Rapport de Parcours Recherche

École des Mines de Nancy

Inférence statistique de réseaux de gènes à partir de graphes dynamiques

Travail réalisé par Romain Maillard

encadré par Ulysse Herbach

Parcours Recherche effectué à
l'Institut Élie Cartan de Lorraine (UMR 7502),
Université de Lorraine, site de Nancy
2021-2022

Remerciements

Je tiens à remercier Ulysse Herbach pour son aide, ses conseils, sa bienveillance, ainsi que ses encouragements.

Table des matières

Introduction

On appelle *expression d'un gène* l'ensemble de processus biochimiques menant à la production de protéines associées à ce gène (fig. 1). La connaissance de ce processus est primordiale à la compréhension du vivant et à la résolution d'un grand nombre de problématiques (différenciation cellulaire, cancer).

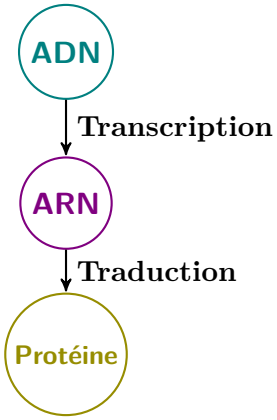


FIGURE 1 – *Schéma simplifié de l'expression d'un gène*

Cependant, comprendre et modéliser le fonctionnement de l'expression des gènes est un problème difficile, car ceux-ci interagissent entre eux via les protéines qu'ils produisent. Les méthodes utilisées par les biochimistes et les généticiens semblant insuffisantes, il faut s'intéresser à de nouvelles approches plus théoriques et abstraites comme celles que peuvent proposer les mathématiques.

L'approche envisagée durant ce projet recherche est celle de la modélisation des interactions par des graphes, dont la structure sera déterminée par inférence bayésienne exacte. Afin d'obtenir des résultats concrets et vérifier la solidité de la théorie, nous implémenterons des codes en Python en lien avec les modèles étudiés.

Chapitre 1

Modélisation du problème

1.1 Un problème biologique

Auparavant, on pensait que l'expression d'un gène était entièrement déterminée par sa séquence de nucléotides - suite de molécules ACTG - (fig. 1.1). Cependant, les chercheurs ont remarqué que le génome ne permet pas à lui seul de prédire quelles protéines vont être produites et en quelle quantité. En effet, deux cellules avec le même ADN et dans le même environnement ne vont pas exprimer leurs gènes de la même manière. Un exemple marquant est celui de la *différenciation cellulaire*, un processus durant lequel une cellule va se spécialiser en un autre « type » de cellule avec des fonctions différentes (cellules musculaires, osseuses, globules rouges, globules blancs...).

Un autre exemple montre également que l'expression des gènes n'est pas uniquement gouvernée par la séquence ADN. On peut expérimentalement observer la quantité d'ARN messager produite au sein d'une cellule par un certain gène grâce à des molécules devenant fluorescentes pendant la *transcription* (fig. 1). Plus le gène s'exprime, plus la cellule devient fluorescente dans cette cellule et produit des protéines. On se rend alors compte expérimentalement que le niveau de fluorescence des cellules est très varié (fig. 1.2) !

L'expression génétique problème complexe, où de **l'aléatoire apparaît pendant la *transcription* et la *traduction* indépendemment de l'environnement** (fig. 1). Il semble que les interactions entre les différents gènes influencent l'expression de ceux-ci, on parle de **réseau de régulation de gènes** (fig. 1.3). Parmi les interactions possibles au cours du temps, **un gène peut en activer un autre, ou bien l'éteindre, et les actions**

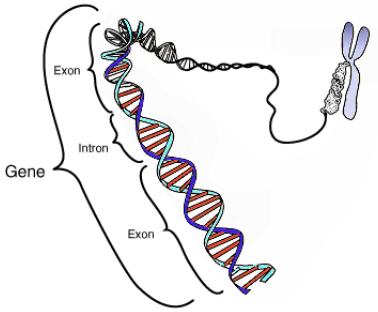


FIGURE 1.1 – *Un gène, portion de l'ADN codant pour une protéine*

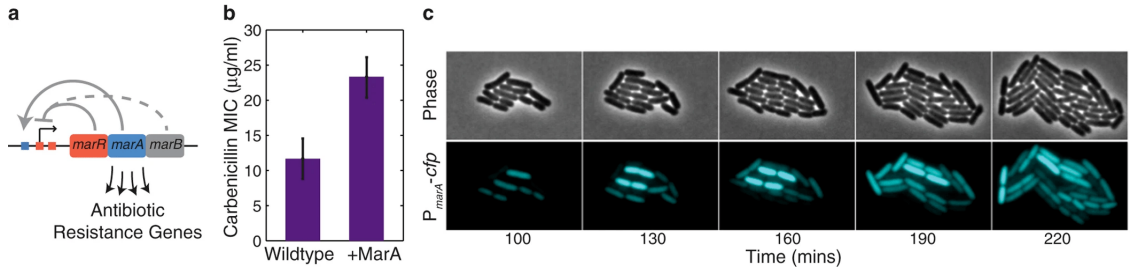


FIGURE 1.2 – [Imane El Meouche, 2016]. *Les gènes ne s'expriment pas tous de la même manière à chaque instant car les niveaux de fluorescence ne sont pas les mêmes.*

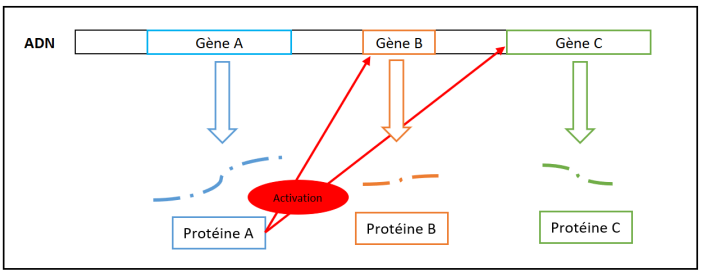


FIGURE 1.3 – *Réseau de régulation de gènes - Le gène A s'exprime pour créer un certain niveau de protéines (on les note « protéines A »). Les protéines A vont activer les gènes B et C au sein de la même cellule, qui vont à leur tour produire des protéines. Si le gène A s'éteint, alors il va arrêter de s'exprimer et va par la suite éteindre les gènes B et C par manque de protéines A.*

peuvent être à sens unique ou à double sens. Afin de formalismer mathématiquement ces interactions, nous allons les représenter par des **graphes**.

1.2 Le formalisme mathématique

Nous nous plaçons pour l'instant au sein d'une cellule, et nous modélisons son réseau de gènes par un graphe, où les noeuds sont les gènes, et les arrêtes orientées sont les interactions. **Les noeuds ne peuvent prendre que deux valeurs : 0 quand le gène ne s'exprime pas (inactif), et 1 quand le gène s'exprime (actif)** (fig. 1.4). Toute la difficulté réside dans le fait qu'on ne connaît pas les interactions (les arrêtes orientés) entre les gènes (les noeuds).

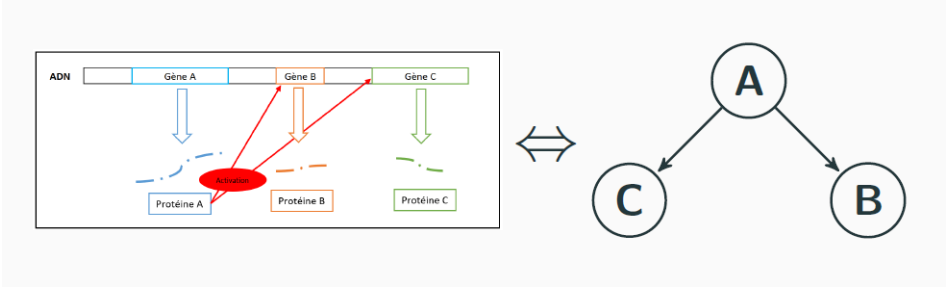


FIGURE 1.4 – *Modélisation d'un réseau de gènes*

Temps	Gène A	Gène B	Gène C
0	0	0	0
0	1	0	0
0	0	0	0
...	...	...	...
1	1	0	1
1	1	1	1
1	1	1	1
...	...	...	...
2	1	0	1
2	0	1	1
2	1	0	0
...	...	...	...

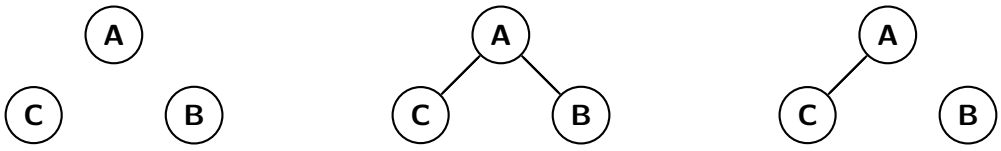
TABLE 1.1 – *Jeu de cellules pour notre exemple avec les 3 gènes A, B, C (une ligne = une cellule). Ici T vaut 3 (mesures aux temps 0, 1 et 2)*

De quelles informations dispose-t-on pour déduire le graphe ?

Nous supposerons dans la suite que nous disposons d'un jeu de cellules que l'on a « photographié » à T instants. Pour chaque instant nous connaissons l'état des gènes étudiés pour les cellules « photographiées ».

Remarque 1.1. La mesure sur une cellule détruit la cellule. C'est pourquoi la cellule donnant la première mesure au temps 0 est différente de celle qui a donné la première mesure au temps 1.

À la vue des données, il sera plus simple pour la suite de déduire le graphe orienté d'une autre façon. **Nous allons déduire un graphe non-orienté à chacun des instants de mesure**. Une suite de graphes représentative possible de notre même exemple serait la suivante :



$t = 0$ $t = 1$ $t = 2$

Ainsi il faut, à partir des données, déduire une suite de graphes dynamiques non-orientés, pour ensuite reconstruire un graphe orienté représentant les interactions entre les différents gènes (fig. 1.5). Pour parvenir au résultat escompté, une approche possible est de faire de l'inférence bayésienne de graphes.

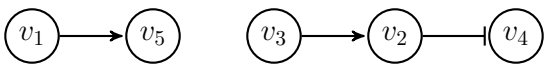


FIGURE 1.5 – *Exemple d'un réseau de 5 gènes à retrouver (notés de v_1 à v_5). L'arrête avec une extrémité fermée correspond à une inhibition (une fois activé, v_2 va éteindre v_4)*

Chapitre 2

Inférence bayésienne exacte de graphes

2.1 Quelques rappels sur l'inférence bayésienne

Faire de l'inférence bayésienne, c'est se fonder sur un **a priori** et prendre en compte des **observations** pour obtenir la meilleure estimation possible d'un (ou plusieurs) ***paramètre(s) caché(s)***. Cette méthode se base principalement sur le Théorème de Bayes.

Théorème 2.1 (Théorème fondamental de Bayes)**.** *Soit $\boldsymbol{\theta}$ une variable aléatoire à valeur dans un espace Θ représentant le* ***paramètre caché*** *$\mathbb{P}_0$ une probabilité a priori (prior) sur $\boldsymbol{\theta}$, et $\mathcal{D}$ une observation :*

$$\mathbb{P}(\boldsymbol{\theta} = \theta \mid \mathcal{D}) = \frac{\mathbb{P}_0(\boldsymbol{\theta} = \theta)\mathbb{P}(\mathcal{D} \mid \boldsymbol{\theta} = \theta)}{\mathbb{P}(\mathcal{D})}, \quad \forall \theta \in \Theta$$

On appelle ***probabilité a posteriori*** *la probabilité $\mathbb{P}(\boldsymbol{\theta} = \theta \mid \mathcal{D})$*

D'une part, les données jouent un rôle crucial dans l'inférence. On voit que pour une prior $\mathbb{P}_0$ fixée, les données vont apporter de l'information et corriger la probabilité d'un évenement à posteriori. D'autre part, le choix de la prior joue aussi un rôle important et représente une part de subjectivité dans le calcul. Enfin $\mathbb{P}(\mathcal{D})$ est appelée la *constante de normalisation* et peut être difficile à calculer dans certains contextes.

Cette philosophie bayésienne est le pillier sur lequel se fonde toutes les recherches réalisées. Nous allons voir comment elle peut être mise en œuvre pour de l'inférence de graphes.

2.2 Comment faire de l'inférence de graphes ?

Mes travaux de recherches s'appuient principalement sur la publication *Tractable Bayesian learning of tree belief networks* de [Meilă and Jaakkola, 2006], car elle donne les fondations théoriques recherchées pour mon problème. J'ai dans un premier temps analysé et retrouvé les résultats de leurs recherches pour, dans un second temps, proposer plusieurs améliorations avec l'aide de mon encadrant.

L'hypothèse de départ de [Meilă and Jaakkola, 2006] est que les graphes d'interaction sont des *arbres*.

Définition 2.2 (Arbre)**.** On appelle *arbre* $(\mathcal{V}, \mathcal{E})$ un graphe acyclique et connexe où $\mathcal{V} = \{v_1, \dots, v_n\}$ (qu'on notera aussi $\mathcal{V} = \{1, \dots, n\}$) est l'ensemble des noeuds, et $\mathcal{E}$ est sa *structure*. L'ensemble de ses arrêtes est sous la forme de couples $\{v_i, v_j\}$.

Pour inférer un graphe, il faut le traduire mathématiquement. Ainsi on lui associe une loi de probabilité pour calculer des probabilités d'états du réseau. On note T une probabilité sur un arbre (fig. 2.1).

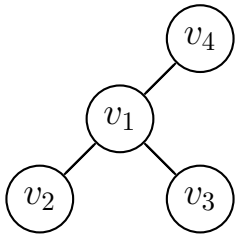


FIGURE 2.1 – *Un arbre à 4 sommets* $\mathcal{V} = \{v_1, \dots, v_4\}$ *et* $T(x) = T(x_{v_1}, x_{v_2}, x_{v_3}, x_{v_4})$ *une probabilité sur cet arbre* ($x = (0, 1, 0, 0)$ *par exemple).*

On suppose également que toutes les probabilités sur les arbres peuvent être *factorisées* ou sont *factorisables*.

Définition 2.3 (Probabilité factorisable selon un arbre)**.** Soit $(\mathcal{V},\mathcal{E})$ un arbre et T une probabilité sur celui-ci. La probabilité est dite *factorisable* si elle peut se mettre sous la forme suivante

$$T(x) = \prod_{v \in \mathcal{V}} T_v(x_v) \prod_{uv \in \mathcal{E}} \frac{T_{uv}(x_u, x_v)}{T_u(x_u) T_v(x_v)}$$

- T_v, T_{uv} les marginales respectives du noeud v et de l'arrête uv
- $x = (x_{v_1}, x_{v_2}, ..., x_{v_n})$ une observation où $x_{v_i} \in \{0, 1\}$

Cette forme traduit les liens de dépendance entre les nœuds. Conditionnellement à ses voisins, un nœud est indépendant de tous les autres. Le graphe un *champs de Markov*. On comprend mieux ces popriétés d'indépendance en écrivant la probabilité factorisée sous une autre forme : une *probabilité factorisée orientée*.

Définition 2.4 (Probabilité factorisée orientée selon un arbre)**.** Soit $(\mathcal{V},\mathcal{E})$ un arbre et T une probabilité *factorisable*. Après avoir choisit une *orientation* de cet arbre, la forme *factorisée orientée* de cette probabilité est de la suivante :

$$T(x) = \prod_{v \in \mathcal{V}} T_{v|pa(v)}(x_v \mid x_{pa(v)})$$

- $T_{v|pa(v)}$ la marginale de v selon son parent (unique !)
- $x = (x_{v_1}, x_{v_2}, ..., x_{v_n})$ une observation où $x_{v_i} \in \{0, 1\}$

On appelle *orientation d'un arbre* le fait de choisir un sommet originel (appelé *racine*) et d'orienter les arrêtes depuis ce sommet. Il existe donc autant d'orientations possible que de sommets. On note $(\mathcal{V}, \bar{\mathcal{E}})$ l'arbre orienté ainsi obtenu. Par exemple si T est une probabilité factorisée selon l'arbre de la (fig. 2.2), alors la forme factorisée et la forme factorisée orientée sont les suivantes :

$$T = \cancel{T_{v_1}} \cancel{T_{v_2}} \cancel{T_{v_3}} T_{v_4} \frac{T_{v_1 v_2}}{T_{v_1} \cancel{T_{v_2}}} \frac{T_{v_1 v_3}}{T_{v_1} \cancel{T_{v_3}}} \frac{T_{v_1 v_4}}{\cancel{T_{v_1}} T_{v_4}} = T_{v_4} T_{v_1|v_4} T_{v_2|v_1} T_{v_3|v_1}$$

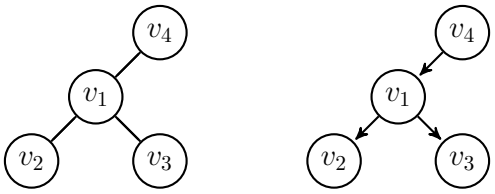


FIGURE 2.2 – *Exemple d'arbre* $(\mathcal{V}, \mathcal{E})$ *et une de ses formes orientées* $(\mathcal{V}, \overline{\mathcal{E}})$.

où l'orientation est faite à partir du sommet 4.

Remarque 2.5. L'orientation de l'arbre n'a rien à voir avec les liens génétiques d'activation. C'est une convention mathématique.

Remarque 2.6. On peut confondre ce qu'on appelle un *arbre* et sa loi T car l'un se déduit de l'autre dans leur forme.

Toutes les hypothèses et définitions ci-dessus sont celles utilisées dans la publication [Meilă and Jaakkola, 2006], mais elles peuvent être améliorées pour obtenir des résultats plus généraux. Pour ce qui suit, on supposera que les graphes appartiennent à une plus grande classe : *les forêts*, dont on note F la probabilité associée. Tous ce qui suit sera une généralisation des résultats de [Meilă and Jaakkola, 2006].

Définition 2.7 (Forêt)**.** On appelle *forêt* $(\mathcal{V}, \mathcal{E})$ un graphe acyclique (pas forcément connexe). Ainsi $\mathcal{V}$ ne change pas par rapport à un arbre, mais la *structure* $\mathcal{E}$ si. On notera $\mathcal{F}_n$ l'ensemble des *structures de forêts non-orientées* et $\overline{\mathcal{F}_n}$ l'ensemble des *structures de forêts orientées*.

Remarque 2.8. La définition de *probabilité factorisable selon un arbre* reste valable pour une forêt. La définition de *probabilité factorisée orientée selon un arbre* s'adapte aux forêts en choisissant une racine par composante connexe.

⇒ **Inférer une forêt, c'est inférer l'objet complexe qu'est la probabilité qui la représente :** F

2.3 Théorème de Bayes et inférence

Dans notre contexte, on veut inférer une forêt à un temps t fixé (on connaît le nombre de sommets). En notant N le nombre d'observations à t fixé (on supposera N indépendant du temps), on obtient d'après le Théorème de Bayes 2.1

$$\mathbb{P}(F \mid \mathcal{D}) = \frac{\mathbb{P}_0(F)\mathbb{P}(\mathcal{D} \mid F)}{\mathbb{P}(\mathcal{D})} = \frac{\mathbb{P}_0(F)\prod_{k=1}^{N} F(x^{(k)})}{\mathbb{P}(\mathcal{D})}$$

- $\mathbb{P}_0$: prior sur la forêt F
- N : nombre d'observations à t fixé
- $\mathcal{D} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$: les observations ***indépendantes*** où on rapelle que $x^{(i)}$ est une ligne du tableau 1.1 pour un même temps t.

2.3.1 La prior

Quelle prior $\mathbb{P}_0$ choisir pour F ?

Tirer une forêt F, c'est d'abord tirer sa structure $\mathcal{E}$ (quelles arrêtes entre quels nœuds ?), puis tirer les paramètres de la probabilité de F (que vaut $F(x_{v_1}, x_{v_2}, x_{v_3}, x_{v_4})$?). Il faut donc une prior pour la structure, et une prior sur l'ensemble des paramètres d'une probabilité (ce qui s'apparente à une probabilité de probabilité).

Remarque 2.9. Pour des raisons de praticité pour l'écriture des formules mathématiques, **nous tirerons des structures orientées $\overline{\mathcal{E}}$.**

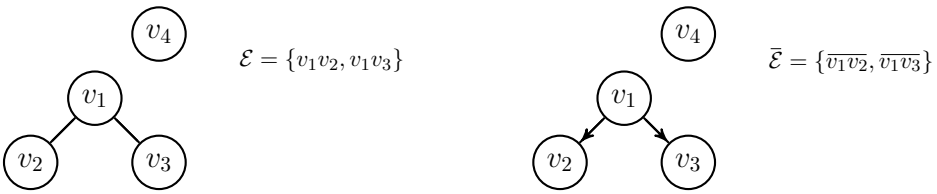
Remarque 2.10. Trouver une prior sur un ensemble de probabilités semble compliqué. Cependant, sachant qu'on tire la structure de l'arbre en premier et qu'on a fait l'hypothèse que les probabilités se factorisent en les marginales de taille 1 (F_v) et 2 ($F_{v|pa(v)}$), il suffit de connaître les paramètres $\{F_v(0), F_v(1)\}$ et $\{F_{v|pa(v)}(0 \mid 0), F_{v|pa(v)}(0 \mid 1), F_{v|pa(v)}(1 \mid 0), F_{v|pa(v)}(1 \mid 1)$ pour connaître F !

En résumé, tirer une forêt c'est tirer les deux objets suivants successivement,

- $\overline{\mathcal{E}} \in \overline{\mathcal{F}_n}$ une structure de forêt orientée à n sommets.
- Sachant $\overline{\mathcal{E}}$, $\Theta_{\overline{\mathcal{E}}} = (\theta_{v|pa(v)})_{v\in\mathcal{V}}$ les paramètres de la forêt tel que,
  $\forall v \in \mathcal{V}, \theta_{v|pa(v)} = F_{v|pa(v)}$ si v n'est pas racine.
  $\forall v \in \mathcal{V}, \theta_{v|pa(v)} = \theta_v = F_v$ si v est racine.

On obtient donc $\mathbb{P}_0(F) = \mathbb{P}_0(\overline{\mathcal{E}})\mathbb{P}_0(\Theta_{\overline{\mathcal{E}}} \mid \overline{\mathcal{E}})$

Exemple 2.11. On suppose pour l'instant qu'on a un moyen de tirer la structure $\overline{\mathcal{E}}$ aléatoirement et qu'on obtient ici



Sachant cette structure, il faut alors tirer les paramètres $\Theta_{\mathcal{E}} = \{\theta_{v_1}, \theta_{v_2}, \theta_{v_3}, \theta_{v_4}, \theta_{v_1v_2}, \theta_{v_1v_3}\}$, c'est-à-dire les 16 valeurs suivantes

$$\begin{array}{ll} \{\theta_{v_1}(0); \theta_{v_1}(1)\} & \{\theta_{v_1v_2}(00); \theta_{v_1v_2}(01); \theta_{v_1v_2}(10); \theta_{v_1v_2}(11)\} \\ \{\theta_{v_2}(0); \theta_{v_2}(1)\} & \{\theta_{v_1v_3}(00); \theta_{v_1v_3}(01); \theta_{v_1v_3}(10); \theta_{v_1v_3}(11)\} \\ \{\theta_{v_3}(0); \theta_{v_3}(1)\} & \\ \{\theta_{v_4}(0); \theta_{v_4}(1)\} & \end{array}$$

Ou bien il faut tirer les paramètres $\Theta_{\overline{\mathcal{E}}} = \{\theta_{v_4}, \theta_{v_1}, \theta_{v_2|v_1}, \theta_{v_3|v_1}\}$, c'est-à-dire les 12 valeurs suivantes

$$\begin{array}{ll} \{\theta_{v_1}(0); \theta_{v_1}(1)\} & \{\theta_{v_2|v_1}(0|0); \theta_{v_2|v_1}(0|1); \theta_{v_2|v_1}(1|0); \theta_{v_2|v_1}(1|1)\} \\ \{\theta_{v_4}(0); \theta_{v_4}(1)\} & \{\theta_{v_3|v_1}(0|0); \theta_{v_3|v_1}(0|1); \theta_{v_3|v_1}(1|0); \theta_{v_3|v_1}(1|1)\} \end{array}$$

Et alors on connaît F. Par exemple,

$$F(0,1,1,1) = F_{v_4}(1)F_{v_1}(0)F_{v_2|v_1}(1|0)F_{v_3|v_1}(1|0) = \theta_{v_4}(1)\theta_{v_1}(0)\theta_{v_2|v_1}(1|0)\theta_{v_3|v_1}(1|0).$$

Remarque 2.12. Il faut que $\mathcal{E}$ respecte une certaine structure (pas de cycle) et que les paramètres θ_{uv} et θ_u respectent certaines relations pour repésenter des probabilités. Nous verrons comment les priors sont choisies pour respecter ces contraintes.

Remarque 2.13. Pour la suite nous ne noterons plus la barre au dessus de *uv* pour signifier son orientation. Ainsi quand nous noterons $uv \in \mathcal{E}$, alors uv est orienté dans le sens $\overline{uv}$ (de u vers v).

Quelle prior $\mathbb{P}_0(\overline{\mathcal{E}})$?

[Meilă and Jaakkola, 2006] suggère de prendre la prior suivante.

Définition 2.14 (Prior décomposable sur $\overline{\mathcal{F}_n}$)**.** On définit les priors *décomposables* sur $\overline{\mathcal{F}_n}$ telles que,

$$\forall \bar{\mathcal{E}} \in \overline{\mathcal{F}}_n,\ \mathbb{P}_0(\bar{\mathcal{E}}) = \frac{1}{A} \prod_{uv \in \bar{\mathcal{E}}} \beta_{uv}$$

où les *hyperparamètres* sont les β_{uv} tels que, $\beta_{uv} = \beta_{vu} \geq 0$ et $\beta_{vv} = 0$ pour tout $u, v \in \mathcal{V}$ et le produit vide est égal à 1.

Remarque 2.15. β_{uv} peut être vu comme le « poids » de l'arrête $uv \in \bar{\mathcal{E}}$

Remarque 2.16. On peut calculer $\mathbb{P}_0(\mathcal{E})$ en sommant toutes les probabilités de forêts orientées $\mathbb{P}_0(\bar{\mathcal{E}})$ correspondantes.

A priori, la constante de normalisation requiert une sommation sur $|\overline{\mathcal{F}_n}| \geqslant |\mathcal{F}_n| = (n+1)^{n-1}$ structures, ce qui est énorme. On peut tout de même calculer cette constante grâce au théorème suivant.

Théorème 2.17 (Calcul de la constante de normalisation d'une prior décomposable sur $\overline{\mathcal{F}_n}$)**.** *Soit $\mathbb{P}_0$ une prior décomposable sur $\overline{\mathcal{F}_n}$ de constante de normalisation A. On définit la matrice $\mathcal{Q}(\beta)$ par :*

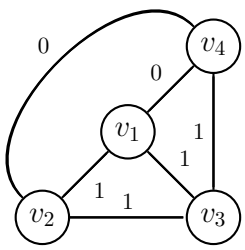
$$\mathcal{Q}_{uv}(\beta) = \mathcal{Q}_{vu}(\beta) = \begin{cases} \beta_{uv} & 1 \leq u < v \leq n \\ \sum_{v'=1}^{n} \beta_{v'v} & 1 \leq u = v \leq n \end{cases}$$

Alors $A = \det(I_n + \mathcal{Q}(\beta))$.

Démonstration. Il s'agit du *matrix-forest theorem* [Chebotarev and Agaev, 2002], une variante du *matrix-tree theorem*, lui-même une généralisation du théorème de Kirchhoff. □

Remarque 2.18. Ce théorème a été démontré par [Meilă and Jaakkola, 2006] mais pour $\mathcal{T}_n$ l'ensemble des arbres à n sommets. Dans ce cas on auarit $A = \det(\mathcal{Q}(\beta))$.

Exemple 2.19. On définit la prior décomposable sur $\overline{\mathcal{F}_4}$ telle que $\beta_{12} = \beta_{13} = \beta_{14} = \beta_{23} = 1$ et $\beta_{uv} = 0$ sinon :



Alors d'après le théorème 2.17 on a :

$$\mathcal{Q}_{uv}(\beta) = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \quad donc \quad \boxed{A = \det(I_4 + \mathcal{Q}_{uv}(\beta)) = 40}$$

Parmi les $|\mathcal{F}_4| = 125$ forêts, il y en a 40 possibles qui sont équiprobablement tirées par cette prior.

Quelle prior $\mathbb{P}_0(\Theta_{\bar{\mathcal{E}}}|\bar{\mathcal{E}})$?

En connaissance de $\bar{\mathcal{E}}$, il faut tirer une distribution de probabilité $\Theta_{\bar{\mathcal{E}}}$ aléatoirement (exemple 2.11). Une bonne manière de procéder est de passer par les *lois de Dirichlet*.

Définition 2.20 (d-simplexe)**.** Soit $d \in \mathbb{N}$. On définit un le *d-simplexe* de la manière suivante,

$$\Delta_{d-1} = \{(\theta_1, \theta_2, ..., \theta_d) \in \mathbb{R}^d \mid \theta_i > 0 \; ; \; \sum_{i=1}^{d} \theta_i = 1\}$$

Tirer un point dans un *d-simplexe* (fig. 2.3) permet de déterminer les poids d'une loi aléatoire discrète prenant d valeurs. En pondérant les possibilités grâce à des paramètres α, on comprend mieux l'utilité des *lois de Dirichlet*.

Définition 2.21 (Loi de Dirichlet)**.** Soit $d \in \mathbb{N}$, $(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, .., \boldsymbol{\theta_d})$ une variable aléatoire suivant une loi de Dirichlet à valeur dans le *d-simplexe* et de paramètre $\alpha = (\alpha_1, \alpha_2, ..., \alpha_d) \in \mathbb{R}_+^d$. En notant D la fonction de probabilité associée on a

$$D(\theta_1, \theta_2, ..., \theta_d \; ; \; \alpha_1, \alpha_2, ..., \alpha_d) = \frac{1}{B(\alpha)} \prod_{i=1}^{d} \theta_i^{\alpha_i - 1}$$

où la constante de normalisation $B(\alpha)$ dépend de la fonction Γ d'Euler et vaut

$$B(\alpha) = \frac{\prod_{i=1}^{d} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{d} \alpha_i)} \quad \text{avec} \quad \Gamma(x) = \int_0^{\infty} s^{x-1} e^{-s} ds.$$

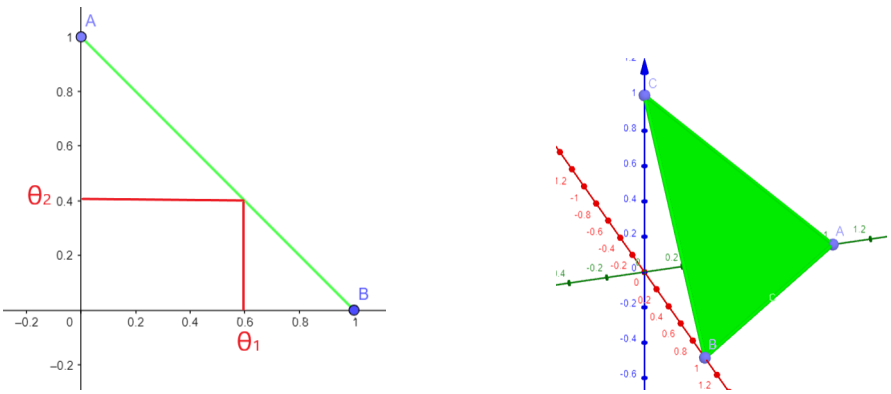


FIGURE 2.3 – *Représentation du 2-simplexe (à droite) et du 3-simplexe (à gauche)*

Remarque 2.22. Les hyperparamètres α_i pondèrent la distribution de manière assez intuitive (fig. 2.4). Pour $d = 2$ et $\alpha_1 = \alpha_2$ proche de 0, la distribution tend vers un Dirac en 0 ou en 1. S'ils sont grands et égaux alors on est presque sûr d'avoir $\theta_1 = \theta_2$. À l'inverse on observe une dissymétrie quand $\alpha_1 \neq \alpha_2$ avec un poids α_i grand qui favorise un θ_i proche de 1.

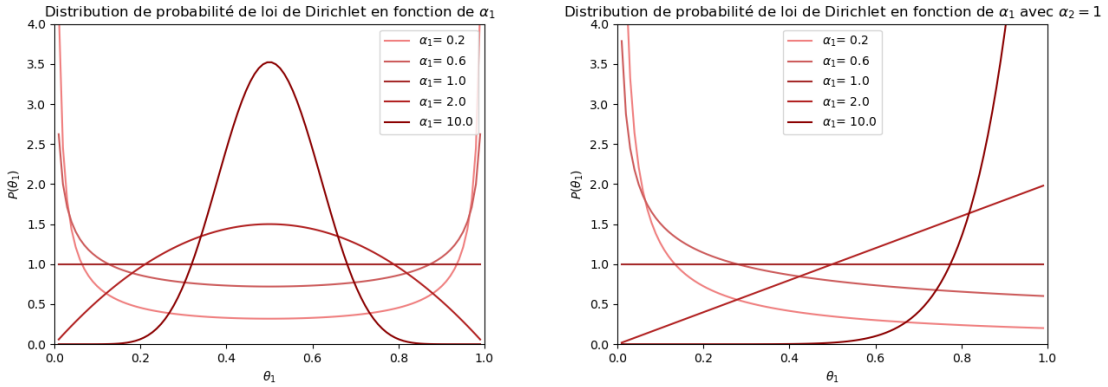


FIGURE 2.4 – *à gauche* $\alpha_1 = \alpha_2$ *et* $D(\theta_1, \theta_2; \alpha_1, \alpha_1) = (\theta_1(1-\theta_1))^{\alpha_1 - 1}$ *et à droite* $\alpha_1 \neq \alpha_2$ *et* $D(\theta_1; \alpha_1, \alpha_2) = \theta_1^{\alpha_1 - 1}(1-\theta_1)^{\alpha_2 - 1}$

Ainsi pour $d = 2$ on peut tirer les paramètres d'une loi suivie par $\boldsymbol{\theta_u}$ ou $\boldsymbol{\theta_{v|pa(v)}}$, et pour $d = 4$ celle d'une loi suivie par $\boldsymbol{\theta_{uv}}$.

Sous certaines hypothèses, [Meilă and Jaakkola, 2006] montrent qu'une prior idéale sur $\Theta_{\bar{\mathcal{E}}}$ serait de la forme suivante.

Définition 2.23 (Prior décomposable sur $\Theta_{\bar{\mathcal{E}}}$)**.** Soit $\bar{\mathcal{E}}$ une structure de forêt orientée. On définit la *prior décomposable sur* $\Theta_{\bar{\mathcal{E}}}$ d'*hyperparamètres* N'_{uv} de la manière suivante :

$$\mathbb{P}_0(\Theta_{\bar{\mathcal{E}}} \mid \bar{\mathcal{E}}) = \prod_{v \in \mathcal{V}} \mathbb{P}_0(\theta_{v|pa(v)})$$

où

$$\mathbb{P}_0(\theta_{v|u}) := D(\theta_{v|u}(0|0), \theta_{v|u}(1|0); N'_{vu}(00), N'_{vu}(10)) \times D(\theta_{v|u}(0|1), \theta_{v|u}(1|1); N'_{vu}(01), N'_{vu}(11))$$

et les *hyperparamètres* $N_{uv}(i,j)$ vérifient :

- $\forall u, v \in \mathcal{V},\ \forall i, j \in \{0,1\},\ N'_{uv}(ij) = N'_{vu}(ji) > 0$
- $\forall u, v \in \mathcal{V},\ \forall j \in \{0,1\},\ N'_{uv}(0j) + N'_{uv}(1j) = N'_v(j)$
- $\forall u \in \mathcal{V},\ N'_u(0) + N'_u(1) =: N'$

Remarque 2.24. On note que $\mathbb{P}_0(\theta_{v|pa(v)}) = \mathbb{P}_0(\theta_v) = D(\theta_v(0), \theta_v(1); N'_v(0), N'_v(1))$ si v est racine de sa composante connexe.

Remarque 2.25. Cette prior fait apparaître les indépendances dans la forme $\bar{\mathcal{E}}$ considérée et prend en compte tous les paramètres associés à cette structure.

Remarque 2.26. Ici les nœuds ne prennent que deux valeurs 0 et 1, donc les lois de Dirichlet sont en réalité des lois Béta. Cependant le modèle est généralisable à un graphe prenant plusieurs valeurs 2,3,4...,r.

2.3.2 La vraisemblance

La vraisemblance vaut

$$\mathbb{P}(\mathcal{D} \mid F) = \mathbb{P}(x^{(1)}, x^{(2)}, ..., x^{(N)}) \mid \Theta_{\bar{\mathcal{E}}}, \bar{\mathcal{E}}) = \prod_{k=1}^{N} F(x^{(k)}) = \prod_{k=1}^{N} \prod_{v \in \mathcal{V}} \theta_{v|pa(v)}(x_v^{(k)} \mid x_{pa(v)}^{(k)})$$

2.3.3 Calcul du posterior, un posterior conjugué à la prior

Les hypotèses faites sur les priors vont avoir des conséquences formidables sur la forme de la posterior $\mathbb{P}(F \mid \mathcal{D})$. Cette dernière sera de la « même forme » que la prior. On dit que la prior et la posterior sont *conjuguées*.

Définition 2.27 (Prior conjuguée / Prior et Posterior conjuguées)**.** Dans le cas où la posterior appartient à la même famille de loi de probabilité que la prior, la prior et la posterior sont dites **conjuguées**.

Théorème 2.28 (Inférence exacte)**.** *Suposons que les priors sur $\Theta_{\bar{\mathcal{E}}}$ et $\bar{\mathcal{E}}$ soient **décomposables**, alors les priors et les posteriors sont conjuguées et on a*

$$\mathbb{P}(\Theta_{\bar{\mathcal{E}}} \mid \mathcal{D}) = \prod_{v \in \mathcal{V}} \prod_{i=0,1} D(\theta_{v|pa(v)}(. \mid i); N''_{vu}(.i))$$

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{\det(I_n + \mathcal{Q}(\beta W))} \prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} W_{uv}$$

où les nouveaux hyperparamètres sont :

- $N''_{uv}(ij) = N'_{uv}(ij) + N_{uv}(ij)$
- $\beta_{uv} W_{uv}$ *tel que* $W_{uv} = \frac{1}{W} \frac{1}{W_u W_v} \prod_{i,j=0,1} \frac{\Gamma(N''_{uv}(ij))}{\Gamma(N'_{uv}(ij))}$ *avec* $W_u = \frac{1}{W} \prod_{i=0,1} \frac{\Gamma(N''_u(i))}{\Gamma(N'_u(i))}$

Démonstration. Développons la posterior totale $\mathbb{P}(F \mid \mathcal{D})$:

$$\mathbb{P}(F \mid \mathcal{D}) = \mathbb{P}(\bar{\mathcal{E}}, \Theta_{\bar{\mathcal{E}}} \mid \mathcal{D}) = \frac{1}{K} . \underbrace{\mathbb{P}(\mathcal{D} \mid \Theta_{\bar{\mathcal{E}}})}_{1)} . \underbrace{\mathbb{P}_0(\Theta_{\bar{\mathcal{E}}} \mid \bar{\mathcal{E}})}_{2)} . \underbrace{\mathbb{P}_0(\bar{\mathcal{E}})}_{3)}$$

Calcul de 1)

$$\mathbb{P}(\mathcal{D} \mid \Theta_{\bar{\mathcal{E}}}) = \prod_{k=1}^{N} \mathbb{P}(x^{(k)} \mid \Theta_{\bar{\mathcal{E}}}) = \prod_{k=1}^{N} \prod_{v \in \mathcal{V}} \theta_{v|pa(v)}(x_v^{(k)} \mid x_{pa(v)}^{(k)})$$

On sépare les nœuds qui sont des racines (R) de ceux qui sont des fils (C) de façon à ce que $\mathcal{V} = R \sqcup C$. Alors,

$$\mathbb{P}(\mathcal{D} \mid \Theta_{\bar{\mathcal{E}}}) = \prod_{k=1}^{N} \left(\prod_{v \in R} \theta_v(x_v^{(k)}) \right) \left(\prod_{v \in C} \theta_{v|pa(v)}(x_v^{(k)} \mid x_{pa(v)}^{(k)}) \right)$$

On regroupe les θ_v et les $\theta_{v|pa(v)}$ en fonction de leur(s) argument(s). On note en exposant

par $N_v(i)$ le nombre de $\theta_v(i)$, et par $N_{vpa(v)}(ij)$ le nombre de $\theta_{v|pa(v)}(i \mid j)$.

$$\mathbb{P}(\mathcal{D} \mid \Theta_{\bar{\mathcal{E}}}) = \left(\prod_{v \in R} \prod_{i=0,1} \theta_v(i)^{N_v(i)} \right) \left(\prod_{v \in C} \prod_{i,j \in \{0,1\}} \theta_{v|pa(v)}(i \mid j)^{N_{vpa(v)}(ij)} \right)$$

Calcul de 2)
Découle de la définition de la prior sur $\Theta_{\bar{\mathcal{E}}}$,

$$\begin{aligned}
\mathbb{P}_0(\Theta_{\bar{\mathcal{E}}} \mid \bar{\mathcal{E}}) &= \left(\prod_{v \in R} \mathcal{D}(\theta_v(.) \mid N'_v(.)) \right) \left(\prod_{v \in C} \prod_{j \in \{0,1\}} \mathcal{D}(\theta_{v|pa(v)}(. \mid j), N'_{vpa(v)}(.j) \right) \\
&= \left(\prod_{v \in R} \frac{\Gamma(N')}{\prod_{i \in \{0,1\}} \Gamma(N'_v(i))} \prod_{i \in \{0,1\}} \theta_v(i)^{N'_v(i)-1} \right) \times \\
&\left(\prod_{v \in C} \prod_{j \in \{0,1\}} \frac{\Gamma(N'_{pa(v)}(j))}{\prod_{i \in \{0,1\}} \Gamma(N'_{vpa(v)}(ij))} \prod_{i \in \{0,1\}} \theta_{v|pa(v)}(i \mid j)^{N'_{vpa(v)}(ij)-1} \right)
\end{aligned}$$

On commence à voir comment 1) et 2) peuvent être regroupés !

Calcul de 3)
Découle de la définition de la prior sur $\bar{\mathcal{E}}$,

$$\mathbb{P}_0(\bar{\mathcal{E}}) = \frac{1}{A(\beta)} \prod_{uv \in \bar{\mathcal{E}}} \beta_{uv}$$

On regroupe tout
On regroupe 1) et 2) pour faire apparaître une nouvelle loi de Dirichlet.

$$\begin{aligned}
\mathbb{P}(F \mid \mathcal{D}) = \frac{1}{K.A(\beta)} &\left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \right) \left(\prod_{v \in R} \frac{\Gamma(N') \prod_{i \in \{0,1\}} \Gamma(N''_v(i))}{\Gamma(N'') \prod_{i \in \{0,1\}} \Gamma(N'_v(i))} \mathcal{D}(\theta_v(.), N''_v(.)) \right) \\
&\left(\prod_{v \in C} \prod_{j \in \{0,1\}} \frac{\Gamma(N'_{pa(v)}(j)) \prod_{i \in \{0,1\}} \Gamma(N''_{vpa(v)}(ij))}{\Gamma(N''_{pa(v)}(j)) \prod_{i \in \{0,1\}} \Gamma(N'_{vpa(v)}(ij))} \mathcal{D}(\theta_{v|pa(v)}(. \mid j), N''_{v|pa(v)}(.j)) \right)
\end{aligned}$$

On regroupe les facteurs.

$$\begin{aligned}
\mathbb{P}(F \mid \mathcal{D}) = \frac{1}{K.A(\beta)} &\left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \right) \left(\prod_{v \in R} \frac{\Gamma(N')}{\Gamma(N'')} \prod_{i \in \{0,1\}} \frac{\Gamma(N''_v(i))}{\Gamma(N'_v(i))} \mathcal{D}(\theta_v(.), N''_v(.)) \right) \\
&\left(\prod_{v \in C} \prod_{i \in \{0,1\}} \frac{\Gamma(N'_{pa(v)}(i))}{\Gamma(N''_{pa(v)}(i))} \prod_{i,j \in \{0,1\}} \frac{\Gamma(N''_{vpa(v)}(ij))}{\Gamma(N'_{vpa(v)}(ij))} \mathcal{D}(\theta_{v|pa(v)}(. \mid j), N''_{v|pa(v)}(.j)) \right)
\end{aligned}$$

Maintenant on va sommer sur tous les θ possibles pour obtenir la posterior $\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D})$

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \int_\theta \mathbb{P}(F \mid \mathcal{D}) d\theta$$

Intégrer les loi de Dirichlet fait 1, et les orientations $vpa(v)$, qui sont plutôt $pa(v)v$ si on les oriente, peuvent être remplacer par uv en changeant les indices (u devient le père).

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{K.A(\beta)} \left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \right) \left(\prod_{v \in R} \frac{\Gamma(N')}{\Gamma(N'')} \prod_{i \in \{0,1\}} \frac{\Gamma(N''_v(i))}{\Gamma(N'_v(i))} \right) \left(\prod_{v \in C} \prod_{i \in \{0,1\}} \frac{\Gamma(N'_{pa(v)}(i))}{\Gamma(N''_{pa(v)}(i))} \prod_{i,j \in \{0,1\}} \frac{\Gamma(N''_{vpa(v)}(ij))}{\Gamma(N'_{vpa(v)}(ij))} \right)$$

Puis on regroupe le produit sur les noeuds et le produit sur les arrêtes.

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{K.A(\beta)} \left(\prod_{v \in R} \frac{\Gamma(N')}{\Gamma(N'')} \prod_{i \in \{0,1\}} \frac{\Gamma(N''_v(i))}{\Gamma(N'_v(i))} \right) \left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \prod_{i \in \{0,1\}} \frac{\Gamma(N'_u(i))}{\Gamma(N''_u(i))} \prod_{i,j \in \{0,1\}} \frac{\Gamma(N''_{uv}(ij))}{\Gamma(N'_{uv}(ij))} \right)$$

Pose $W = \dfrac{\Gamma(N'')}{\Gamma(N')}$, $W_u = \dfrac{1}{W} \prod_{i \in \{0,1\}} \dfrac{\Gamma(N''_v(i))}{\Gamma(N'_v(i))}$, $W_{uv} = \dfrac{1}{W} \dfrac{1}{W_u W_v} \prod_{i,j \in \{0,1\}} \dfrac{\Gamma(N''_u v(ij))}{\Gamma(N'_u v(ij))}$, et on complète le produit sur R par $\mathcal{V}$, l'ensemble des nœuds, en compensant dans le produit sur $\bar{\mathcal{E}}$

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{K.A(\beta)} \left(\prod_{v \in \mathcal{V}} \frac{1}{W} \prod_{i \in \{0,1\}} \frac{\Gamma(N''_v(i))}{\Gamma(N'_v(i))} \right) \left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \, W \prod_{i \in \{0,1\}} \frac{\Gamma(N'_u(i))}{\Gamma(N''_u(i))} \prod_{i \in \{0,1\}} \frac{\Gamma(N'_v(i))}{\Gamma(N''_v(i))} \prod_{i,j \in \{0,1\}} \frac{\Gamma(N''_{uv}(ij))}{\Gamma(N'_{uv}(ij))} \right)$$

Et donc enfin,

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{K.A(\beta)} \left(\prod_{v \in \mathcal{V}} W_v \right) \left(\prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \, W_{uv} \right) \tag{2.1}$$

Maintenant, on connait $A(\beta)$ d'après le Théorème 2.17, et en sommant sur toutes les structures $\bar{\mathcal{E}}$ pour la prior d'hyperparamètres $\beta_{uv} W_{uv}$, on peut en déduire K !

$$1 = \sum_{\bar{\mathcal{E}} \in \overline{\mathcal{F}_n}} = \frac{1}{K} \left(\prod_{v \in \mathcal{V}} W_v \right) \frac{\det(I_n + \mathcal{Q}(\beta W))}{\det(I_n + \mathcal{Q}(\beta))}$$

En remplaçant K dans (2.1) on obtient la **posterior conjuguée** sur $\bar{\mathcal{E}}$

$$\mathbb{P}(\bar{\mathcal{E}} \mid \mathcal{D}) = \frac{1}{\det(I_n + \mathcal{Q}(\beta W))} \prod_{uv \in \bar{\mathcal{E}}} \beta_{uv} \, W_{uv}$$

Pour finir, en reprenant la forme de $\mathbb{P}(F \mid \mathcal{D})$ avant la sommation sur tous les θ, on en déduit la posterior sur $\Theta_{\bar{\mathcal{E}}}$

$$\mathbb{P}(\Theta_{\bar{\mathcal{E}}} \mid \mathcal{D}) = \prod_{v \in \mathcal{V}} \prod_{i=0,1} D(\theta_{v|pa(v)}(. \mid i); N''_{vu}(.i))$$

□

Pendant la démonstration, on comprend mieux comment les données $N_{uv}(ij)$ viennent s'intégrer et modifier la posterior sous la forme de *statistiques suffisantes* $N''_{uv}(ij)$. Elles modifient les lois de Dirichlet et donc la pondérations des probabilités du réseau.

Remarque 2.29. Les $N''_{uv}(ij)$ vérifient encore les relations

- $\forall u, v \in \mathcal{V}, \forall i, j \in \{0,1\},\ N''_{uv}(ij) = N''_{vu}(ji) > 0$
- $\forall u, v \in \mathcal{V}, \forall j \in \{0,1\},\ N''_{uv}(0j) + N''_{uv}(1j) = N''_{v}(j)$
- $\forall u \in \mathcal{V},\ N''_{u}(0) + N''_{u}(1) =: N''$

Remarque 2.30. [Meilă and Jaakkola, 2006] évoque ce théorème seulement pour des arbres. La différence pour les forêts provient du $\frac{1}{W}$ qui s'ajoute à W_{uv} et W_v.

2.3.4 Conclusion sur l'inférence

Faire de l'inférence bayésienne de graphe est peu commun à cause des nombreux paramètres cachés à prendre en compte. C'est pourquoi ce domaine fait partie du domaine de la recherche active. Cependant ici, sous certaines hypothèses, on parvient à une *forme explicite* pour la posterior sur F ce qui est plutôt rare ! On parle d'*inférence bayésienne exacte de graphes*.

Les posteriors s'obtiennent par un calcul simple en modifiant seulement les hyperparamètres β_{uv} et $N'_{uv}(ij)$ (Théorème 2.28). Cela va s'avérer très intéressant pour l'implémentation algorithmique de l'inférence.

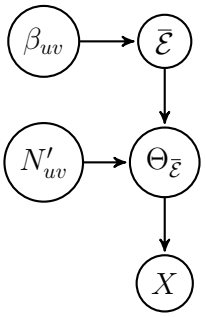


FIGURE 2.5 – *Résumé du tirage d'une forêt*

2.4 Implémentation

2.4.1 Un package Python

Il est possible de créer des *packages* Python facilement, et c'est ce que j'ai fait pour implémenter l'inférence. dans le dossier *« Bayestrees »* (fig. 2.6).

Le dossier `bayestress` comporte les modèles d'inférence et toutes sortes de fonctions utiles. Le dossier `data` contient les données sous la forme du tableau 1.1 où on note `network?-X.txt` pour dire qu'il s'agit d'un jeu de X cellules par unité de temps (par défaut le nombre de temps T vaut **11**). Enfin le dossier `tests` permet d'observer les graphiques et les graphes désirés pour tous les jeux de données.

2.4.2 Principe du modèle d'inférence

Le modèle d'inférence dans le fichier `model.py` est représenté sous la forme d'un *objet Python* (Python est un bon language orienté objet très pratique ici). Il possède différents attributs comme `n` le nombre de gène, `weight` un dictionnaire représentant les poids β_{uv}, `counts` un dictionnaire représentant le comptage des statistiques suffisantes N_{uv} qui seront très utiles pour l'inférence...

En effet pour mettre à jour le modèle, il suffit, **à chaque temps**, de mettre à jour (fig. 2.7) les hyperparamètres β_{uv} et N'_{uv} **d'un modèle vide** grâce aux données sous la forme N_{uv}, et tout cela grâce au Théorème 2.28. Il reste cependant un choix à faire... celui des hyperparamètres de la prior.

Quels hyperparamètres β_{uv} et N'_{uv} pour la prior ?

Ne connaissant aucune information particulière sur le réseau de gènes à étudier, on souhaite que les hyperparamètres soient uniformément pondérés. Ainsi après plusieurs essais, j'ai choisi $\boldsymbol{\beta_{uv} = 1}$ pour tout $u, v \in \mathcal{V}$ et $\boldsymbol{N'_{uv}(ij) = 0.5}$ pour tout $u, v \in \mathcal{V}$ et $i, j \in \{0, 1\}$ (prior de [Schwaller et al., 2019] pour N'_{uv}). Il faut que l'influence de la prior soit assez grande pour ne pas se laisser dominer par les données, mais pas trop forte pour éviter que les données soient négligeables.

Par exemple, pour le jeu de données à 5 gènes de 500 cellules par unité de temps, on sait que le comptage des données va rajouter 500 unités à chaque tableau N_{uv} (rappel : $N''_{uv}(ij) = N'_{uv}(ij) + N_{uv}(ij)$, Théorème 2.28). Comme ils sont initialisés à $0.5 * 4 = 2$, on

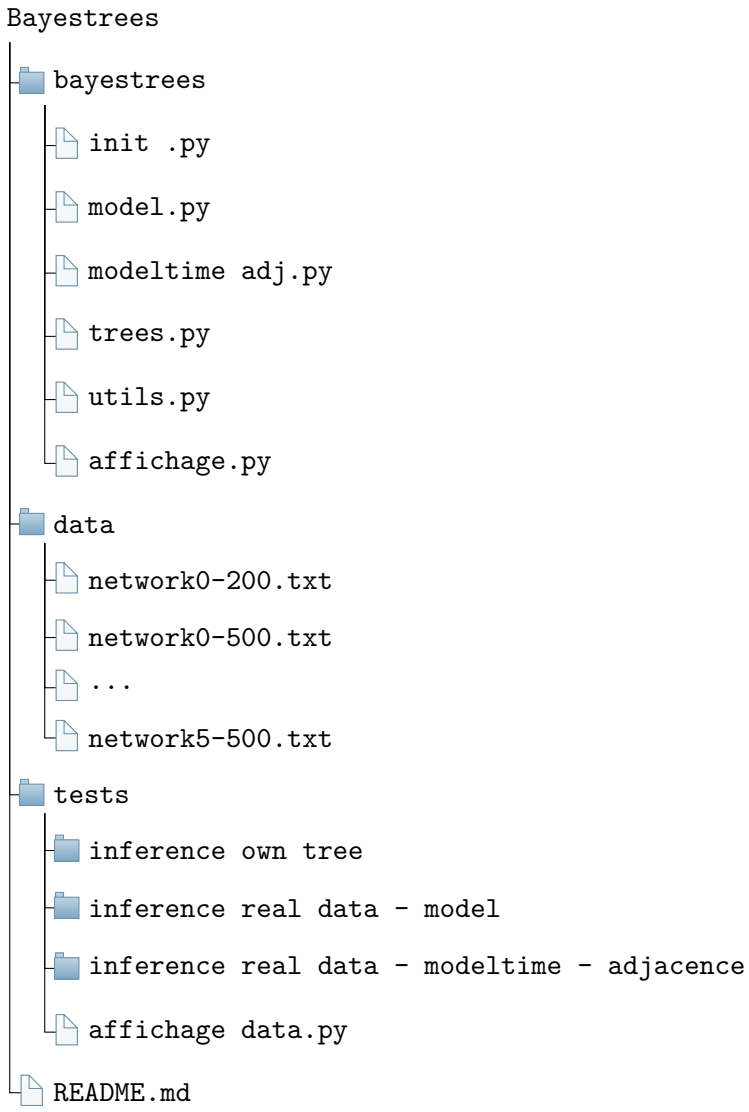


FIGURE 2.6 – Arborescence des dossiers du package *Bayestrees*

```
for t in time:
    models[t] = Model(n)
    models[t].update(x[t])
```

FIGURE 2.7 – *Partie du code où, pour chaque temps, on initialise un model vide* `Model(n)` *à n nœuds et on réalise l'inférence grâce à la fonction* `update` *qui met à jour les hyperparamètres grâce aux données* `x[t]`.

laisse les données dominer la posterior N'_{uv}. Cependant cela n'implique pas forcément de grands hyperparamètres β_{uv} relativement aux valeurs de la prior (fig. 2.8).

2.4.3 Représentation des résultats

• **Les $\theta_u(1)$ en moyenne**

Une première source d'informations sont les $\theta_u(1)$, qui représentent **la probabilité du gène u à être activé au cours du temps**. Après inférence, on peut les calculer en simulant une loi de Dirichlet à deux hyperparamètres (aussi appelée loi Bêta) qui sont les N'_u mis à jour. Mais il est plus judicieux de les tracer en moyenne pour s'affranchir de l'aléatoire de la simulation. Ainsi avec $(\theta_u(0), \theta_u(1))$ suivant une loi de Dirichlet,

$$D(\theta_u(0), \theta_u(1); N''_u(0), N''_u(1)) = \theta_u(0)^{N''_u(0)-1} * \theta_u(1)^{N''_u(1)-1}$$

on sait que,

$$\mathbb{E}(\theta_u(1)) = \frac{N''_u(0)}{N''_u(0) + N''_u(1)}$$

Voilà, on peut tracer les courbes $\mathbb{E}(\theta_u(1))$ pour tout $u \in \mathcal{V}$.

• **Les ρ_{uv} en moyenne**

Une seconde source d'informations sont les ρ_{uv}, qui sont les coefficients de corrélation entre les nœuds du réseau. Avec $X_u \sim \mathcal{B}(\theta_u(1))$ on a,

$$\rho_{uv} = \frac{Cov(X_u, X_v)}{\sigma_{X_u}\sigma_{X_v}}$$

et avec $Cov(X_u, X_v) = \theta_{uv}(1,1) - \theta_u(1)\theta_v(1)$ et $\sigma_{X_u} = \sqrt{\theta_u(0)\theta_u(1)}$ on obtient

$$\rho_{uv} = \frac{\theta_{uv}(1,1) - \theta_u(1)\theta_v(1)}{\sqrt{\theta_u(0)\theta_u(1)\theta_v(0)\theta_v(1)}}$$

On peut aproximer cela par les moyennes des θ_u et θ_{uv},

$$\mathbb{E}(\theta_{uv}(1,1)) = \frac{N''_{uv}(11)}{N''_{uv}(00) + N''_{uv}(01) + N''_{uv}(10) + N''_{uv}(11)}$$

Voilà, on peut tracer les courbes ρ_{uv} pour tout $u, v \in \mathcal{V}$.

```
models[0].weight
Out[1]:
{(1, 2): 0.3002481435371691,
 (1, 3): 0.1944005318938638,
 (1, 4): 0.19443160530158965,
 (1, 5): 0.1633717038528807,
 (2, 3): 0.3390147365240113,
 (2, 4): 0.14423655074754568,
 (2, 5): 0.16320088635611069,
 (3, 4): 0.2124377460727293,
 (3, 5): 1.5721163486086736,
 (4, 5): 0.6754772009928851}

models[8].weight
Out[2]:
{(1, 2): 1.1268152094370536,
 (1, 3): 1.3830868597558466,
 (1, 4): 0.2853169724666559,
 (1, 5): 0.4671744501842231,
 (2, 3): 39.9609752968436,
 (2, 4): 2027877780.0767949,
 (2, 5): 13.58378770226504,
 (3, 4): 0.6867090770864063,
 (3, 5): 32.75056435834777,
 (4, 5): 1.1275543555762573}
```

FIGURE 2.8 – *Résultat de l'inférence aux temps 0 et 8 pour* `network1-500.txt`. *On voit que les poids sont plus petits que 1 pour* $t = 0$ *alors que certains explosent pour* $t = 8$.

• **Les graphes dynamiques**

On dispose de nouveaux poids β_{uv} après inférence pour tout t, et on aimerait en obtenir une suite de graphe cohérente. Pour cela, **on peut calculer quelle est la probabilité d'une arrête** :

Théorème 2.31. *Soit $\bar{\mathcal{E}} \in \overline{\mathcal{F}_n}$ une structure de forêt et $Z = (Z_{uv})_{i,j\in\{1,\dots,n\}}$ sa matrice d'adjacence (pas forcément symétrique). En notant $\alpha_{uv} := \log(\beta_{uv})$*

$$\forall u, v \in \mathcal{V},\ \mathbb{P}(uv \in \bar{\mathcal{E}}) = \mathrm{Tr}((I_n + \mathcal{Q}(\alpha))^{-1}\partial_{\alpha_{uv}}\mathcal{Q}(\alpha))$$

Démonstration. On commence par exprimer la prior sur $\bar{\mathcal{E}}$ par sa forme exponentielle. En notant $\alpha_{uv} := \log(\beta_{uv})$, et $Z = (Z_{uv})_{u,v\in\{1,\dots,n\}}$ la matrice d'adjacence d'une forêt orientée, on a (en notant abusivement $A(\beta) = A(\alpha)$)

$$\mathbb{P}_0(\bar{\mathcal{E}}) = \frac{1}{A(\beta)}\prod_{uv\in\bar{\mathcal{E}}}\beta_{uv} = \frac{1}{A(\alpha)}\exp\left(\sum_{1\leqslant u,v\leqslant n}\alpha_{uv}Z_{uv}\right) =: \mathbb{P}_0(Z)$$

On a alors,

$$\begin{aligned}\mathbb{P}(uv \in \bar{\mathcal{E}}) = \mathbb{E}(\mathbb{1}_{uv\in\bar{\mathcal{E}}}) = \mathbb{E}(Z_{uv}) &= \sum_{Z\in\mathcal{F}_n} Z_{uv}\mathbb{P}_0(Z) \\ &= \frac{1}{A(\alpha)}\sum_{Z\in\mathcal{F}_n} Z_{uv}\exp\Big(\sum_{\substack{u\in\{1,\dots,n-1\}\\ v\in\{u+1,\dots,n\}}}\alpha_{uv}Z_{uv}\Big)\end{aligned}$$

On remarque que,

$$\mathbb{P}(uv \in \bar{\mathcal{E}}) = \partial_{\alpha_{uv}}\log(A(\alpha)) = \frac{1}{A(\alpha)}\partial_{\alpha_{uv}}A(\alpha)$$

or d'après le Théorème 2.17 on a $A(\alpha) = \det(I_n + \mathcal{Q}(\alpha))$ et d'après la formule de dérivation du déterminant, on obtient

$$\mathbb{P}(uv \in \bar{\mathcal{E}}) = \mathrm{Tr}((I_n + \mathcal{Q}(\alpha))^{-1}\partial_{\alpha_{uv}}\mathcal{Q}(\alpha))$$

□

Grâce à ce théorème, on peut calculer les probabilités de chaque arrêtes $\mathbb{P}(uv \in \mathcal{E}) = \mathbb{P}(uv \in \bar{\mathcal{E}}) + \mathbb{P}(vu \in \bar{\mathcal{E}})$ et les représenter sur un graphe. **Plus la probabilité est grande, plus l'arrête sera foncée et épaisse**. Passons maintenant à la mise en pratique avec des jeux de données.

2.5 Résultats

Pour présenter les résultats de l'inférence, j'ai choisi deux jeux de données en particulier : `network1-500.txt` qui est un arbre à 5 nœuds, et `network5-500.txt` qui est une forêt à 8 nœuds formée de `network1-500.txt` et de 3 autres sommets à part formant un arbre (donc une forêt composée de deux arbres).

2.5.1 Un arbre : network1-500.txt

Celui-ci possède $t = 11$ temps et 500 cellules par unité de temps. Il représente le réseau de gène (fig 2.9).

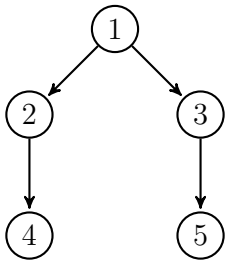


FIGURE 2.9 – *Réseau de gènes qui a été simulé par un modèle biologique et qu'il faut retrouver par inférence statistique grâce à ses données produites.*

On obtient, après inférence, l'évolution des $\theta_u(1)$ et des ρ_{uv} en moyenne pour tout $u \in \mathcal{V}$, ainsi que la suite des graphes (fig 2.11). On voit bien dans l'évolution des $\theta_u(1)$ que le gène 1 est activé en premier, puis le 2 et le 3, pour finir par le 4 et le 5. On est donc presque sûr que 1 active 2 et 3, mais on ne peut pas savoir qui active 4 et 5. Les corrélations ρ_{uv} mettent en valeur une corrélation positive 2-4 (en vert) et 3-5 (en marron). De plus les graphes montrent bien que cela ne peut être que 2-4 et 3-5 (pas d'arrêtes 1-4 ni 1-5 et pas de croisement flagrant 2-5 3-4). **Ainsi on a fortement à penser que le bon réseau d'interaction est celui (fig. 2.9) !**

2.5.2 Une forêt : network5-500.txt

Ce jeu de données possède aussi $t = 11$ temps et 500 cellules par unité de temps. Il a été généré par le même modèle biologique que ci-dessus, et représente les interactions génétiques de la figure (2.10) qui est une forêt à deux arbres.

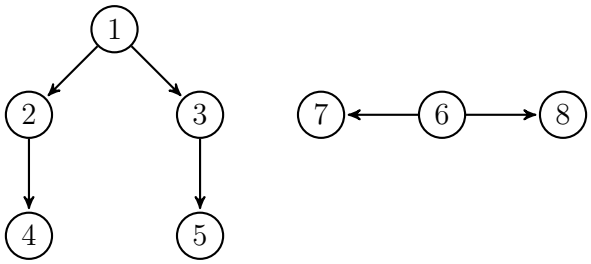


FIGURE 2.10 – *Réseau de gènes simulé par le même modèle biologique que la figure 2.9.*

On obtient, après inférence, l'évolution des $\theta_u(1)$ et des ρ_{uv} en moyenne pour tout $u \in \mathcal{V}$, ainsi que la suite des graphes (fig 2.12). Ici on voit bien dans l'évolution des $\theta_u(1)$ que le gène 1 et 6 sont activés en premier, le 4 et le 5 en dernier, et le reste au milieu. Les graphes permettent d'isoler 2 composantes connexes : 1-2-3-4-5 et 6-7-8. Puis en reprenant les analyses ci-dessus, on est assez certain que le bon schéma est celui (fig 2.10). On remarque cependant qu'avec les noeuds disposés de façon circulaire et non préférentielle, il est plus difficile de voir les interactions de l'arbre 1-2-3-4-5.

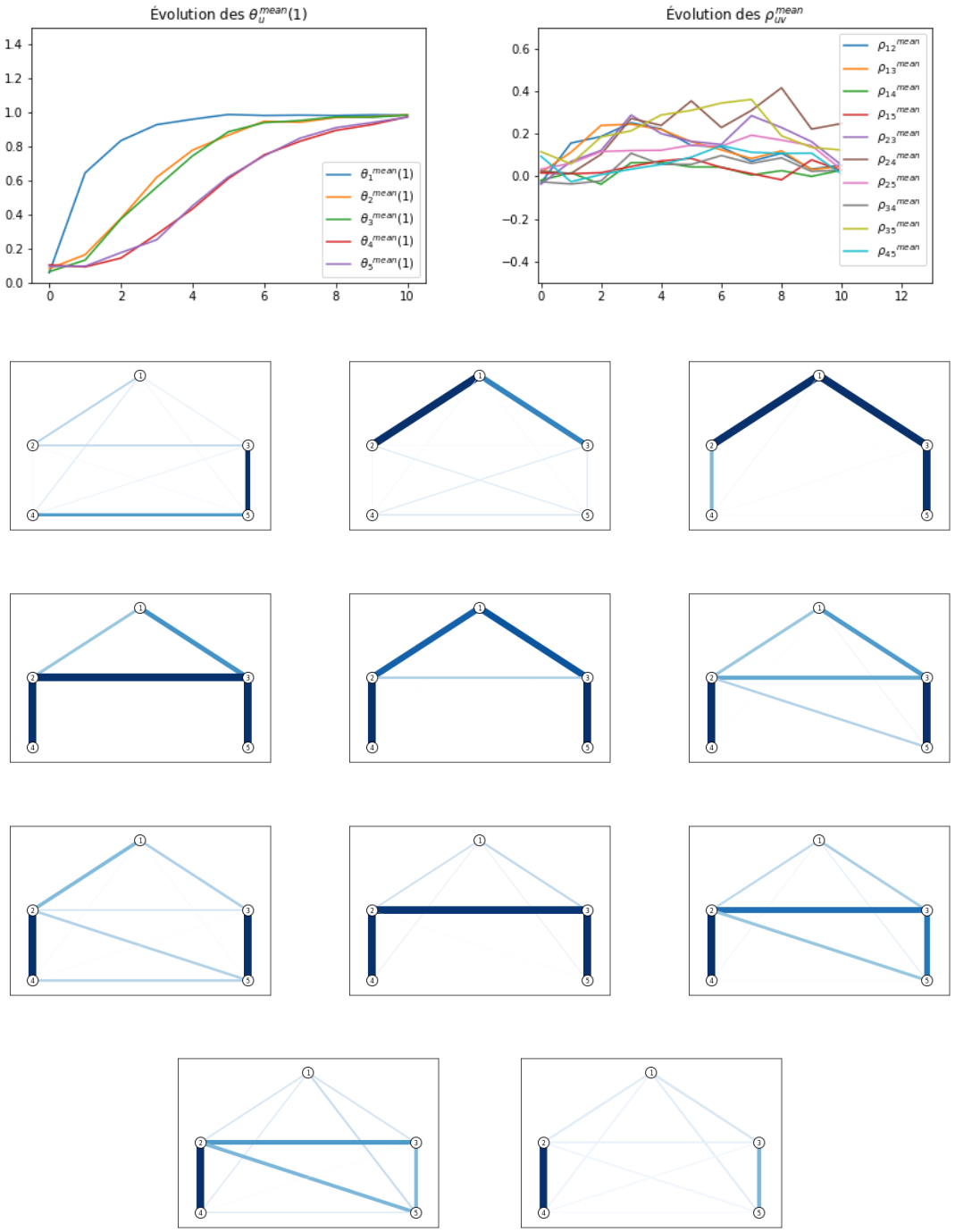


FIGURE 2.11 – *Résultat de l'inférence pour* `network1-500.txt` - *Dans un premier temps affichage des* $\theta_{uv}(1)$ *et les* ρ_{uv} *en moyenne. Puis évolution temporelle des graphes de gauche à droite et de haut en bas. Plus l'arrête est épaisse et sombre, plus la probabilité de présence d'une arrête est grande (la probabilité est calculée à l'aide du Théorème 2.31).*

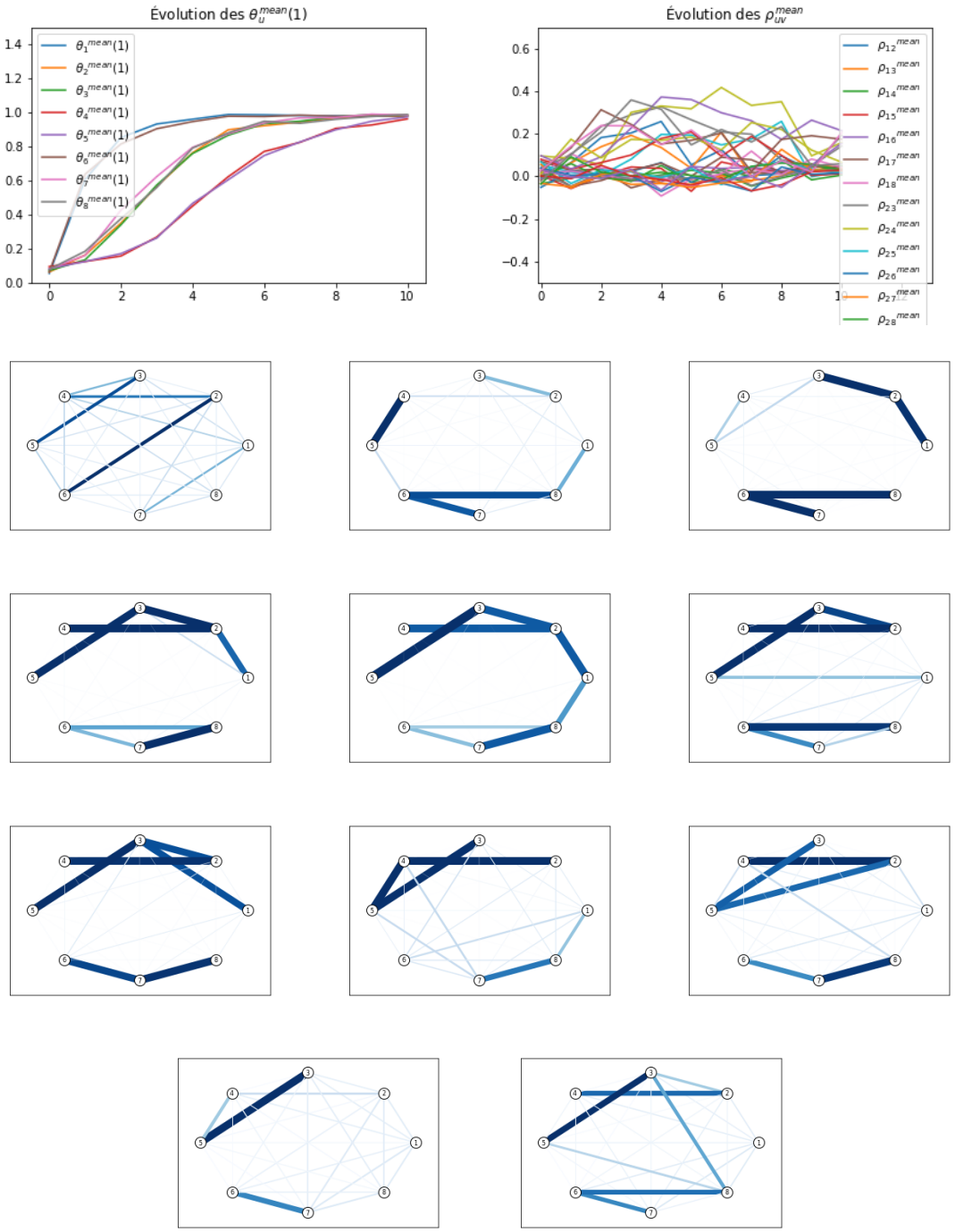


FIGURE 2.12 – *Résultat de l'inférence pour* `network5-500.txt` - *Dans un premier temps affichage des* $\theta_{uv}(1)$ *et les* ρ_{uv} *en moyenne. Puis évolution temporelle des graphes de gauche à droite et de haut en bas. Plus l'arrête est épaisse et sombre, plus la probabilité de présence d'une arrête est grande (la probabilité est calculée à l'aide du Théorème 2.31).*

Chapitre 3

Extensions du modèle

3.1 Introduction

Une difficulté majeure apparaît dans l'inférence des graphes. Il est difficile de voir directement le cheminement de l'information et de comprendre quels sont les liens entre les sommets à travers le temps. En effet beaucoup d'arrêtes qui n'existent pas dans la réalité ont parfois des probabiliés élevées et « remplacent » les vraies arrêtes. Pour régler ce problème, nous allons imposer une certaine « continuité » dans le film des graphes, et cela passe par une redéfinition de la prior.

3.2 Nouveau prior et MCMC par la méthode de Gibbs

3.2.1 Un nouveau prior plus « continu »

Nous allons forcer deux graphes temporellement adjacents à « se ressembler ». Il ne s'agit donc plus d'inférer les graphes un par un indépendamment, mais bien un **vecteur de graphe** directement sous la forme $(Z(0), Z(1), ..., Z(T))$ (où $Z(t)$ est la matrice d'adjacence de la forêt au temps t).

Pour le **premier modèle**, on avait indépendance de l'inférence entre les différents temps. La prior serait donc (et pour la posterior la même chose mais avec des hyperparamètres modifiés par l'information N_{uv}),

$$\mathbb{P}_0(Z(0), Z(1), ..., Z(T)) = \frac{1}{A_{\text{tot}}} \exp\left(\sum_{t=0}^{T} \sum_{1 \leqslant u,v \leqslant n} \alpha_{uv} Z_{uv}(t)\right)$$

Dans un **second modèle**, on favorise les suites de graphes qui partagent des arrêtes en commun d'un temps à l'autre.

Définition 3.1. On définit une nouvelle prior sur l'ensemble des vecteurs de structure des graphes

$$\mathbb{P}_0(Z(0), Z(1), ..., Z(T)) = \frac{1}{A_{\text{tot}}} \exp\left(\sum_{t=0}^{T} \sum_{1 \leqslant u,v \leqslant n} \alpha_{uv} Z_{uv}(t) + \boldsymbol{\lambda_1} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \sum_{\boldsymbol{1 \leqslant u,v \leqslant n}} \boldsymbol{Z_{uv}(t) Z_{uv}(t-1)}\right)$$

où $\boldsymbol{\lambda_1}$ est un coefficient réel positif à faire varier. Plus on le choisit grand avant l'inférence, plus on souhaite favoriser le tirage de graphes « proches », dans le sens qu'ils partagent beaucoup d'arrêtes en commun.

Pour simuler le vecteur $(Z(0), Z(1), ..., Z(T))$, nous allons nous tourner vers des méthodes de simulation numérique de **Monte-Carlo par chaînes de Markov (MCMC)**, notamment la **méthode de Gibbs**. Nous cherchons donc à simuler la loi a posteriori de $(Z(0), Z(1), ..., Z(T))$, ce vecteur étant vu comme l'état d'une chaîne de Markov à états finis sur laquelle on va se déplacer.

3.2.2 Méthode de Gibbs pour simulation d'une chaîne de Markov

Pour la simulation, nous allons partir d'un certain état $X^{(0)} = (Z^{(0)}(0), Z^{(0)}(1), ..., Z^{(0)}(T))$ de la chaîne et nous allons générer une suite de vecteurs censée tendre vers une réalisation de la loi a posteriori de $\boldsymbol{(Z(0), Z(1), ..., Z(T))}$. Pour y parvenir, nous allons mettre en place un algorithme de Gibbs dont voici le pseudocode :

Algorithm 1 Algorithme de Gibbs

Input : $X^{(0)} = (Z^{(0)}(0), Z^{(0)}(1), ..., Z^{(0)}(T))$ **and** k_{max}
Output : res**, a vector of length** k_{max} **of forests**
$X \leftarrow X^{(0)}$
$res \leftarrow [X^{(0)}]$
for $0 = k \leqslant k_{max}$ **do**
  $X^{(k+1)}$ new vector of length $T + 1$
  $\sigma \leftarrow$ random permutation of $[|1, T|]$
  for $t \in \{1, \dots, T\}$ **do**
    $X^{(k+1)}_{\sigma(t)} \leftarrow$ simulated by $L(X_{\sigma(t)} | X_{\neg\sigma(t)} = X^{(k)}_{\neg\sigma(t)})$ ▷ we note X by X^k
  end for
  $X \leftarrow X^{(k+1)}$ and $X^{(k+1)}$ is added to res
end for

On voit qu'il **suffit juste de connaître les lois conditionnelles a posteriori de chaque forêt en fonction de toutes les autres pour faire avancer la simulation**. Pour être plus clair sur le pseudocode, on a pour première simulation :

$$X^{(1)}_{\sigma(t)} \rightsquigarrow Loi(\boldsymbol{X_{\sigma(t)}} | \boldsymbol{X_{\neg\sigma(t)}} = X^{(0)}_{\neg\sigma(t)})$$

Avec un k_{max} grand (nombre d'itérations de l'algorithme de Gibbs) on a de grandes chances d'obtenir une simulation de la nouvelle loi recherchée $\mathbb{P}_{posterior}$. Il ne reste plus qu'à montrer qu'on connaît effectivement toutes ces lois conditionnelles.

3.2.3 Loi conditionnelle pour la méthode de Gibbs

Notons $\mathcal{W}(\beta)$ (ou $\mathcal{W}(\alpha)$ par abus de notation) la loi de la prior suivie par $\overline{\mathcal{E}}$ dans le premier modèle 2.14. Alors on peut montrer le résultat suivant,

Théorème 3.2. *Les lois conditionnelles sont*

$$\begin{aligned}
\mathbb{P}(Z(0) \mid Z(1), ..., Z(T), \mathcal{D}) &\sim \mathcal{W}(\beta_{uv} W_{uv} \exp(\lambda_1 Z(1)_{uv})) \\
\mathbb{P}(Z(1) \mid Z(0), ..., Z(T), \mathcal{D}) &\sim \mathcal{W}(\beta_{uv} W_{uv} \exp(\lambda_1 (Z(0)_{uv} + Z(2)_{uv})) \\
&\vdots \\
\mathbb{P}(Z(T) \mid Z(0), ..., Z(T-1), \mathcal{D}) &\sim \mathcal{W}(\beta_{uv} W_{uv} \exp(\lambda_1 (Z(T-1)_{uv}))
\end{aligned}$$

Remarque 3.3. En prenant $\lambda_1 = 0$ on retrouve la posterior du premier modèle

3.3 Implémentation

3.3.1 Une nouvelle classe Python

J'ai implémenté le nouveau modèle dans un autre script Python `modele adj.py` du dossier `bayestrees` (fig. 2.6). La structure globale est similaire au premier modèle mais la méthode de simulation change.

Mes seuls choix à faire sont $\boldsymbol{\lambda_1}$ et **le vecteur d'initialisation de l'algorithme de Gibbs** (ainsi que les poids β_{uv} et le comptage N_{uv} comme dans le premier modèle).

3.3.2 Représentation des résultats

Les arbres n'étant plus indépendants, on ne peut plus utiliser le Théorème (2.31) pour calculer les probabilités d'arrêtes. Cependant la **loi des grands nombres pour les chaînes de Markov** nous permet de tendre vers leur vraie valeur.

Pour s'assurer que l'algorithme de Gibbs est bien implémenté, on peut le tester pour $\lambda_1 = 0$, et nous devons retrouver des probabilités d'arrêtes convergents vers les valeurs calculées par le premier modèle grâce au Théorème (2.31).

3.4 Résultats

Avec $\lambda_1 = 0$ et en partant du vecteur de forêt nul, la simulation de l'algorithme de Gibbs pour `network0-200.txt` donne les résultats suivants (fig. 3.1). On voit que les probabilités des arrêtes convergent bien vers les probabilités théoriques calculées par le premier modèle.

Un grand défaut survient cependant pour cet algorithme de Gibbs. Nous avons testé l'algorithme pour des $\lambda_1 > 0$, mais la simulation devient interminable dès que le réseau s'aggrandit un peu. Cela est dû à des parcours de graphe trop long à cause de certains poids β_{uv} qui prédominent sur les autres... Les résultats de cette approche sont encourageants mais pas encore assez aboutis.

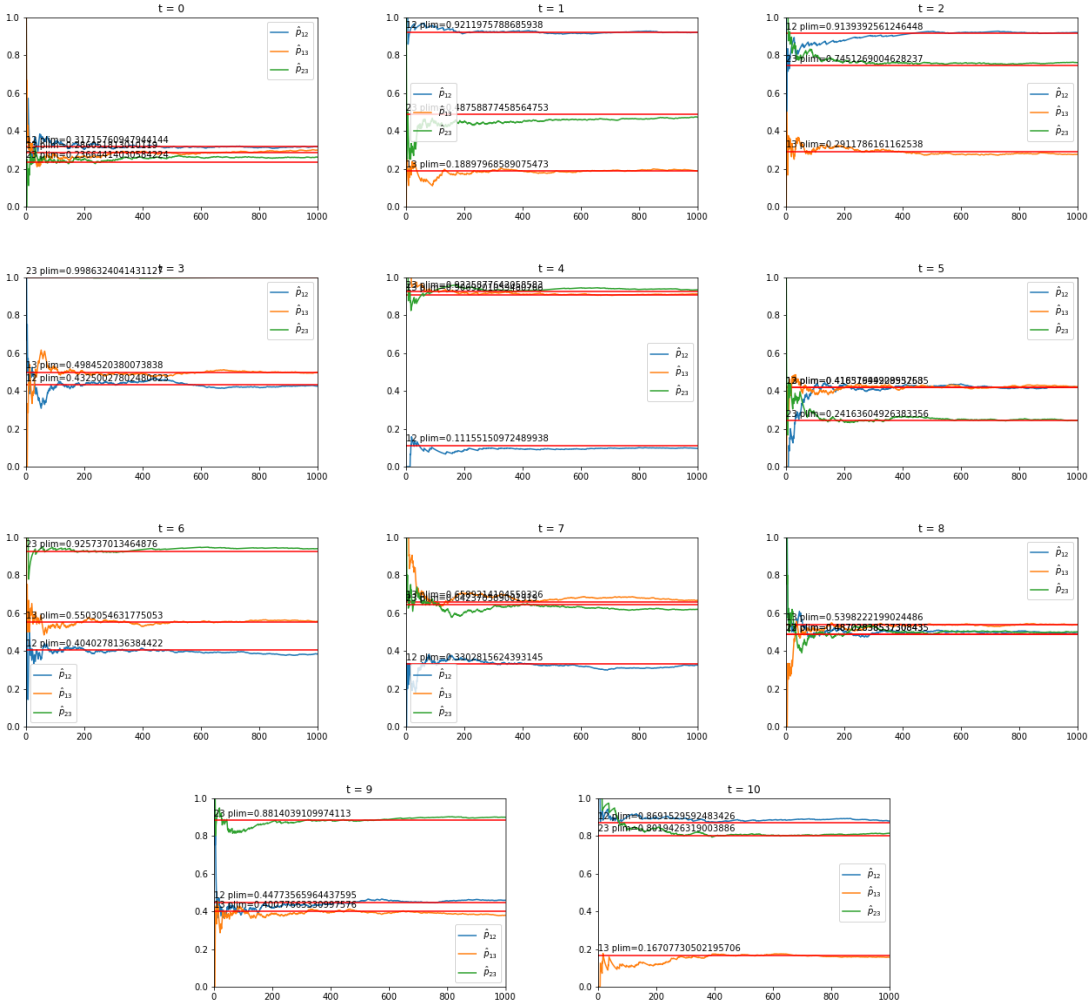


FIGURE 3.1 – *Simulation de Gibbs pour* $\lambda_1 = 0$ *et le jeu de données* `network0-200.txt`. *Les asymptotes horizontales en rouge sont les probabilités théoriques recherchées. On voit que la simulation de Gibbs converge bien vers les valeurs attendues.*

Bibliographie

[Chebotarev and Agaev, 2002] Chebotarev, P. and Agaev, R. (2002). Forest matrices around the Laplacian matrix. *Linear Algebra and its Applications*, 356(1-3) :253–274.

[Imane El Meouche, 2016] Imane El Meouche, Yik Siu, M. J. D. (2016). Stochastic expression of a multiple antibiotic resistance activator confers transient resistance in single cells. *Scientific reports*, page 2.

[Meilă and Jaakkola, 2006] Meilă, M. and Jaakkola, T. (2006). Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 16(1) :77–92.

[Schwaller et al., 2019] Schwaller, L., Robin, S., and Stumpf, M. (2019). Closed-form bayesian inference of graphical model structures by averaging over trees. *Journal de la société française de statistique*, 160(2) :1–23.