivity the following additional edit operations can be considered:

Orientation. An undirected edge in a MAG is oriented in the two directions.

Disorientation. A directed edge in a MAG is disoriented.

The major drawback of this search space is the huge number of local optima when considering DPMAGM (see figure 4.1). Except for few simple cases (e.g. a graph without any edge), the neighborhood of a minimal I-map is composed of graphs that are not minimal I-maps. As a consequence, these graphs are degraded in order to correspond to constraints (1) and (2) for ML parameter inference. Most of these minimal I-maps are therefore local optima (considering the neighborhood), the local search in this search space is stuck rapidly in local optima and inferred DPMAGM will not be relevant.

Local search in Quotient Acyclic Graph (QAG) space An alternative is therefore to change the search space. This approach is inspired from the Greedy Equivalent Search (GES) defined by Chickering (2003). The GES is a local search for DAGM graph inference that does not consider the DAG search space but the Equivalent Directed Acyclic Graph (EDAG) search space. An EDAG is a Partially Directed Acyclic Graph (PDAG) representing the set of DAGs that are Separation equivalent (S-equivalent) (see Chickering (2002, 2003) for more details). Note that a PDAG is a graph containing both undirected edges and directed edges but that is not assimilated to a MAG since its separation properties are those of the S-equivalent DAGs it represents. For few models (mostly non-parametric and Gaussian ones), S-equivalent DAGs share the same score. The GES is therefore a local search that operates over the EDAG search space in order to limit score redundancy and local optima.

In our case, the idea is to define a search space that operates over the MAG that are possible minimal I-maps for DPMAGM. To this end, let us first define the Quotient Acyclic Graph (QAG). A QAG of a MAG $\mathcal{G}$ is pair $(\mathcal{G}_\Pi, \Pi)$ where:

- $\Pi = \mathcal{H}_\mathcal{G}$ is a quotienting,

$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{D}_a(\mathcal{X})\rvert$	$\lvert\mathcal{Q}_a(\mathcal{X})\rvert$	$\lvert\mathcal{M}_a(\mathcal{X})\rvert$
1	**1**	**1**	**1**
2	**3**	**4**	**4**
3	**25**	**34**	**50**
4	**543**	**715**	**1,688**
5	**29,281**	**35,381**	**142,624**
6	**3,781,503**	**4,258,357**	**28,903,216**
7	**1,138,779,265**	**1,222,487,933**	**13,663,125,680**
8	**783,702,329,343**	**816,625,721,787**	**14,762,428,500,992**

Table 4.2 – Number of Directed Acyclic Graphs (DAGs), Quotient Acyclic Graphs (QAGs) and Mixed Acyclic Graphs (MAGs) as function of the vertex number. $\lvert\mathcal{X}\rvert$ is the vertex number. The number of DAGs as function of the vertex number, noted $\lvert\mathcal{D}_a(\mathcal{X})\rvert$, has been calculated by Robinson (1973). The number of MAGs, noted $\lvert\mathcal{M}_a(\mathcal{X})\rvert$, has been calculated by Steinsky (2003). The number of QAG, noted $\lvert\mathcal{M}_a(\mathcal{X})\rvert$ is given by (4.6)

- $\mathcal{G}_\Pi$ is the quotient graph of $\mathcal{G}$ induced by the quotienting Π that is a DAG.

As a direct consequence of constraints (1) and (2) for a given $\mathcal{X}$ vertex set, there is a one-to-one mapping between the space of DPMAGM minimal I-maps and the QAG space, noted $\mathcal{Q}_a(\mathcal{X})$. Since:

- The Stirling number of second kind

$$\left\{ \begin{matrix} \lvert\mathcal{X}\rvert \\ x+1 \end{matrix} \right\},$$

  gives the number of ways of partitioning the vertex set into $x+1$ non-empty cliques.

- For each of these partitions, a QAG can be defined.

The number of QAG, noted $\lvert\mathcal{Q}_a(\mathcal{X})\rvert$ is given by

$$\lvert\mathcal{Q}_a(\mathcal{X})\rvert = \sum_{x=0}^{\lvert\mathcal{X}\rvert-1} \left\{ \begin{matrix} \lvert\mathcal{X}\rvert \\ x+1 \end{matrix} \right\} \lvert\mathcal{D}_a(\{0,\cdots,x\})\rvert \tag{4.6}$$

with $\lvert\mathcal{D}_a(\{0,\cdots,x\})\rvert$ the number of DAGs with vertex set $\{0,\cdots,x\}$. The QAG space is therefore far less large than the MAG space (see table 4.2). Moreover, if the sampling distribution is faithful to a DPMAGM, its minimal I-map is in the QAG space.

As a consequence, the local search is conducted in the QAG search space. Since QAGs are DAGs, the edit operations for DAGs can be used. Nevertheless, if only these operations are considered, the search space is not connected since quotients remain unchanged. We furthermore considered the following operations:

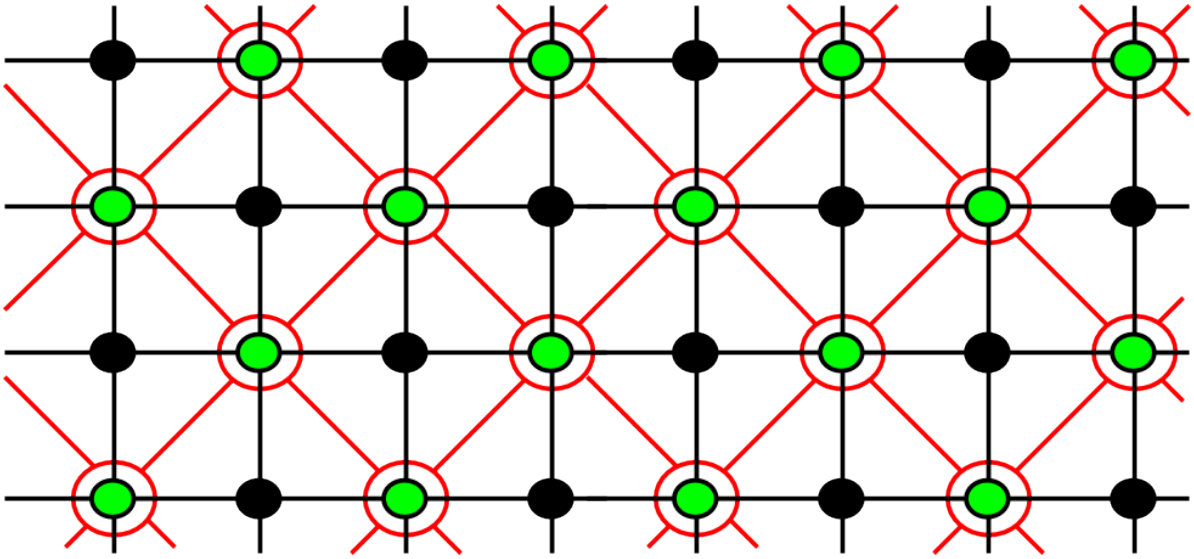


Figure 4.1 – Local search in Mixed Acyclic Graph (MAG) and Quotient Acyclic Graph (QAG) search spaces. The MAG search space vertices are represented by black or green disks and edges by black lines. The QAG search space vertices are represented by red circles and edges by red lines. The surrounding of a vertex in MAG search space by a vertex in the QAG search space represents the fact that they encode the same MAG. Since DPMAGMs are considered, the MAG search space is not relevant as most of minimal Independence-maps (I-maps) represented in green have non-minimal I-maps represented in black as neighbors. This induces the fact that most I-maps are local optima in the MAG search space. A contrario *since the QAG search space only considers the minimal I-maps of DPMAGM, the probability of being stuck in local optima is lower.*

Quotient merging. Two quotients $\mathcal{A}$ and $\mathcal{B}$ of Π are merged into $\mathcal{A} \cup \mathcal{B}$ if the closure of parent set of $\mathcal{A}$ is the parent set of $\mathcal{B}$. This results in the deletion of a vertex in the QAG.

Quotient splitting. A vertex c of a quotient $\mathcal{C} \in \Pi$ is used to form a new quotient that is a singleton and has parents $\{\mathcal{C} \cup \mathrm{pa}(\mathcal{C})\} \setminus \{c\}$. This results in the insertion of a vertex in the QAG.

4.4 Application to Multi-Type Branching Processes (MTBPs): the case of mango tree asynchronisms

As other tropical trees, mango tree is characterized by strong phenological asynchronisms, between and within trees, entailing patchiness (Chacko, 1986). Patchiness is characterized by clumps of either vegetative or reproductive Growth Units (GUs) within the canopy: while some parts of the tree canopy develop vegetative GUs, other parts may

remain in rest or produce inflorescences at the same time. These asynchronisms concern more or less large branching systems (Ramírez and Davenport, 2010). They entail various agronomic problems, such as the repeated use of pesticides to protect recurrent sensible phenological stages from pests, or a too extended period of fruit maturity, which may lead to difficulties to organize fruit harvesting.

At a given date, if all terminal GUs produced both vegetative and reproductive child GUs in the same proportions and synchronously (i.e. at the same burst dates), all branching systems would grow synchronously and would have the same distribution of fates. Patchiness results from mutual exclusions, at the local scale of sibling GUs, between some of their burst dates, and/or fates. Our objective was to identify and characterize such exclusions and to open new perspectives to eventually connect them to patchiness at the canopy scale.

Previous studies showed that the fate and burst date of a child GU are strongly affected by those of some ancestor GUs (Dambreville et al., 2013). This approach, based on regression models, only made possible to identify the effects of several factors (e.g. timing of development or fate of the parent GU, fruit load) on a single response variable, called GU feature (e.g. either the timing of development or the fate of a single child GU). This approach suffered from two main limitations:

- features of a GU cannot be predicted together in an obvious manner,
- a feature cannot be globally predicted for all child GUs if interactions exist between sibling GUs, additionally to those with the parent GU.

To characterize dependencies (in particular, exclusions) between child GUs through their architectural and phenological contexts, it is necessary that the notion of GU state combines (see figure 4.2 and 4.3):

The growth cycle delay. The growth cycle i of mango trees extends from July, 1^{st} of year $i-1$ until March, 1^{st} of year $i+1$. During this growth cycle, the vegetative phase corresponding to GU burst takes place between July, 1^{st} of year $i-1$ until June, 30^{th} of year i, then there is the flowering phase until September 30^{th} of year i, and finally there is the fructifying phase until March 1^{st} of year $i+1$. With respect to its parent GU burst date, a child GU can burst in the same (S) growth cycle or in the next (N) one.

The flush of burst of a GU. Each growth cycle of a mango tree is divided into three flushes. The early flush corresponds to the period where the vegetative phase of a cycle overlaps the flowering phase of the previous cycle. The intermediate flush corresponds to the period where the vegetative phase of a cycle overlaps the fructifying phase of the previous cycle. The late flush corresponds to the period where the vegetative phase does not overlap the previous or the next cycles. A GU can thus burst in the early (E), the intermediate (I) or the late (L) flush of the growth cycle.

The fate of a GU. The most important characteristic of a GU is its vegetative or reproductive character. Three cases have to be considered, the GU is vegetative

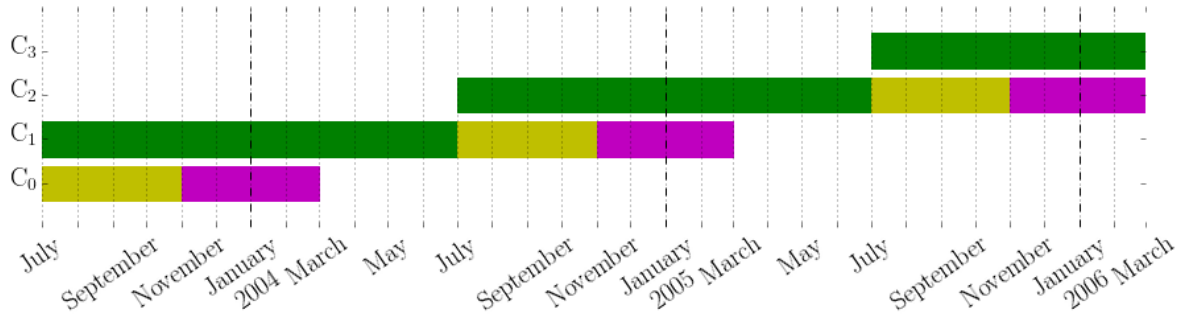


Figure 4.2 – Scheme of mango trees growth cycles. There are 3 *phases in the mango tree growth cycle: the vegetative phase (in green), the flowering phase (in yellow) and the fructifying phase (in magenta). Since there is a new growth cycle by year and that a growth cycle last for a year and a half, the vegetative phase is decomposed into* 3 *flushes (early, intermediate and late) corresponding to the overlap of this phase with the phases of the previous cycle.*

(V), reproductive with terminal flowering (T) or reproductive with lateral flowering (L).

Over the 18 states defined by the Cartesian product of the GU characteristics, only eleven states were observed[1]:

$$\mathcal{X} = \{\text{SEV, SLV, NEV, NIV, NLV, SIT, SLT, NIT, NLT, SIL, NIL}\}$$

Eleven DPMAGMs or PMAGMs were thus identified, each one associated with one parent GU state. We focus on graphs presented in figure 4.4 associated with the parent state SIT. Considering the QAG of the PMAGM estimated, 5 quotients are identified. Note that since we are considering PMAGM, covariances are positive in each quotient. Except for states NLV, NLT and NEV for which the edges correspond to positive regression coefficients, all the other edges are associated with negative ones. In the following we therefore group these former states. As a consequence, there are 3 exclusive strategies:

NIL & NIT children. In this configuration only flowering children produced in the intermediate flush of the next cycle are found.

NIV & SLT children. Note that SLT children are quite rare. In this group only NIV children are of relative importance. Such vegetative children are produced in the same period as in the previous configuration.

NEV, NLV & NLT children. In this configuration children can be both flowering or vegetative and are spread on periods (the early and late flushes in the next cycle) not represented by the two previous configurations.

[1]states are defined by concatenation of period, flush and fates abbreviations instead of positive integers for clarity purposes

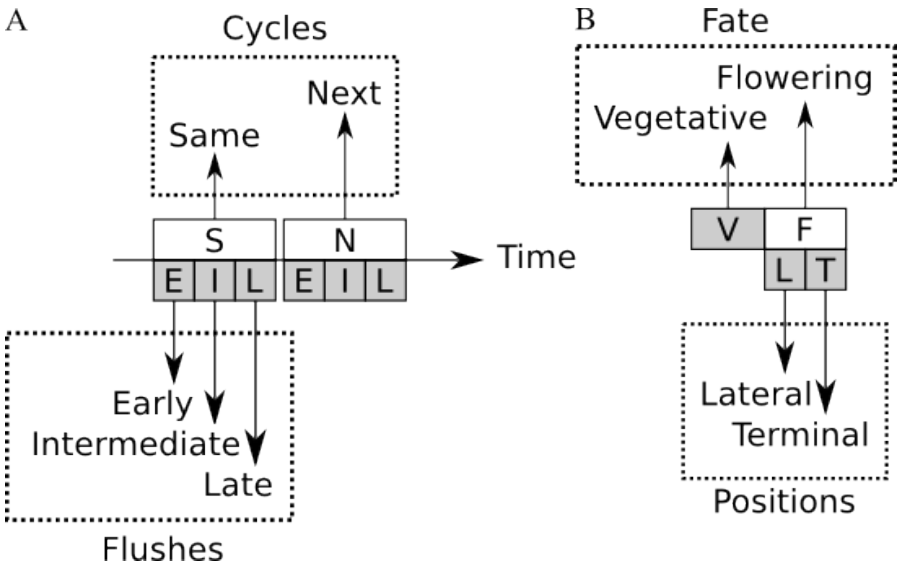


Figure 4.3 – States of Growth Units (GUs) in mango trees. (A) Temporal components of states. States of GUs are defined combining two temporal characteristics. A relative one focusing on the fact that a GU can burst in the same cycle as its parent GU or in the following one. An absolute one focusing on the fact that given the flush, the competitions for GU resource allocation is not the same. During the early flush new GUs are in competition with flowering GUs. In the intermediate flush new GUs are in competition with developing fruits. In the late flush new GUs are only in competition with themselves. (B) Fate components of states. The most important characteristic of a GU is its vegetative or reproductive character. Note that in the case of flowering, the position of the flower is important in an agronomic point-of-view since available space for child GUs is not the same.

These three strategies are thus highly consistent with the patchiness set-up. The first two are contrasted regarding the fate of the children (vegetative against reproductive) but not on the period of burst. The last one is contrasted to former ones in term of periods (early or late against intermediate flushes in the same cycle) but no particular fate is represented. Given SIT parent GUs, our results highlights mutual exclusions between some of their burst periods or fates. These results illustrate the ability of the parametric MTBPs to identify in which contexts a given parent GU can or cannot have child GUs at different flushes or with different fates, which can be interpreted as the origin of asynchronisms.

4.5 Concluding remarks

Discrete Parametric Mixed Acyclic Graphical Models (DPMAGMs) Let consider the given parametric catalog of discrete multivariate distributions:

- multinomial and derived distributions,

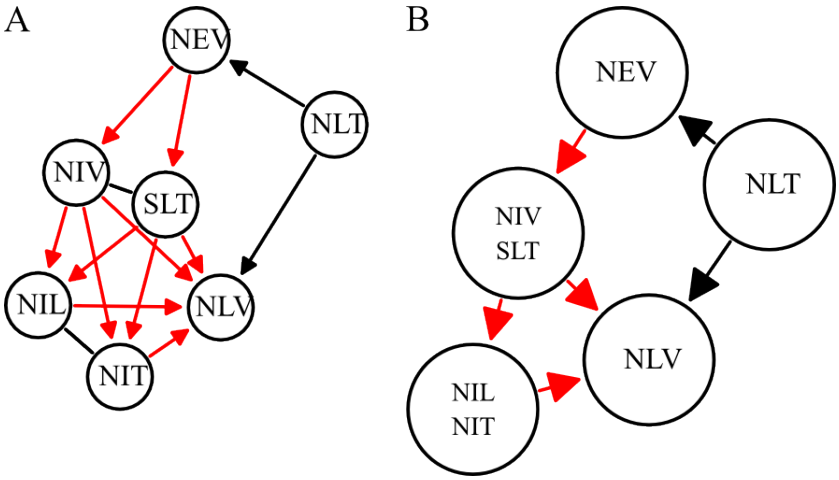


Figure 4.4 – (A) Mixed Acyclic Graph (MAG) and (B) Quotient Acyclic Graph (QAG) of a generation distribution. The parent state of the generation distribution is SIT, a flowering GU burst in the same cycle as its parent during the intermediate flush. It is there not surprising that children in states SEV, SLV, SIT, and SIL are not observed (and therefore not represented) since these states are temporally or biologically incompatible (the succession of three GUs burst in the same cycle is highly improbable). The QAG is very convenient as it represents the MAG at a coarser scale, which is less complicated. Black edges are associated with positive covariances and red edges with negative covariances.

- negative multinomial distribution,
- multivariate Poisson distribution.

All these distributions impose the same sign on covariances between random variables. If it can be seen as a flaw when they are directly used to model multivariate count data, when plugged in DPMAGMs it enables a two-stage interpretation of the model.

As illustrated in the results, the same sign of covariances in quotients help to interpret within quotients relations and to bring to the foreground biological meaning of these quotients. Then, the use of the QAG allows to investigate efficiently relations between quotients by representing the MAG at a coarser scale.

Gaussian Mixed Acyclic Graphical Models (GMAGMs) In this chapter only MAGM for multivariate count data were considered. In case of collections of real-valued outcomes, Gaussian Mixed Acylic Graphical Models (GMAGMs) are of marked interest. Since GMAGMs are not constrained, contrarily to DPMAGMs, they offer a relevant alternative to Gaussian undirected graphical models or Gaussian Directed Acylic Graphical Models (GDAGMs).

If for graph identification of GMAGMs the local search in the MAG space could thus be used, it is important to remark that the local search space in the QAG could be more relevant. In fact, using Lasso estimators such as the Gaussian graphical Lasso

(Friedman et al., 2008), the optimal MAG, which is a partial graph of a MAG encoded by a QAG, is selected easily and the estimation is consistent. Therefore the local search in QAG combined with the Lasso estimator is an interesting alternative to the local search in MAG space. Current work consists in implementing such estimators in order to perform a sensibility analysis of our heuristic and to compare it to the local search space in MAG space.

A study of emerging patterns Considering the motifs highlighted by generation distributions is only a local point of view on asynchronisms. This local point of view can be turned into a more integrated view by predicting, using the MTBP model, the total number of descendant GUs at each flush and each fate using limit theorems (Yang, 2003) or simulation approximations. Note that this patchiness can be viewed also as a long-range pattern present in trees and it will be therefore relevant to consider long-range dependency models for tree-indexed as discussed in the next chapter.

References

G. I. Allen and Z. Liu. A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1–6. IEEE, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6392619. 94

E. Chacko. Physiology of vegetative and reproductive growth in mango (*Mangifera indica* L.) trees. In *Proceedings of the First Australian Mango Research Workshop*, volume 1, pages 54–70. CSIRO Australia, Melbourne, 1986. 99

D. Chickering. Learning equivalence classes of Bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002. URL http://dl.acm.org/citation.cfm?id=944800. 89, 97

D. Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003. URL http://dl.acm.org/citation.cfm?id=944933. 97

A. Dambreville, P. Fernique, C. Pradal, P.-E. Lauri, F. Normand, Y. Guédon, and J.-B. Durand. Deciphering mango tree asynchronisms using Markov tree and probabilistic graphical models. In R. Sievänen, E. Nikinmaa, C. Godin, A. Lintunen, and P. Nygren, editors, *FSPM2013 - 7th International Workshop on Functional-Structural Plant Models*, pages 210–212, Saariselkä, Finlande, 2013. URL http://hal.inria.fr/hal-00847614. ISBN 978-951-651-408-9. 100

M. Drton and M. D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008. URL http://www.sciencedirect.com/science/article/pii/S0378375807002303. 90

D. Edwards. *Introduction to graphical modelling*. Springer, 2000. 90

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3):432–441, 2008. URL http://biostatistics.oxfordjournals.org/content/9/3/432.short. 89, 104

W. R. Gilks. *Markov Chain Monte Carlo*. John Wiley & Sons, Ltd, 2005. doi: 10.1002/0470011815.b2a14021. URL http://dx.doi.org/10.1002/0470011815.b2a14021. 94

P. Haccou, P. Jagers, and V. A. Vatutin. *Branching processes: variation, growth, and extinction of populations*. Cambridge University Press, 2005. 91

N. Johnson, A. Kemp, and S. Kotz. *Univariate discrete distributions*. Wiley-Interscience, 1993. 92, 95

N. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*. Wiley New York, 1997. 92, 95, 96

D. Karlis. An EM algorithm for multivariate Poisson distribution and related models. *Journal of Applied Statistics*, 30(1):63–77, 2003. URL http://www.tandfonline.com/doi/abs/10.1080/0266476022000018510. 92, 96

D. Karlis and L. Meligkotsidou. Multivariate Poisson regression with covariance structure. *Statistics and Computing*, 15(4):255–265, 2005. URL http://link.springer.com/article/10.1007/s11222-005-4069-4. 92, 96

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 89, 90, 96

S. Lauritzen. *Graphical models*, volume 17. Oxford University Press, 1996. 93

Z. Ma, X. Xie, and Z. Geng. Structural learning of chain graphs via decomposition. *Journal of machine learning research: JMLR*, 9:2847, 2008. 90

K. V. Mardia. *Families of bivariate distributions*, volume 27. Hafner Publishing Co. Griffin London, 1970. 92

P. McCullagh and J. Nelder. *Generalized linear models. Monographs on Statistics and Applied Probability 37*. Chapman & Hall, London, 1989. 94

F. Ramírez and T. L. Davenport. Mango (*Mangifera indica* L.) flowering physiology. *Scientia Horticulturae*, 126(2):65–72, 2010. URL http://www.sciencedirect.com/science/article/pii/S0304423810002992. 100

R. Robinson. Counting labeled acyclic digraphs. In *Combinatorial mathematics V*, pages 28–43. Springer, 1973. 98

B. Steinsky. Enumeration of labelled chain graphs and labelled essential directed acyclic graphs. *Discrete Mathematics*, 270(1):267–278, 2003. 98

E. Yang, P. Ravikumar, G. Allen, and Z. Liu. Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems 25*, pages 1367–1375, 2012. 89, 93, 94

E. Yang, Y. Baker, P. Ravikumar, G. Allen, and Z. Liu. Mixed graphical models via exponential families. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 1042–1050, 2014. 94

S. Yang and K.-C. Chang. Comparison of score metrics for Bayesian network learning. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 32(3):419–428, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1046072. 90, 96

W. Yang. Some limit properties for Markov chains indexed by a homogeneous tree. *Statistics & Probability Letters*, 65(3):241–250, 2003. URL http://www.sciencedirect.com/science/article/pii/S0167715203002608. 104

Chapter 5

Quantification of plant patchiness via tree-structured statistical models: a tree-segmentation/clustering approach

Abstract The classical multiple change-point models for path-indexed data are transposed to tree-indexed data. The objective of multiple change-point models is to partition a heterogeneous tree-indexed data into homogeneous subtree-indexed data of consequent sizes. Since optimal algorithms of multiple change-point models for sequences cannot be transposed to trees, we propose here an efficient heuristic for tree segmentation. The segmented subtrees are grouped in a post-processing phase, since the occurrence of similar disjoint patches in the canopy is observed. Application of such models is illustrated on mango trees where subtrees are assimilated to plant patches and clusters of patches to patch types (e.g. vegetative, flowering or resting patch).

Keywords Change-point model; plant architecture; mango tree; tree clustering; tree pattern; tree segmentation

Contents

5.1 Introduction . . . 109
5.2 Material and methods . . . 110
5.2.1 Tree-structured representation of plants . . . 110
5.2.2 Modeling plant patchiness with tree segmentation/clustering models 111
5.2.3 Plant material . . . 113
5.3 Results . . . 115
5.3.1 Tree segmentation . . . 115
5.3.2 Subtree clustering . . . 116
5.3.3 Cultivar comparisons . . . 118
5.4 Discussion . . . 118
References . . . 121

5.1 Introduction

As other tropical trees, mango tree is characterized by strong phenological asynchronisms, between and within trees, entailing patchiness (Chacko, 1986). Patchiness is characterized by clumps of either vegetative or reproductive Growth Units (GUs) within the canopy: while some parts of the tree canopy develop vegetative GUs, other parts may remain in rest or produce inflorescences at the same time. These asynchronisms concern more or less large branching systems (Ramírez and Davenport, 2010). They entail various agronomic problems, such as the repeated use of pesticides to protect recurrent sensible phenological stages from pests, or a too extended period of fruit maturity, which may lead to difficulties to organize fruit harvesting. The objective here is to define a statistical methodology in order to identify and quantify such patchiness patterns. This approach is particularly interesting since it could enable the quantification of this phenomenon, and more generally, allow to highlight patchiness patterns for species where such patterns are not directly apparent in the data.

Tree-indexed data are used as plant architecture representation and it is assumed that plant patches can be assimilated to a partition of tree-indexed data into subtrees. It is therefore assumed that there are subtrees within which the botanical entity characteristics follow the same or nearly the same distribution and between which botanical entity characteristics have different distributions. The detection of such subtrees can be thus stated as tree-indexed data segmentation. Although patchiness is a spatio-temporal phenomenon, we focus here on its spatial aspect on given trees observed at given dates. Such a point of view introduces a lot of missing values in tree-indexed data since over these periods mostly vertices corresponding to the canopy (i.e. leaf of trees) are observed. Classical statistical models for tree-indexed data (Crouse et al., 1998; Durand et al., 2004, 2005) based on Markovian hypotheses are no more relevant since internal vertices are not observed. The chosen strategy is the search for abrupt changes in the proportions of GU types within the tree. This is the analog of sequence segmentation problems (Hupé et al., 2004; Olshen et al., 2004; Picard et al., 2005) carried on trees. It is worth noting that exact methods for determining the most probable segmentation of a sequence cannot be transposed to tree-structured data. We here therefore propose a greedy algorithm for segmenting trees. As underlined by Picard et al. (2007), the output of the segmentation procedure is a partition of trees considering that each element of this partition is different from each others while two non-adjacent subtrees can be very similar. We therefore propose a two-stage tree segmentation/clustering algorithm based on the previous segmentation procedure combined with a mixture model in order to identify similar subtrees.

The remainder of this chapter is organized as follows. Following the presentation of tree-structured representation of plants, the segmentation/clustering models are developed and practical aspects of the application of these models to botanical data are addressed in section 5.2. The contribution of these segmentation/clustering models for tree-indexed data in plant architecture is then illustrated in section 5.3 through the patchiness application. Finally, efficiency and technical difficulties concerning these models are discussed in section 5.4.

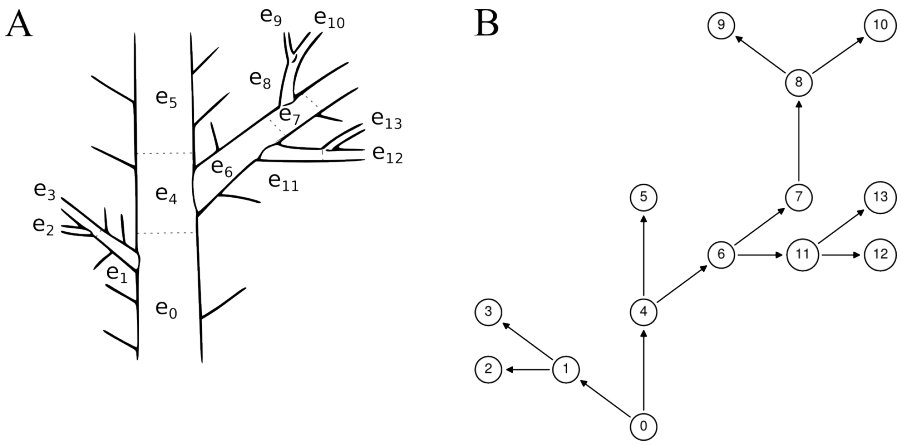


Figure 5.1 – Tree-indexed data extraction from plants (Durand et al., 2005). (A) A plant is observed at the GU scale where each GU is denoted by e_v with $v \in [\![0, 14[\![$. (B) Tree-graph representation of the same plant is drawn: each GU e_v is represented by a vertex v.

5.2 Material and methods

5.2.1 Tree-structured representation of plants

As discussed by Godin and Caraglio (1998), plant topology can be described formally through Multiscale Tree Graphs (MTGs). In a MTG, each vertex corresponds to a botanical entity at a given scale (e.g. metamer, GU) and each edge represent the physical connections between two botanical entities and each scale to a more-or-less macroscopic viewpoint on the plant (see chapter 2 for more details). Considering the methodology presented by Durand et al. (2005, see figure 5.1) for Hidden Markov Tree (HMT) models, a plant can also be represented by a tree-graph corresponding to a single scale of a MTG for statistical analysis purposes.

Data of interest are thus univariate tree-indexed data $\bar{x} = (x_t)_{t \in \mathcal{T}}$ – or more generally multivariate tree-indexed data noted $\bar{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t \in \mathcal{T}}$ – where $\mathcal{T} \subset \mathbb{N}$ is the set of vertices of a directed tree-graph $\tau = (\mathcal{T}, \mathcal{E})$ and $\mathcal{E} \subset \mathcal{T} \times \mathcal{T} \setminus \mathcal{R}$ the set of directed edges representing lineage relationships between vertices. $\mathcal{R}$ represent the set of roots and $\mathcal{L}$ the set of leaves of τ. Until further notice, we consider that τ is *sensu stricto* a tree and the only root of τ is denoted by r. Let $\mathrm{pa}(.)$ denote the parent, $\mathrm{ch}(.)$ the child set, $\mathrm{de}(.)$ the descendant set and $\mathrm{nd}(.)$ the non-descendant set of a vertex. These notations also apply to set of vertices (see Lauritzen, 1996, for graph terminology). Capitalized versions indicate the closure of the corresponding notation,

$$\forall t \in \mathcal{T},\ \mathrm{De}(t) = \mathrm{de}(t) \cup \{t\}.$$

For any set $\mathcal{A} \subseteq \mathcal{T}$, $\bar{x}_{\mathcal{A}}$ denote the subset of $\bar{x}$ obtained by only considering the vertices in $\mathcal{A}$,

$$\forall \mathcal{A} \subseteq \mathcal{T},\ \bar{x}_{\mathcal{A}} = (x_t)_{t \in \mathcal{A}}.$$

and $\tau_{\mathcal{A}}$ the subtree induced by $\mathcal{A}$. The in-degree of a vertex t in a tree, τ is denoted by $\deg_{\tau}^{-}(t)$. This in-degree is equal to 0 if the vertex is a root or 1 otherwise.

5.2.2 Modeling plant patchiness with tree segmentation/clustering models

To simplify notations we will consider in the following the case where $\bar{x}$ is the realization of a $\mathcal{X}$-valued stochastic process $\bar{X} = (X_t)_{t\in\mathcal{T}}$ such that $\mathcal{X} \subset \mathbb{N}$ is called the observation space.

Contrarily to Picard et al. (2007) who proposed segmentation/clustering models for sequences where the segmentation and the clustering were performed in a single stage we here propose a two-stage approach. In a first stage, each tree is quotiented into homogeneous subtrees considering tree segmentation models. In a second stage, a mixture model is used to group these homogeneous subtrees into clusters with similar biological characteristics.

Segmentation models A segmentation model is defined by a vertex quotienting, noted Π, such that each quotient induces a *sensu stricto* tree (any path between two vertices of one quotient is composed of vertices in the same quotient). Given these quotients, vertices in the same quotient are supposed to be independent and identically distributed. The parametrization of a segmentation model is therefore defined by these quotients and completed by one observation distribution for each quotient. As a consequence of these assumptions the log-likelihood, $\mathcal{L}(\bar{x}; \Pi, \theta_{\Pi})$ of the model decomposes as follows:

$$\mathcal{L}(\bar{x}; \Pi, \theta_{\Pi}) = \sum_{\pi\in\Pi} \sum_{v\in\pi} \log f_{\pi}(x_v),$$

where $f_{\pi}(\cdot)$ denotes the observation distribution of the quotient $\pi \in \Pi$ and θ_{Π} the set of parameters of these observation distributions.

The quotients in Π can also be identified by the set of change points, noted $\mathcal{P}$. Each of them corresponds to the root of the subtree induced by the considered quotient

$$\forall \Pi \in \mathfrak{P}(\mathcal{T}),\ \mathcal{P} = \left\{ t \in \mathcal{T} \,\middle|\, \exists \pi \in \Pi,\ [t \in \pi] \wedge \left[\deg_{\tau_\pi}^{-}(t) = 0\right] \right\},$$

where $\mathfrak{P}(\cdot)$ denotes the powerset of a set. The function $\nu(\cdot)$ denotes the function that returns the quotienting associated to a set of change points:

$$\begin{array}{rccc} \nu & : & \mathfrak{P}(\mathcal{T}) & \to & \mathfrak{P}(\mathcal{T}) \\ & & \mathcal{P} & \mapsto & \Pi \end{array}.$$

Inference of quotients In our practical case, given a quotienting Π, the inference of observation distributions is a simple Maximum Likelihood (ML) inference within each quotient. A major issue is, given a number K of quotients, to find the quotienting that maximizes the log-likelihood. Exact methods for determining the most probable segmentation of a path-indexed data cannot be transposed to tree-indexed data. We

therefore propose an heuristic approach in order to find a local optimal solution (see Hawkins (1976) for a similar approach on path-indexed data).

Let $\mathcal{P}^{(k)}$ denotes the change points set associated to $k+1$ quotients and corresponding to a local optimum of the log-likelihood. By definition, $\mathcal{P}^{(0)}$ is the change points set inducing one quotient and therefore only contains the root of the tree,

$$\mathcal{P}^{(0)} = \{r\}\,.$$

Finding the change points set $\mathcal{P}^{(1)}$, which maximizes the log-likelihood of the segmentation model with two quotients, can easily be done by testing successively all non-root vertices as change points

$$\mathcal{P}^{(1)} = P^{(0)} \cup \left\{ \arg\max_{t \in \mathcal{T}} \left\{ \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}^{(0)} \cup \{t\}\right), \theta_{\nu\left(\mathcal{P}^{(0)} \cup \{t\}\right)}\right)\right\}\right\}.$$

The optimal segmentation of a tree into 2 subtrees is therefore easily found. The principle of the heuristic presented in algorithm 4 is to use this principle to iteratively build the quotienting. Note that in order to reduce the probability of being stuck in local optima, at each step, if a new change point has been found, the removal of change points is considered until no more removal increases the log-likelihood.

Selection of the number of quotients If the number of quotients is unknown it has to be selected. Since the purpose of the segmentation is to reveal plant patches, the estimation of the number of quotient remains central. This problem can be handled in the more general context of model selection, as for the path-indexed data cases, using statistical criteria adapted to the case of segmentation models (Zhang and Siegmund, 2007; Rigaill et al., 2012) or slope heuristics (Lebarbier, 2005; Baudry et al., 2012).

In our practical context of categorical observations, penalized-likelihood criteria with fixed penalties select over-parametrized models and are not adapted. We therefore considered the data-driven slope heuristic method implemented by Baudry et al. (2012). Since this method requires the computation of over-parametrized models, we thus considered the computation of change-points sequences up to 20 change points.

Tree clustering models The segmentation models enable the detection of subtrees such that the observations do not change substantially within each subtree but change markedly between two adjacent subtrees. Nevertheless, the occurrence of similar non-adjacent subtrees in the tree is an important feature. It is therefore assumed that:

- There are a finite and small number of these different types of quotients and that all vertices in a quotient are of the same type.
- Vertices in the same quotient are independent and identically distributed given the type of the quotient.

The Expectation-Maximization (EM) and the Maximum A Posteriori (MAP) assignment of quotients of standard mixture models (McLachlan and Peel, 2004) under the constraint that vertices belonging to a given quotient are assigned to the same component were therefore applied in this context to group similar patches.

Algorithm 4 Computing a sequence of change points set

Require: $\bar{x}$, $\mathcal{T}$, the tree-indexed data

function TreeSegmentation(K)
  $\left(\mathcal{P}^{(k)}\right)_{k\in\{1,K-1\}} \leftarrow (\emptyset)_{k\in\{1,K-1\}}$ ▷ change-points set initialization
  $\left(\mathcal{L}^{(k)}\right)_{k\in\{1,K-1\}} \leftarrow (-\infty)_{k\in\{1,K-1\}}$ ▷ change-points set score initialization
  $k \leftarrow 0$ ▷ Step
  $\mathcal{P}^{(0)} \leftarrow \{r\}$ ▷ The root is the optimal first change point
  while $k < K$ **do**
    $\mathcal{P} \leftarrow \mathcal{P}^{(k)} \cup \left\{ \underset{t\in\mathcal{T}}{\arg\max} \left\{ \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}^{(k)} \cup \{t\}\right), \theta_{\nu\left(\mathcal{P}^{(k)}\cup\{t\}\right)}\right)\right\}\right\}$
    if $\mathcal{L}^{(k+1)} \leq \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}\right), \theta_{\nu(\mathcal{P})}\right)$ **then**
      $k \leftarrow k+1$
      $\mathcal{P}^{(k)} \leftarrow \mathcal{P}$ ▷ Add a change point
      $\mathcal{L}^{(k)} \leftarrow \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}^{(k)}\right), \theta_{\nu\left(\mathcal{P}^{(k)}\right)}\right)$ ▷ Update the score
      $\mathcal{P} \leftarrow \mathcal{P}^{(k)} \setminus \left\{ \underset{p\in\mathcal{P}^{(k)}\setminus\{r\}}{\arg\max} \left\{ \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}^{(k)} \setminus \{p\}\right), \theta_{\nu\left(\mathcal{P}^{(k)}\setminus\{p\}\right)}\right)\right\}\right\}$
      while $\mathcal{L}^{(k-1)} < \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}\right), \theta_{\nu(\mathcal{P})}\right)$ **do**
        $k \leftarrow k-1$
        $\mathcal{P}^{(k)} \leftarrow \mathcal{P}$ ▷ Remove a change point
        $\mathcal{L}^{(k)} \leftarrow \mathcal{L}\left(\bar{x}; \nu\left(\mathcal{P}^{(k)}\right), \theta_{\nu\left(\mathcal{P}^{(k)}\right)}\right)$ ▷ Update the score
    else
      $k \leftarrow k+1$
  return $\left(\mathcal{P}^{(k)}, \mathcal{L}^{(k)}\right)_{k\in\{0,\cdots,K\}}$

5.2.3 Plant material

Experimental design The experimental orchard was located at the Cirad[1] research station in Saint-Pierre, Réunion Island. For each of the following cultivars, 5 mango trees were described at the GU scale (Dambreville et al., 2013):

- Cogshall,
- Jose,
- Kensington Pride,
- Irwin,
- Kent,
- Nam Doc Mai,

[1]French Agricultural Research Center for International Development

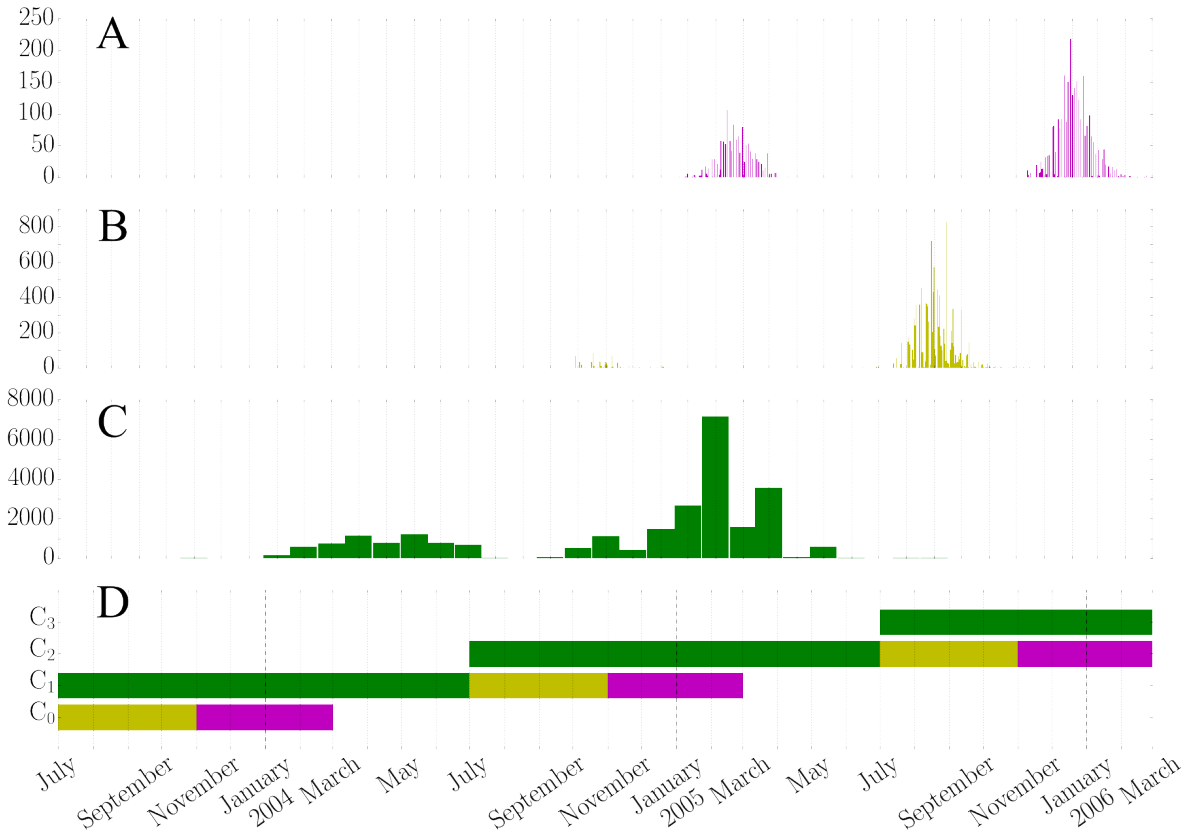


Figure 5.2 – Mango tree growth cycles. (A) Number of new fructifying GUs by day. (B) Number of new flowering GUs by day. (C) Number of new GUs by month. (D) Scheme of mango tree growth cycles. There are 3 phases in the mango tree growth cycle: the vegetative phase (in green), the flowering phase (in yellow) and the fructifying phase (in magenta). Since there is a new growth cycle each year and that a growth cycle lasts for a year and a half, the vegetative phase is decomposed into 3 flushes (early, intermediate and late) corresponding to the overlap of this phase with the phases of the previous cycle.

- Tommy Atkins.

These trees were fully described for (see figure 5.2):

- Vegetative GUs bursting between September, 2003 and November 2005.
- Reproductive GUs flowering or fructifying between July, 2004 and March, 2006.

Since the mango growth cycle of year i is a period ranging from July, 1^{st} of year $i-1$ to March, 1^{st} of the year $i+1$, 2 growth cycles were completely observed (see figure 5.2), but the complete tree topology has been reported.

Temporal resolution While the patchiness is a spatio-temporal phenomenon, we focus here on its spatial aspect on given trees observed at given dates (see figure 5.3).

In particular, considering a growth cycle (see figure 5.2) there are 3 periods of marked interest:

The early flush. The early flush corresponds to the period where the vegetative phase of a growth cycle overlaps the flowering phase of the previous cycle.

The intermediate flush. The intermediate flush corresponds to the period where the vegetative phase of a growth cycle overlaps the fructifying phase of the previous cycle.

The late flush. The late flush corresponds to the period where the vegetative phase of a growth cycle does not overlap the previous or the next cycles.

Patchiness was therefore investigated at the flush temporal resolution. For each of these flushes and each growth cycles, tree-indexed data were extracted from plants at the GU scale as follows:

1. Any GU which burst occurred before the considered date was not considered.
2. Any reproductive GU which flowering occurred in previous growth cycle was not considered because of the limited lifetime of these structures.
3. Any GU which burst or flowering occurred in the current growth cycle and flush was labeled as F for a reproductive GU or V for a vegetative GU.
4. Any leaf of the tree graph which had no label was labeled R for resting GU.

As a consequence we obtained 181 trees within which mostly leaf vertices were observed with the following observation space[2]

$$\mathcal{X} = \{F, R, V\}\,.$$

5.3 Results

5.3.1 Tree segmentation

Over the 181 trees, only 132 were successfully segmented. The major reason of these failures was the presence of trees with a very low level of noise, therefore over-parametrized models for penalty computation could not be computed for these trees. Note that even if we did not considered these trees, these failures to obtain over-parametrized models could be considered as an indication of trees that are patches.

Nevertheless, as illustrated on figure 5.4, the segmentation of trees successfully detected 608 patches with various compositions and relative sizes. Note that only few patches of height 0 were detected (6%), which indicates that there were relatively few over-segmented trees.

[2]observations are defined by GUs characteristics abbreviations instead of positive integers for clarity purposes



Figure 5.3 – Illustration of mango tree patchiness (Dambreville, 2012). This mango tree is separated into two parts. The left one in dark green is a clump of old GUs where fruits can be found. In contrary the right one in light green is a clump of new vegetative GUs. This visual patchy aspect is mostly due to GUs situated in the canopy at a given date.

5.3.2 Subtree clustering

If the composition of patches varies, most of them were close to the vegetative, flowering or resting poles (see figure 5.4). The second stage of clustering was therefore highly relevant since the occurrence of similar non-adjacent subtrees in the tree was an important feature.

For the mixture model, we considered three different states that enabled the clustering of subtrees into 3 clusters and to assess the general composition of theses patches (see figure 5.4). According to the observations distributions:

Flowering patches have been assigned to state 0,

$$f_0\,(\mathrm{F}) = 0.7,$$
$$f_0\,(\mathrm{R}) = 0.26,$$
$$f_0\,(\mathrm{V}) = 0.04.$$

Vegetative patches have been assigned to state 1,

$$f_1\,(\mathrm{F}) = 0.08,$$
$$f_1\,(\mathrm{R}) = 0.13,$$
$$f_1\,(\mathrm{V}) = 0.79.$$

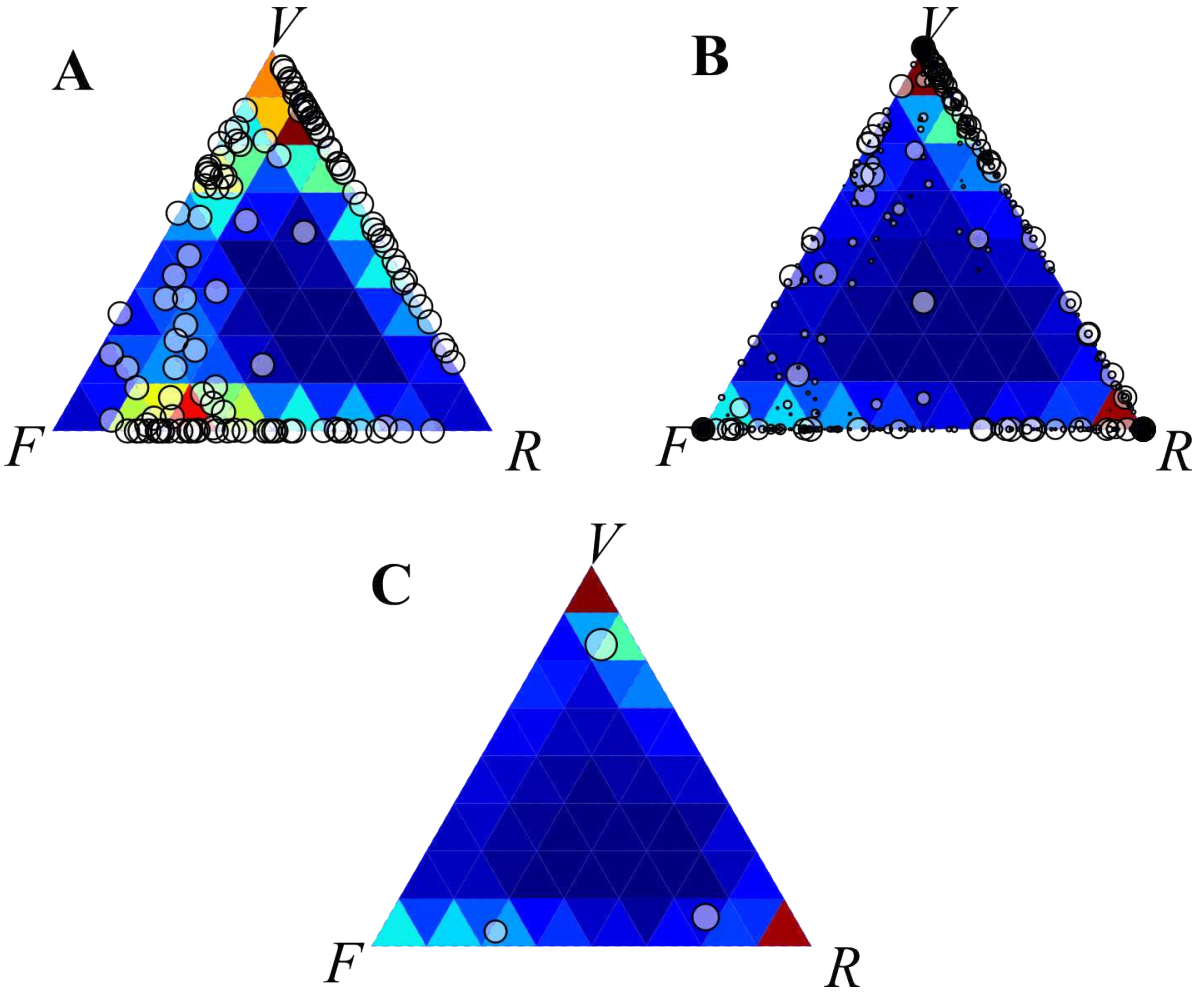


Figure 5.4 – Ternary plots of the outputs of the segmentation/clustering algorithm (A) Ternary plot of the initial trees. (B) Ternary plot of the segmented trees. In both graphs each tree or subtree is identified by a blank disk, which size is proportional to its relative size with respect to the original tree. (C) Ternary plot of clustered subtrees. In this graph each cluster of subtrees is identified by a blank disk, which size is proportional to its weights in the inferred mixture model. In these ternary plots, the left bottom corner of the triangles represents the pure flowering trees, the right bottom corner the pure resting trees and the top corner the pure vegetative trees. Therefore, a tree near a corner of the triangle is an almost pure tree. On the contrary, if it is near an edge it has a very low proportion of the characteristic represented at the corner opposed to the edge. The colored triangles in the background of these ternary plots correspond to bins of histograms colored according to a heat map (from dark blue corresponding to low tree frequency to red for high frequency). In (A), the histogram of initial trees is represented and in (B,C), the histogram of segmented subtrees are represented.

Resting patches have been assigned to state 2,

$$f_2(\mathrm{F}) = 0.2,$$
$$f_2(\mathrm{R}) = 0.72,$$
$$f_2(\mathrm{V}) = 0.08.$$

According to the weights

$$\pi_0 = 0.22,$$
$$\pi_1 = 0.46,$$
$$\pi_2 = 0.32,$$

there is a slight excess of vegetative patches, but all patches are clearly presents. Note that this excess of vegetative patches can be biologically understood since the observed mango trees were young and therefore not at their permanent production regime, which induces more flowering GUs.

While there is quite an opposition between vegetative and flowering GUs within patches, resting GUs are present in non negligible quantities in each patches.

5.3.3 Cultivar comparisons

The interest of tree segmentation/clustering models is that given the patches and their identities, the different cultivars can be compared. For instance, we computed by cultivar (see figure 5.5):

The relative sizes of patches. The empirical cumulative distribution functions of relative sizes of patches enabled the comparison of cultivar behaviors in terms of patch sizes. The relative size of a patch is defined as the ratio of the number of vertex in the patch to the number of vertex in the complete tree. If most of the cultivars have almost the same behavior, three cultivars are relatively different. Irwin has the biggest patches, in contrast to Tommy Atkins that has the smallest patches. Jose is also quite interesting since it is the cultivar with the most heterogeneous patch sizes and contrarily to other cultivars it has no marked plateau for intermediate patch sizes.

The MAP assignment of quotients. MAP assignment of quotients brings information about patch representations in cultivars. The most marked differences concern Tommy Atkins. Tommy Atkins has only 2 categories of patches, flowering patches being quasi-absent, these being partly compensated by a significant proportion of flowers in resting patches.

5.4 Discussion

Performance Our segmentation approach is based on an heuristic. We therefore assessed the performance of this heuristic approach assuming that the number of quotients was known.

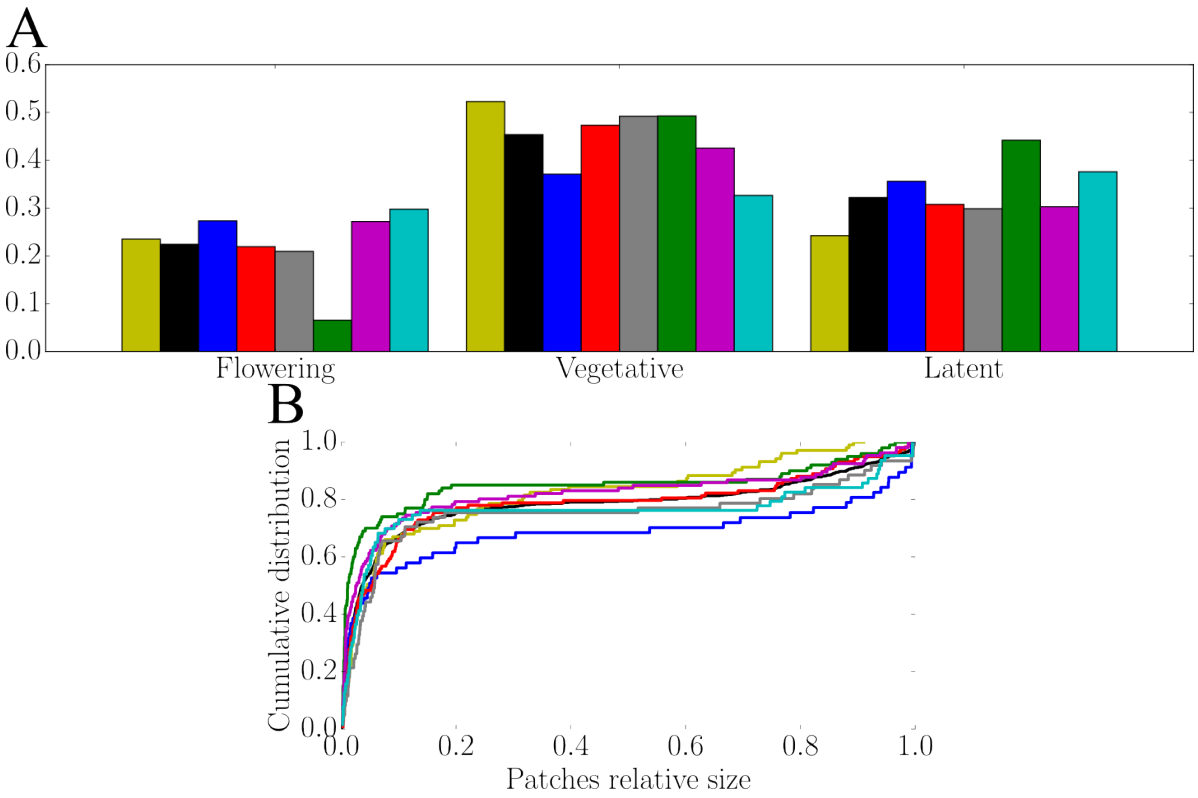


Figure 5.5 – Comparisons of patch patterns for the different cultivars. (A) MAP assignment of quotients. (B) Cumulative distribution functions of the relative size of patches. The relative size of a patch is defined as the ratio of the number of vertex in the patch to the number of vertex in the complete tree. For both graphs, Cogshall cultivar is in red, Jose in yellow, Kensington Pride in magenta, Tommy Atkins in green, Nam Doc Mai in cyan, Irwin in blue, Kent in gray and the all cultivar together are represented in black.

To this end we simulated 100 different trees using simple Watson and Galton (1875) processes with patches at random heights. Once the height was simulated, given a topological ordering of the change points, their identities were simulated with periodic Markov chains of period 2 (two consecutive vertices cannot thus have the same identity). Then, each of these identities were projected onto corresponding leaf vertices. For each of these leaf-labeled trees, 10 different noise intensities (ranging from 0.0 up to 1.0) were simulated, with the noise intensity defined as the frequency of re-labeled vertices.

For these $1,000$ trees obtained, our heuristic method was used to recover the quotienting corresponding to the number of simulated quotients. As presented in Denœud and Guénoche (2006), the comparison of obtained and simulated quotienting was based on the comparison of their quotienting matrices. A quotienting matrix $\underline{\Pi}$ of a given quotienting Π of vertices $\mathcal{T}$, is the square matrix of general element $\underline{\Pi}_{i,j}$ defined as

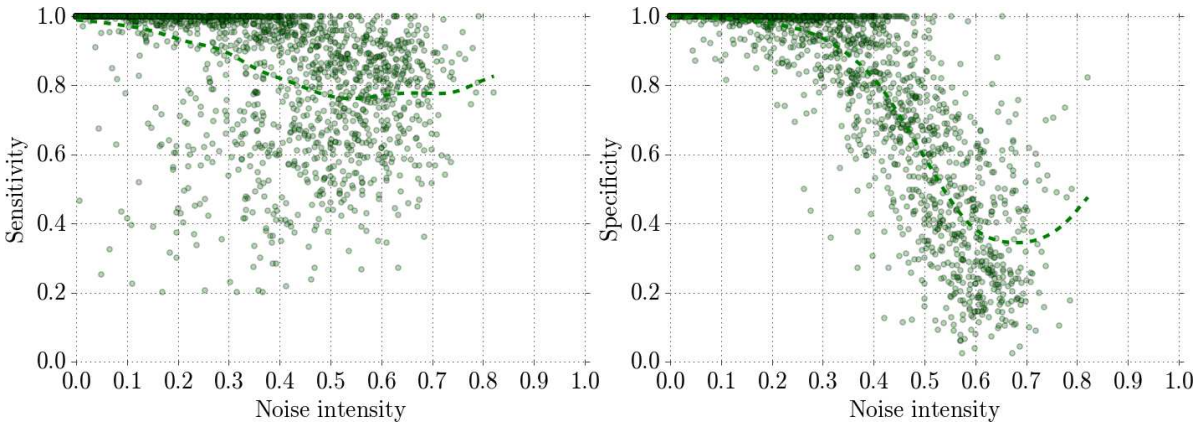


Figure 5.6 – Performance of the segmentation heuristic for tree-indexed data. This performance was assessed with a simulation study by comparing the simulated and segmented quotients. These comparisons were conducted using the sensitivity and specificity scores of the results.

follows:

$$\forall (i,j) \in \mathcal{T}^2, \underline{\Pi}_{i,j} = \begin{cases} 0 & \text{if } [i=j], \\ 1 & \text{if } \exists \pi \in \Pi,\ [i \in \pi] \wedge [j \in \pi], \\ 0 & \text{otherwise.} \end{cases}$$

Comparisons of specificity and sensibility of such matrices indicated that the approach was well adapted in order to recover the simulated quotienting (see figure 5.6). Note that, even in some cases of very low noise, the sensibility can be surprisingly low. The reason of this problem is due to identifiably issues that can be summarized with the following question: 'Is it a flowering tree with vegetative patches or a vegetative tree with flowering patches?'. At some point, if proportions of simulated states are quite similar, a small level of noise can make the difference, if it was considered as 'a flowering tree with vegetative patches' but that the heuristic method found that it was 'a vegetative tree with flowering patches', the corresponding comparison of simulated and segmented quotients induces a low sensibility but a high the specificity.

Scale comparisons The scale of patch expression is of marked interest. If in one tree this can be tackled by the comparison of height, depth or width distributions according to the different type of patches, as soon as we are in presence of a forest, this is approach is no more relevant. Therefore, the approach consisting in computing distributions of relative heights, depths or widths with respect to the tree within which the patch is found can be used (see figure 5.5). Nevertheless, since plant topology can be described formally through MTGs, it could be relevant to consider the quotiented tree resulting from tree segmentation as an inferred scale and to compare it to the biological scales encoded into the MTG. This could be tackled using distances between tree quotienting defined by Ferraro et al. (2003). Using this distance, the distances between tree quotienting obtained by the segmentation stage and the nested biological

quotientings could contribute to identify the scale of patchiness patterns within the different cultivars and their modifications across time.

References

J.-P. Baudry, C. Maugis, and B. Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012. URL http://link.springer.com/article/10.1007/s11222-011-9236-1. 112

E. Chacko. Physiology of vegetative and reproductive growth in mango (*Mangifera indica* L.) trees. In *Proceedings of the First Australian Mango Research Workshop*, volume 1, pages 54–70. CSIRO Australia, Melbourne, 1986. 109

M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4): 886–902, 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668544. 109

A. Dambreville. *Croissance et développement du manguier (*Mangifera indica *L.) in natura: approche expérimentale et modélisation de l'influence d'un facteur exogène, la température, et de facteurs endogènes architecturaux.* PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2012. URL http://hal.archives-ouvertes.fr/tel-00860484/. 116

A. Dambreville, P. Fernique, C. Pradal, P.-E. Lauri, F. Normand, Y. Guédon, and J.-B. Durand. Deciphering mango tree asynchronisms using Markov tree and probabilistic graphical models. In R. Sievänen, E. Nikinmaa, C. Godin, A. Lintunen, and P. Nygren, editors, *FSPM2013 - 7th International Workshop on Functional-Structural Plant Models*, pages 210–212, Saariselkä, Finlande, 2013. URL http://hal.inria.fr/hal-00847614. ISBN 978-951-651-408-9. 113

L. Denœud and A. Guénoche. Comparison of distance indices between partitions. In *Data Science and Classification*, pages 21–28. Springer, 2006. URL http://link.springer.com/chapter/10.1007/3-540-34416-0_3. 119

J.-B. Durand, P. Goncalvès, and Y. Guédon. Computational methods for hidden Markov tree models–An application to wavelet trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1323262. 109

J.-B. Durand, Y. Guédon, Y. Caraglio, and E. Costes. Analysis of the plant architecture via tree–structured statistical models: the hidden Markov tree models. *New Phytologist*, 166(3):813–825, 2005. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.2005.01405.x/full. 109, 110

P. Ferraro, C. Godin, et al. An edit distance between quotiented trees. *Algorithmica*, 36(1):1–39, 2003. URL http://link.springer.com/article/10.1007/s00453-002-1002-5. 120

C. Godin and Y. Caraglio. A multiscale model of plant topological structures. *Journal of Theoretical Biology*, 191(1):1–46, 1998. URL http://www.sciencedirect.com/science/article/pii/S0022519397905610. 110

D. M. Hawkins. Point estimation of the parameters of piecewise regression models. *Applied Statistics*, 25(1):51–57, 1976. 112

P. Hupé, N. Stransky, J.-P. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, 20(18):3413–3422, 2004. URL http://bioinformatics.oxfordjournals.org/content/20/18/3413.short. 109

S. Lauritzen. *Graphical models*, volume 17. Oxford University Press, 1996. 110

É. Lebarbier. Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal processing*, 85(4):717–736, 2005. URL http://www.sciencedirect.com/science/article/pii/S0165168404003196. 112

G. McLachlan and D. Peel. *Finite mixture models*. Wiley New York, 2004. 112

A. B. Olshen, E. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–572, 2004. URL http://biostatistics.oxfordjournals.org/content/5/4/557.short. 109

F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin. A statistical approach for array CGH data analysis. *BMC bioinformatics*, 6(1):27, 2005. 109

F. Picard, S. Robin, E. Lebarbier, and J.-J. Daudin. A segmentation/clustering model for the analysis of array CGH data. *Biometrics*, 63(3):758–766, 2007. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1541-0420.2006.00729.x/full. 109, 111

F. Ramírez and T. L. Davenport. Mango (*Mangifera indica* L.) flowering physiology. *Scientia Horticulturae*, 126(2):65–72, 2010. URL http://www.sciencedirect.com/science/article/pii/S0304423810002992. 109

G. Rigaill, E. Lebarbier, and S. Robin. Exact posterior distributions and model selection criteria for multiple change-point detection problems. *Statistics and computing*, 22(4):917–929, 2012. URL http://link.springer.com/article/10.1007/s11222-011-9258-8. 112

H. W. Watson and F. Galton. On the probability of the extinction of families. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 4:138–144, 1875. 119

N. R. Zhang and D. O. Siegmund. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32, 2007. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1541-0420.2006.00662.x/full. 112

Work in progress and perspectives

This thesis aimed at proposing a statistical modeling framework for studying patterns in tree-indexed data. To this end, two major different classes of statistical models were investigated.

In chapter 3, we considered Hidden Markov Out-Tree (HMOT) models relying on local dependency assumptions and dedicated to motif analysis. A generalization of state-of-the-art HMOT models have been introduced in order to consider dependences between siblings and randomness of the generation process. This work required the design of an upward-downward smoothing algorithm in order to implement efficiently the E-step of the Expectation-Maximization (EM) algorithm and a dynamic programming algorithm for the restoration of the most probable state tree for this family of models. The upward-downward algorithm developed corresponds to an instance of the generic algorithm for graphical models proposed by Jensen et al. (1990). Compared to this generic algorithm, the more dedicated upward-downward algorithm have the following desirable properties:

- The upward and downward recursions are numerically stables.
- It is a true smoothing algorithm and its outputs, that is state profiles, can be used as a diagnostic tool.
- Intermediate results of the upward-downward algorithm, clearly defined in terms of conditional probabilities can be used in different contexts such as, the computation of the log-likelihood of the observed data or the simulation of a state tree given an observed tree.

In chapter 4, focus was on unordered trees with high number of child vertices in the context of a categorical observed process. Since the state process of Hidden Markov Unordered Out-Tree (HMUOT) models studied in chapter 3 is modeled by Multi-Type Branching Processes (MTBPs), this chapter considered the case of MTBPs and focused on the design of parametric version of HMUOT models in a simpler case. Inference of MTBPs mostly reduces to inference of their generation distributions, which are discrete multivariate distributions. Since the analysis of multivariate count data is a recurrent and crucial issue in numerous modeling problems, particularly in the fields of biology, ecology, sociology and econometrics, the scope of the problems that could be dealt with such models is much more larger than the considered application to MTBPs. In order to characterize dependences between components of these discrete multivariate distributions, we introduced Discrete Parametric Mixed Acyclic Graphical Model (DPMAGM). If parameter inference for such models is a classical issue in statistics we considered the inference of the structure that has been considered less often in the literature. Structure inference was tackled using a local search within the Quotient Acyclic Graph (QAG) search space instead of the Mixed Acyclic Graph (MAG) one in order to minimize the probability of being stuck in local optima. The interest of this search space is not limited to DPMAGMs and we are now considering the case of Gaussian Mixed Acylic Graphical Model (GMAGM) to test and compare our heuristic to the local search space in the

MAG space and other methods (Edwards, 2000; Ma et al., 2008; Drton and Perlman, 2008).

In chapter 5, the generalization of multiple change-points models from path-indexed data to tree-indexed data was investigated. In contrast to HMOT models, such models belong to the class of long-range dependency models and are suitable for segmentation analysis. Since exact methods for determining the most probable segmentation of path-indexed data cannot be transposed to tree-indexed data, we proposed an effective heuristic approach. If in our application context, focus was on categorical variables, this approach can also be applied to any types of random variables or vectors. Note that in a few cases where decompositions of the log-likelihood are available (e.g. Poisson piecewise constant parameters or Gaussian change in mean and variance models), little work could be done in order to improve time and space complexity of such algorithms.

In the remainder of this chapter we describe work in progress and perspectives conveyed by this thesis. We first focus on the effort dedicated to software development. All methods and models developed by team members are integrated in a common software component, *V-Plants*, within the *OpenAlea* platform (Pradal et al., 2008). An overview of the software resulting from the implementation of statistical models and methods developed in this thesis in order to make them available to the team members and partners is therefore done. Then, if chapter 3 focus was on HMOT models, Hidden Markov In-Tree (HMIT) – discussed by Durand et al. (2005) and developed by Bacciu et al. (2010) – are related models that also take into account dependencies between children. Such models and their parametrizations are therefore discussed with respect to HMOT models. Concerning the generation distributions of HMOT, in chapter 4 we considered the use of graphical models in order to reveal exclusion and inclusion patterns in child fates. An alternative model, based on mixture models, is presented and the different hypotheses induced by these two models are hereafter discussed. Finally, we revisit the patchiness phenomenon discussed in chapters 4 and 5 and present an integrative analysis that could be conducted in order to decipher mango tree asynchronisms and patchiness phenomena.

StatisKit: graphical model inference in C++ and Python

Project description *StatisKit* is a bipartite library (C++ and Python) developed during the thesis concerning the domain of graphical models ranging from Hidden Markov Tree (HMT) models inference on the basis of tree-indexed data to undirected graph, Directed Acyclic Graph (DAG) and Mixed Acyclic Graph (MAG) model parameters and structure inference for multivariate mixed data. It is distributed under the CeCILL-C license as a package of the *OpenAlea* platform, encouraging its use and development in academic settings. *StatisKit* provides implementations of most recent work of Virtual Plants team in the field of statistics (in particular methods presented in Peyhardi (2013) and in this thesis) This work started in 2011 and a beta version

Programming Language	Source lines of code	
	Count	Percentage
C++	**38, 207**	**57.16%**
Python	**28, 637**	**42.84%**
All	**66, 844**	

Table 6.1 – Repartition of lines of code in StatisKit. *This table uses data generated by Wheeler (2001)* SLOCCout *program.*

was publicly released as a package of the *OpenAlea* platform in 2013[3]. Since then, the project have been augmented with Markovian models and segmentation algorithms for tree-indexed data and a new released is planned at the end of 2014.

Underlying technologies The C++ and Python programming languages are popular languages for scientific computing. The C++ language designed to be compiled into low-level code allows to design efficient libraries. Nevertheless, such libraries are not intuitive and easy to manipulate for data analysis. The high-level interactive nature of Python language is an appealing choice for non-specialists of computer science such as biologists. In *OpenAlea*, a common choice is therefore to combine C++ language for the design of library and Python language for the definition of its Application Programming Interface (API) (see table 6.1).

Code design To facilitate the extensibility of the module a particular attention has been paid to the inheritance diagrams of model and estimators classes. Moreover, rather than providing as many features as possible, the project goal has been to provide solid implementations of estimators. Hence, the code quality is ensured by:

1. The design of generic algorithms that can be tested on benchmark data. This is done by an extensive use of templates and virtual classes in C++ allowing code factoring.

2. The design of specific algorithms that are more efficient than the former ones but more specialized. It is in particular possible to test by simulation complexity improvements and to ensure that results are consistent with the generic algorithms.

For object of database types no hierarchy is imposed. The only requirement of such classes is to have a clear identification of models available (e.g. univariate distributions for univariate data, tree processes for tree-indexed data) in order to build corresponding model selection environment (classes combining a data and the best model or the ordered or unordered collection of proposed models during estimation procedures).

[3]Available under the name *statistic* at http://openalea.gforge.inria.fr/dokuwiki/doku.php?id=packages:statistic:statistic

Diffusion Stability of releases are ensured in the Python version using unit and functional tests (with *Nose* and *DocTest* Arbuckle, 2010). Moreover, in a context of reproducible research, all analyzes performed in this thesis are available in *IPython Notebooks* (Perez and Granger, 2007) tutorials combining our models and a few *R* procedures (R Development Core Team, 2011). At this stage, the software is mostly missing licenses in headers and functions documentation required to be published.

Dependencies The minimal dependency for installation is the *OpenAlea* platform core (Pradal et al., 2008, for deployment), the *Eigen* library (Guennebaud et al., 2010, for linear algebra), *Boost* libraries (Schling, 2011, for many things). For an usage in Python the usual packages *Numpy* (Van Der Walt et al., 2011, for linear algebra data structure and basic arithmetic operations), *Scipy* (Jones et al., 2001, for linear algebra) and *Matplotlib* (Hunter, 2007) are required.

Hidden Markov In-Tree (HMIT) models

In chapter 3 we only focused on Hidden Markov Out-Tree (HMOT) models but contrarily to sequences, directed trees are non-symmetrical structures and this induces fundamental differences in model parametrization for Markov Out-Tree (MOT) (edges directed from the root to the leaves) and Markov In-Tree (MIT) models (edges directed from the leaves to the root). For a given tree structure with vertices labeled with discrete states, we have a small number of potentially complex multivariate transition distributions in the MOT model case and a large number of simple univariate transition distribution in the MIT model case. The parametrization of a MIT model is analogous to the parametrization of a high-order Markov Chain (MC) model and the different approaches to build parsimonious high-order MC models (i.e. full parametric approaches such as the mixture transition distribution model of Raftery (1985) and memory selection approaches leading to variable-order Markov chains (Ron et al., 1996; Bühlmann et al., 1999; Csiszár and Talata, 2006) can be transposed to MIT models.

Parsimonious Markov Ordered In-Tree (MOIT) models Considering the Markov property, vertices are independent of their descendants given their children

$$\forall t \in \mathcal{T},\ X_t \perp\!\!\!\perp \bar{X}_{\mathrm{de}(\mathrm{ch}(t))} \,\Big|\, \bar{X}_{\mathrm{ch}(t)}\ .$$

This is equivalent to the following factorization of the process distribution

$$P\left(\bar{X} = \bar{x}\right) = \prod_{l \in \mathcal{L}} P\left(X_l = x_l\right) \prod_{t \in \mathcal{T} \setminus \mathcal{L}} P\left(X_t = x_t \,\Big|\, \bar{X}_{\mathrm{ch}(t)} = \bar{x}_{\mathrm{ch}(t)}\right). \tag{6.1}$$

leading to MIT models. It is worth noting that due to the opposite direction – from leaf to root vertices – of the Markov property, random generation of children cannot be modeled as in MOT model cases. According to (6.1) a MIT model is defined by the following parameters:

- An initial distribution for leaf vertices,

$$\forall l \in \mathcal{L},\ \pi_{x_l} = P\left(X_l = x_l\right),$$

with $\sum_{x \in \mathcal{X}} \pi_x = 1$.

- Transition distributions from child vertices to their parent for each given configuration of child states

$$\forall v \in \mathcal{V},\ \Gamma_{\bar{x}_{\mathrm{ch}(v)}}\left(x_v\right) = P\left(X_v = x_v \,\middle|\, \bar{X}_{\mathrm{ch}(v)} = \bar{X}_{\mathrm{ch}(v)}\right),$$

with

$$\forall x \in \mathcal{X},\ \forall d \in \deg^+(\mathcal{T} \setminus \mathcal{L}),\ \forall \bar{x} \in \mathcal{X}^d,\ \sum_{x \in \mathcal{X}} \Gamma_{\bar{x}}\left(x\right) = 1.$$

Without any further hypotheses there is a total of

$$|\mathcal{X}| - 1 + (|\mathcal{X}| - 1) \sum_{d \in \deg^+(\mathcal{T} \setminus \mathcal{L})} |\mathcal{X}^d|$$

independent parameters to define but in practice, the parametrization of a MIT model is analogous to the parametrization of a high-order MC model, the out-degree of the vertices playing the role of the order of the MC. Concerning high-order MC models for path-indexed graphs two approaches have been proposed to build parsimonious models:

Mixture transition distribution. These models were introduced by Raftery (1985) and later generalized as Mixed Memory Markov Chain (MMMC) models by Saul and Jordan (1999). This kind of parametric high-order MC relies on an analogy with Auto-Regressive (AR) models where high-order transition probabilities are represented as convex combinations – or mixture – of first-order transition probabilities. The constraints on the dependencies induced by this kind of parametric modeling stay unclear. This principle has been transposed to Hidden Markov In-Tree (HMIT) models by Bacciu et al. (2010).

Variable-order Markov chain. In these models (Ron et al., 1996; Bühlmann et al., 1999; Csiszár and Talata, 2006), the memory length is variable and depends on the context. The idea here is to aggregate memories with similar suffixes (i.e. most recent states) that share the same transition distributions. The aggregation of memories relies on the fact the successive states in memories are ordered. Contrarily to mixture transition distribution models and mixed memory Markov models, this nonparametric approach does not impose constraints on the dependencies that can be represented. This approach could be transposed to MIT model by imposing order constraints either on the children of a vertex and/or on the states.

Parsimonious Markov Unordered In-Tree (MUIT) models In the case of Markov Unordered In-Tree (MUIT) models, transition distributions simplify into

$$\forall t \in \mathcal{T} \setminus \mathcal{L},\ \Gamma_{\boldsymbol{n}_t}\left(x_t\right) = P\left(X_t = x_t \,|\, \boldsymbol{N}_t = \boldsymbol{n}_t\right),$$

with

$$\forall \boldsymbol{n} \in \mathbb{N}^{|\mathcal{X}|}, \sum_{x \in \mathcal{X}} \Gamma_{\boldsymbol{n}}(x) = 1.$$

which imply a total of

$$|\mathcal{X}| - 1 + (|\mathcal{X}| - 1) \sum_{d \in \deg^{+}(\mathcal{T} \setminus \mathcal{R})} \binom{|\mathcal{X}| + d - 1}{d},$$

independent parameters, but in practice each of these transition distributions can be parsimoniously modeled by categorical regressions (Tutz, 2011).

If we reconsider the example developed in chapter 3, since the roots of lineage trees are not systematically at time 0h, it would be relevant to orient the lineage trees from the leaf vertices at the last time point to the roots. This leads to the Hidden Markov Unordered In-Tree (HMUIT) model, which is parametrized by transition probabilities. The transition probability matrix is described by the regression matrix of the categorical regressions. Since branching viewed backward is coalescence, this model can be viewed as a hidden coalescence process. We expect this model to give complementary information, with respect to the HMOT, regarding in particular the cell identities identified at the first time points on the basis of cell identities propagated from the last time points (see figure 6.7). Nevertheless simple categorical regressions suffer from the fact that left-right models cannot be defined. In order to allow an ordering or a partial ordering on states it could highly relevant to consider the extension of categorical regressions proposed by Peyhardi et al. (2014a,b).

Multivariate mixture models in Multi-Type Branching Processes (MTBPs)

In chapter 4, we considered discrete multivariate parametric distributions that fulfill the following criteria:

2. These multivariate parametric distributions can have zero-inflated, right-skewed and natural number valued marginals, so that discretized multivariate Gaussian distributions are not appropriate.

3. These multivariate parametric distributions can easily be simulated and probability masses can easily be computed in order to investigate motifs induced by generation distributions and long-range patterns stemming from these generation distributions as trees develop.

4. Child states that tend to appear simultaneously or on the contrary to be incompatible can be identified.

To this end, we defined Discrete Parametric Mixed Acyclic Graphical Model (DPMAGM) that are consistent with criterion (4). Nevertheless this imposed the constraint of gradual changes considering exclusion patterns. It could therefore be interesting to

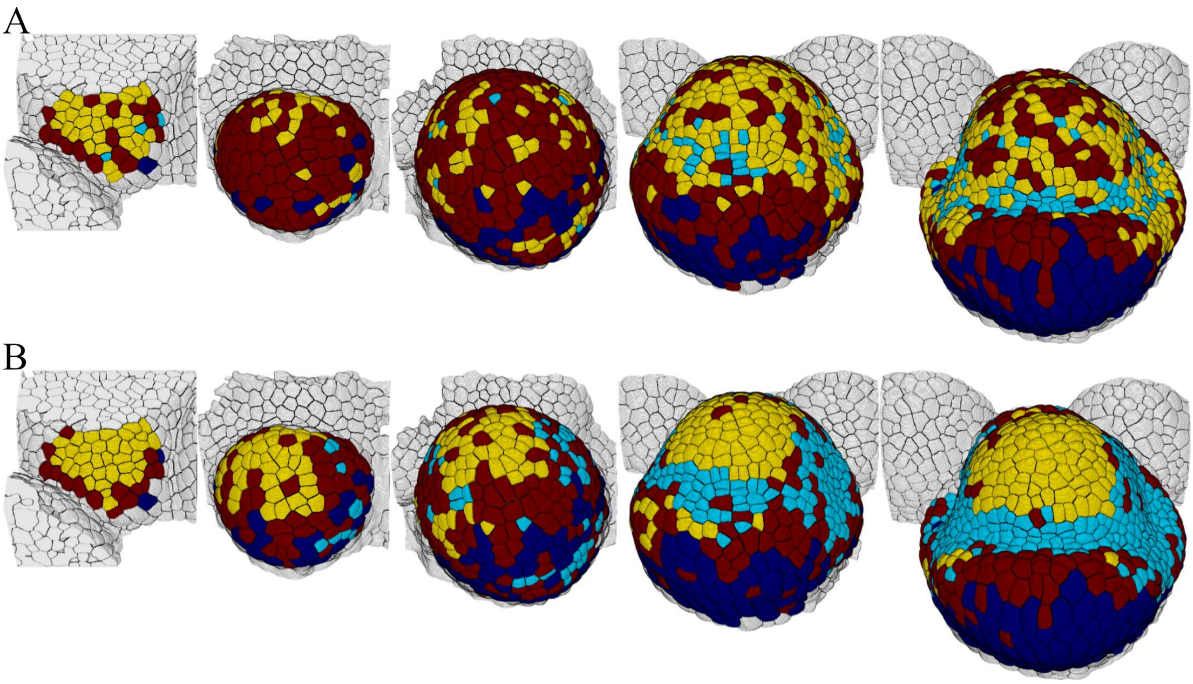


Figure 6.7 – Comparison of the restoration of hidden states using the Viterbi-like algorithm for HMIT and HMOT models. Images, from left to right, were respectively taken at 0h, 26h, 44h, 56h *and* 69h *after the beginning of the experiment. (A) Spatial projection of the four states obtained using the HMUIT model. (B) Spatial projection of the four states obtained using the Hidden Markov Unordered Out-Tree (HMUOT) model. Both models were estimated using epidermis surface, internal surface, volume, curvatures and inertia as cell characteristics. State* 0 *is in dark blue, state* 1 *in light blue, state* 2 *in yellow and state* 3 *in dark red. Sepals are mostly identified by states* 0 *and* 3*, the dome by state* 2 *and boundary cells by state* 1*.*

consider models, in which on contrary abrupt changes are considered. To this end, we propose to use the following mixture decomposition of the generation distributions

$$\begin{aligned} P\left((N)_{x\in\mathcal{X}} = (n_x)_{x\in\mathcal{X}}\right) &= P\left(\boldsymbol{N} = \boldsymbol{n}\right) \\ &= \sum_{m\in\mathcal{M}} \pi_m P_m\left(\boldsymbol{N} = \boldsymbol{n}\right), \end{aligned}$$

where $\mathcal{M} \subset \mathbb{N}$ represents the set of components, $(\pi_m)_{m\in\mathcal{M}}$ the weights of the components and $(P_m(\cdot))_{m\in\mathcal{M}}$ are the discrete multivariate parametric generation distributions, which fulfill criteria (2)-(4). These mixture models have been used extensively in the literature, especially for representing the univariate case the over-representation of zeros. Note that in the multivariate case, this issue is more difficult since there are different ways of seeing the over-representation of zeros. We propose to address this issue as a problem of variables quotienting with $|\mathcal{M}|$ quotients such as:

$$\forall m \in \mathcal{M},\, \forall x \in \mathcal{X} \setminus \mathcal{X}_m,\, P_m\left(N_x > 0\right) \approx 0,$$

where $\mathcal{X} = \bigcup_{m \in \mathcal{M}} \mathcal{X}_m$. We therefore address this issue by using a model within which in each component there is only one quotient of states has a significant number of children and such that these quotients have only a significant number of children in only one component. The mixture model thus fulfills criterion (4).

The major problem in this model is to identify the components. Let us consider the dichotomous random vector $(B_x)_{x \in \mathcal{X}}$ where

$$\forall x \in \mathcal{X},\ B_x = \begin{cases} 0 & \text{if } N_x = 0, \\ 1 & \text{otherwise.} \end{cases}$$

and the random variable $S = \sum_{x \in \mathcal{X}} B_x$. Precise investigations on the distribution of the random vector $(B_x)_{x \in \mathcal{X}}$ given $S \geq 2$ could be of marked interest are we expect to have:

- Negative covariances between pairs of variables that are not in the same components,
- Positive covariances between pairs of variables that are in the same components.

Let $\mathcal{G}$ be the graph of positive covariances composed of the vertex set $\mathcal{X}$ and the edge set $\mathcal{E}$. $\mathcal{E}$ is defined as the set of edges that correspond to positive covariances in the conditional dichotomous random vector,

$$\mathcal{E} = \left\{ (i,j) \in \mathcal{X}^2 \,\middle|\, [i \neq j] \wedge [Cov\,(B_i, B_j \,|\, S \geq 2) > 0] \right\}.$$

If there is a small level of noise, the quotienting could be identified by the connected components of the graph $\mathcal{G}$. In case of a high level of noise, a preliminary step of covariance selection (see for Yang et al. (2012, 2014) for examples) could be used to detect non-significant covariances.

Note that if this kind of procedure could identify the components, once the components are known the Expectation-Maximization (EM) algorithm (McLachlan and Peel, 2004) for parameter inference could be initialized quite easily by the following approximation of the initial posterior probabilities:

$$\forall \boldsymbol{n} \in \mathcal{N}^{|\mathcal{X}|}, \forall x \in \mathcal{X},\ b_x = \begin{cases} 0 & \text{if } n_x = 0, \\ 1 & \text{otherwise} \end{cases},$$
$$\forall m \in \mathcal{M},\ P\,(M = m \,|\, \boldsymbol{N} = \boldsymbol{n}) \propto \sum_{x \in \mathcal{X}} b_x \times \mathrm{I}\,(x \in m)\,,$$

where $\mathrm{I}\,(.)$ denotes the indicator function.

Integrative models for deciphering mango tree asynchronisms

In chapter 5 the patches were assimilated to hidden states in the subtree clustering stage. The motif analysis conducted in chapter 4 could therefore be enhanced and

used to decipher more precisely mango tree asynchronisms incorporating hidden states, assimilated to patch identities. The use of HMUOT defined in chapter 3 combined with the DPMAGM studied in chapter 4 for modeling generation distributions would lead to parametric HMOT where motifs could be easily interpreted considering the graphical representation of generation distributions.

When considering the motifs highlighted by generation distributions, we only capture a local point of view on asynchronisms. This local point of view can be turned into a more integrated view by simulating state trees corresponding to the observed mango trees. The change-point detection algorithm defined in chapter 5 could then be applied to these simulated trees. This step would be used in order to test whether the long-range patchiness patterns emerge or not when these motifs are chained during tree growth.

References

D. Arbuckle. *Python Testing: Beginner's Guide*. Packt Publishing Ltd, 2010. 127

D. Bacciu, A. Micheli, and A. Sperduti. Bottom-up generative modeling of tree-structured data. In K. Wong, B. Mendis, and A. Bouzerdoum, editors, *Neural Information Processing. Theory and Algorithms*, volume 6443 of *Lecture Notes in Computer Science*, pages 660–668. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17536-7. doi: 10.1007/978-3-642-17537-4_80. URL http://dx.doi.org/10.1007/978-3-642-17537-4_80. 125, 128

P. Bühlmann, A. J. Wyner, et al. Variable length Markov chains. *The Annals of Statistics*, 27(2):480–513, 1999. URL http://projecteuclid.org/euclid.aos/1018031204. 127, 128

I. Csiszár and Z. Talata. Context tree estimation for not necessarily finite memory processes, via bic and mdl. *Information Theory, IEEE Transactions on*, 52(3):1007–1016, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1603768. 127, 128

M. Drton and M. D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008. URL http://www.sciencedirect.com/science/article/pii/S0378375807002303. 125

J.-B. Durand, Y. Guédon, Y. Caraglio, and E. Costes. Analysis of the plant architecture via tree–structured statistical models: the hidden Markov tree models. *New Phytologist*, 166(3):813–825, 2005. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1469-8137.2005.01405.x/full. 125

D. Edwards. *Introduction to graphical modelling*. Springer, 2000. 125

G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 127

J. D. Hunter. Matplotlib: A 2D graphics environment. *IEEE Transactions on Computing in Science & Engineering*, 9(3):0090–95, 2007. URL http://doi.ieeecomputersociety.org/10.1109/mcse.2007.55. 127

F. V. Jensen, S. L. Lauritzen, and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational statistics quarterly*, 4:269–282, 1990. URL http://forskningsbasen.deff.dk/Share.external?sp=S31b3ec00-15ae-11dc-a5a4-000ea68e967b&sp=Saau. 124

E. Jones, T. Oliphant, and P. Peterson. Scipy: Open source scientific tools for python. *http://www. scipy. org/*, 2001. 127

Z. Ma, X. Xie, and Z. Geng. Structural learning of chain graphs via decomposition. *Journal of machine learning research: JMLR*, 9:2847, 2008. 125

G. McLachlan and D. Peel. *Finite mixture models*. Wiley New York, 2004. 131

F. Perez and B. E. Granger. IPython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3):21–29, 2007. URL http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.53. 127

J. Peyhardi. *Une nouvelle famille de modèles linéaires généralisés (GLMs) pour l'analyse de données catégorielles; application à la structure et au développement des plantes*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2013. URL http://tel.archives-ouvertes.fr/tel-00936845/. 125

J. Peyhardi, C. Trottier, and Y. Guédon. Partitioned conditional generalized linear models for categorical data. *arXiv preprint arXiv:1405.5802*, 2014a. URL http://arxiv.org/abs/1405.5802. 129

J. Peyhardi, C. Trottier, and Y. Guédon. A new specification of generalized linear models for categorical data. *arXiv preprint arXiv:1404.7331*, 2014b. URL http://arxiv.org/abs/1404.7331. 129

C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier, and C. Godin. OpenAlea: a visual programming and component-based software platform for plant modelling. *Functional plant biology*, 35(10):751–760, 2008. URL http://www.publish.csiro.au/?paper=FP08084. 125, 127

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0. 127

A. E. Raftery. A model for high-order markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 528–539, 1985. URL http://www.jstor.org/stable/2345788. 127, 128

D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3):117–149, 1996. URL http://link.springer.com/article/10.1023/A:1026490906255. 127, 128

L. K. Saul and M. I. Jordan. Mixed memory Markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37(1):75–87, 1999. URL http://link.springer.com/article/10.1023/A:1007649326333. 128

B. Schling. *The boost C++ libraries*. Xml Press, 2011. 127

G. Tutz. *Regression for categorical data*, volume 34. Cambridge University Press, 2011. 129

S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *IEEE Transactions on Computing in Science & Engineering*, 13(2):22–30, 2011. URL http://scitation.aip.org/content/aip/journal/cise/13/2/10.1109/MCSE.2011.37. 127

D. A. Wheeler. SLOCCount, 2001. 126

E. Yang, P. Ravikumar, G. Allen, and Z. Liu. Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems 25*, pages 1367–1375, 2012. 131

E. Yang, Y. Baker, P. Ravikumar, G. Allen, and Z. Liu. Mixed graphical models via exponential families. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 1042–1050, 2014. 131

Index of references

Amestoy et al. (1996), 20, 40
Andersson et al. (1996), 35, 40
Bacciu et al. (2010), 3, 6, 62
Banerjee et al. (2008), 39, 40
Barrett et al. (2005), 51, 62
Barthélémy and Caraglio (2007), 55, 56, 58, 62
Barthélémy et al. (1989), 55, 62
Berry et al. (2004), 20, 40
Bloesch (1993), 49, 62
Bron and Kerbosch (1973), 19, 40
Buchheim et al. (2002), 49, 50, 62
Buntine (1991), 39, 40
Buntine (1996), 39, 40
Campbell and Reece (1984), 52, 63
Cazals and Karande (2008), 19, 40
Chacko (1986), 58, 63
Chan et al. (1997), 49, 63
Chickering (2002), 39, 41
Choi and Baraniuk (2001), 2, 51, 63
Cox and Wermuth (1993), 35, 41
Crouse et al. (1998), 2, 3, 51, 61–63
Dahl et al. (2005), 39, 41
Dambreville et al. (2013), 58, 63
Dambreville (2012), 58, 59, 63
Das et al. (2009), 54, 63
Dasgupta et al. (2001), 51, 63
Dempster (1972), 38, 39, 41
De Campos and Puerta (2001), 39, 41
Drton and Perlman (2008), 40, 41
Drton (2009), 35, 41
Durand et al. (2004), 2, 3, 61–63
Durand et al. (2005), 2, 3, 6, 51, 57, 60, 64
Eades (1984), 11–13, 41
Eades (1991), 49, 64
Edwards (2000), 40, 41
Ephraim and Merhav (2002), 3, 61, 64
Fernandez et al. (2010), 53, 54, 64
Fernandez (2010), 53, 64
Friedman and Goldszmidt (1997), 39, 41
Friedman et al. (1999), 39, 41
Friedman et al. (2008), 39, 41
Fruchterman and Reingold (1991), 12, 13, 41, 49, 64
Frydenberg (1990), 35, 42
Gamez et al. (2011), 39, 42
Gatsuk et al. (1980), 55, 64
Godin and Caraglio (1998), 47, 48, 55, 57, 58, 64
Haccou et al. (2005), 3, 60, 64
Hallé et al. (1978), 55, 64
Harris (2002), 3
Heckerman et al. (1995), 39, 42
Hunter (2007), 15, 18, 42, 51, 64
Johnson et al. (1997), 61, 64
Kimmel and Axelrod (2002), 3, 60
Kirkpatrick et al. (1983), 12, 42
Kobourov (2012), 11, 12, 42
Koller and Friedman (2009), 9, 35, 42
Lauritzen and Wermuth (1989), 35, 42
Lauritzen (1996), 9, 19, 30, 31, 35, 39, 42
Legrand (2014), 53–55
Ma et al. (2008), 40, 42
Meinshausen and Buhlmann (2006), 39, 42
Moussouris (1974), 31, 33, 42
Neapolitan et al. (2004), 40, 42
Nultsch (1998), 52
Olariu et al. (2009), 2, 51, 61
Pradal et al. (2008), 6
Ramírez and Davenport (2010), 58
Reingold and Tilford (1981), 49
Ronen et al. (1995), 3
Rose et al. (1976), 20, 43
Sabatier et al. (1998), 56
Speed and Kiiveri (1986), 39, 43
Spirtes et al. (2000), 40, 43
Sugiyama and Misue (1995), 14, 15, 17, 43, 49

Tamassia (2007), 11, 14, 43, 48–50
Tarjan (1972), 17, 18, 43
Tibshirani (1996), 39, 43
Tomita et al. (2006), 19, 43
Walker (1990), 49, 50
Watson and Galton (1875), 3
Wermuth and Lauritzen (1990), 37, 43
Wermuth (1980), 38, 39, 43
Wermuth (1992), 38, 39, 43
Yang and Chang (2002), 39, 43
Yang et al. (2012), 39, 43
Yang (2003), 3
Yannakakis (1981), 20, 44

“Nous avons les moyens de vous faire parler”

”Papa Schulz” dans *Babette s’en va-t-en guerre* réalisé par Christian-Jaque

Titre Un cadre de modélisation statistique pour l'analyse de données indexées par des arborescences – Application au développement des plantes à l'échelle microscopique et macroscopique

Résumé Nous nous intéressons à des modèles statistiques pour les données indexées par des arborescences. Dans le contexte de l'équipe Virtual Plants, équipe hôte de cette thèse, les applications d'intérêt portent sur le développement de la plante et sa modulation par des facteurs environnementaux et génétiques. Nous nous restreignons donc à des applications issues du développement de la plante, à la fois au niveau microscopique avec l'étude de la lignée cellulaire du tissu biologique servant à la croissance des plantes, et au niveau macroscopique avec le mécanisme de production de branches. Le catalogue de modèles disponibles pour les données indexées par des arborescences est beaucoup moins important que celui disponible pour les données indexées par des chemins. Cette thèse vise donc à proposer un cadre de modélisation statistique pour l'étude de patterns pour données indexées par des arborescences. À cette fin, deux classes différentes de modèles statistiques, les modèles de Markov et de détection de ruptures, sont étudiées.

Mots-clés Architecture des plantes; données indexées par des arborescences lignage cellulaire; modèle de détection de ruptures; modèle de Markov; modèle graphique

---

Title A statistical modeling framework for analyzing tree-indexed data – Application to plant development at microscopic and macroscopic scales

Abstract We address statistical models for tree-indexed data. In the context of the Virtual Plants team, host team of this thesis, applications of interest focus on plant development and its modulation by environmental and genetic factors. We thus focus on plant developmental applications, both at the microscopic level with the study of the cell lineage in the biological tissue responsible for the plant growth, and at the macroscopic level with the mechanism of production of branches. The catalog of models available for tree-indexed data is far less important than the one available for path-indexed data. This thesis therefore aims at proposing a statistical modeling framework for studying patterns in tree-indexed data. To this end, two different classes of statistical models, Markov and change-point models, are investigated.

Keywords Cell lineage; change-point model; graphical model; Markov model; plant architecture; tree-indexed data