Inférence de la structure d'interactions de données bruitées

Mémoire

Simon Lizotte

Sous la direction de:

Antoine Allard, directeur de recherche
Jean-Gabriel Young, codirecteur de recherche

Résumé

La science des réseaux est notamment à la recherche de modèles mathématiques capables de reproduire le comportement de systèmes complexes empiriques. Cependant, la représentation usuelle, le graphe, est parfois inadéquate étant donné sa limitation à encoder uniquement les relations par paires. De nombreux travaux récents suggèrent que l'utilisation de l'hypergraphe, une généralisation décrivant les interactions d'ordre supérieur (plus de deux composantes), permet d'expliquer des phénomènes auparavant incompris avec le graphe. Or, la structure de ces réseaux complexes est rarement ou di□cilement observée directement. De fait, on mesure plutôt une quantité intermédiaire, comme la fréquence de chaque interaction, pour ensuite *reconstruire* la structure originale. Bien que de nombreuses méthodes de reconstruction de graphes aient été développées, peu d'approches permettent de retrouver les interactions d'ordre supérieur d'un système complexe.

Dans ce mémoire, on développe une nouvelle approche de reconstruction pouvant déceler les interactions connectant trois noeuds parmi des observations dyadiques bruitées. Basée sur l'inférence bayésienne, cette méthode génère la distribution des hypergraphes les plus plausibles pour un jeu de données grâce à un algorithme de type *Metropolis-Hastings-within-Gibbs*, une méthode de Monte-Carlo par chaînes de Markov. En vue d'évaluer la pertinence d'un modèle d'interactions d'ordre supérieur pour des observations dyadiques, le modèle d'hypergraphe développé est comparé à un second modèle bayésien supposant que la structure sous-jacente est un graphe admettant deux types d'interactions par paires. Les résultats obtenus pour des hypergraphes synthétiques et empiriques indiquent que la corrélation intrinsèque à la projection d'interactions d'ordre supérieur améliore le processus de reconstruction lorsque les observations associées aux interactions dyadiques et triadiques sont semblables.

Abstract

Network science is looking for mathematical models capable of reproducing the behavior of empirical complex systems. However, the usual representation, the graph, is sometimes inadequate given its limitation to encode only pairwise relationships. Many recent works suggest that the use of the hypergraph, a generalization describing higher-order interactions (more than two components), allows to explain phenomena previously not understood with graphs. However, the structure of these complex networks is seldom or hardly observed directly. Instead, we measure an intermediate quantity, such as the frequency of each interaction, and then *reconstruct* the original structure. Although many graph reconstruction methods have been developed, few approaches recover the higher-order interactions of a complex system.

In this thesis, we develop a new reconstruction approach which detects interactions connecting three vertices among noisy dyadic observations. Based on Bayesian inference, this method generates the distribution of the most plausible hypergraphs for a dataset using a *Metropolis-Hastings-within-Gibbs* algorithm, a Markov chain Monte Carlo method. In order to evaluate the relevance of a higher-order interaction model for dyadic observations, the developed hypergraph model is compared to a second Bayesian model assuming that the underlying structure is a graph admitting two types of pairwise interactions. Results for synthetic and empirical hypergraphs indicate that the intrinsic correlation to the projection of higher-order interactions improves the reconstruction process when observations associated with dyadic and triadic interactions are similar.

Table des matières

Liste des tableaux

Liste des figures

Remerciements

Ce mémoire est le produit fini d'un long parcours de recherche. J'ai été privilégié d'avoir un encadrement et un soutien aussi remarquables.

Je voudrais commencer par remercier mon superviseur, Antoine Allard, sans qui je n'aurais pas eu la chance de poursuivre cette maîtrise. N'étant pas un étudiant qui se démarque par l'excellence de ses notes, je n'étais pas en position d'obtenir une bourse et je ne croyais donc pas pouvoir joindre un groupe de physique théorique. Sans bien me connaître, Antoine a tout de même accepté de me prendre sous son aile pour un cours de projet 1, puis pour les études graduées. Au-delà de cette opportunité, je remercie Antoine pour son encadrement exemplaire. Sa bienveillance, sa pédagogie, sa disponibilité et ses grandes connaissances jointes à son esprit décontracté m'ont permis de me développer en tant que chercheur avec le sourire. Soucieux de ses étudiants, Antoine prend toujours la peine d'en faire plus pour maximiser nos apprentissages et pour former un groupe de recherche soudé.

Ma supervision ne serait toutefois pas aussi excellente sans mon codirecteur Jean-Gabriel Young. Je remercie Jean-Gabriel de m'avoir accompagné malgré la distance et de m'avoir amené à repousser mes limites. Son expertise, sa rigueur et son intuition m'ont permis d'explorer un projet intéressant et original qui sort du domaine d'expertise des étudiants et des professeurs du groupe de recherche. Je suis honoré d'avoir été son premier étudiant gradué. Malheureusement, compte tenu de la pandémie, je ne l'ai rencontré que brièvement en personne. Néanmoins, au cours de nos rencontres bimensuelles, une belle chimie s'est développée entre nous et je n'attends que l'occasion de pouvoir discuter autour d'une bière.

Je remercie Patrick et Khader qui ont accepté d'évaluer mon colloque et ce mémoire. Grâce à leur aide, j'ai pu approcher les di□érents problèmes mathématiques plus rigoureusement. Je remercie Patrick pour sa passion et sa pédagogie qui m'inspirent à me dépasser. Je n'ai pas eu beaucoup d'occasions de collaborer avec lui au cours de ma maîtrise, mais je souhaite que l'occasion se présente. Je remercie également Khader pour les multiples discussions sur l'inférence statistique et sur les méthodes MCMC. Son expertise m'a permis d'acquérir une compréhension plus profonde de ces sujets.

Ajouté à l'excellence de ces professeurs, je salue le groupe de recherche Dynamica. Ce groupe

positif, accueillant et ouvert d'esprit rend l'environnement agréable et propice au travail appliqué et rigoureux. Il est di□cile de remercier chaque membre vu la taille du groupe, mais je les remercie tous individuellement d'avoir, de près ou de loin, fait la di□érence. Plus particulièrement, je remercie François pour le soutien, pour la compagnie, pour les multiples moments passés au café le Fou Aéliers, pour les discussions plus ou moins sérieuses et pour m'avoir introduit à son groupe de badminton ; Béatrice pour son bonheur contagieux et sa présence rayonnante ; Charles et Vincent pour le *zoo* et pour avoir été des étudiants séniors et parrains hors pair.

Bien que le groupe de recherche et les professeurs soient directement liés à mes travaux, il m'aurait été di□cile d'accomplir ce travail sans mon entourage. Je remercie les *shitposteux* Tony, Cléroux (ou Alexandre pour les moins intimes), Zachary, François (encore), Axel, Olivier, Jean-David et Gabrielle de m'avoir fait découvrir les bières de microbrasserie et de m'avoir aidé à décrocher de ma recherche. Je remercie aussi mes autres bons amis du baccalauréat Nicolas, Justine et Colin pour leur soutien et les belles sorties à Montréal.

Malgré leur distance de ma vie universitaire, ma famille a joué un rôle important dans ma vie quotidienne. Je remercie mes parents Hélène et Michel, qui m'ont soutenu de toutes les manières imaginables et qui m'ont permis de m'épanouir depuis mon plus jeune âge. Je remercie ma grande soeur Maude, mes beaux-parents Isabelle et Denis, mon oncle Sylvain et ma tante Martine pour leur présence et pour être aussi attentionnés. Finalement, je remercie Daniela pour son soutien, son amour et sa présence inconditionnels qui m'apportent le sourire aux lèvres chaque matin.

Avant-propos

Le chapitre 2 de ce mémoire contient l'article *Hypergraph reconstruction from noisy pairwise observations*. L'auteur principal de cet article est également l'auteur de ce mémoire : il a développé les di□érents modèles, a écrit le code des simulations numériques, a produit les figures et a rédigé l'article. Les coauteurs Antoine Allard et Jean-Gabriel Young ont contribué à l'élaboration du projet de recherche, à la progression du projet et à la rédaction de l'article.

Cet article a été soumis le 13 octobre 2022 au journal en libre accès à comité de lecture *Communications Physics* publié par *Nature Portfolio*. Il est actuellement en processus de révision par les pairs. L'article est disponible dans l'archive *arXiv* : https://arxiv.org/abs/2208.06503.

Introduction

La science des réseaux occupe une place importante dans une grande variété de domaines. Ceci est dû à l'outil mathématique fondamental dans ce domaine, le graphe, qui permet de représenter abstraitement de quelconques relations (liens) entre des paires d'éléments (noeuds). Grâce à cette universalité, ce type d'analyse systémique permet de traiter avec une même méthodologie les interactions entre protéines [1], les réseaux trophiques [2], les réseaux sociaux [3], la propagation de maladies infectieuses [4] ou encore le cerveau [5] pour en énumérer que quelques exemples. En fait, cette interdisciplinarité de la science des réseaux est une de ses principales forces : elle permet de combiner les connaissances de nombreux domaines.

Bien que cette modélisation soit puissante et polyvalente, la structure en graphe d'un système ne peut parfois pas être observée directement. Par exemple, contrairement à un réseau de distribution électrique où les câbles qui relient les transformateurs sont visibles, la nourriture que consomme une espèce d'animal peut di□cilement être déterminée avec certitude [6]. Certaines observations pourraient être manquantes ou d'autres pourraient être spécifiques à la zone où elles ont été mesurées, ce qui rend di□cile la généralisation des résultats.

L'*inférence de la topologie des graphes*, une branche de la science des réseaux, s'intéresse au problème de déterminer la structure en graphe d'un système à partir de données incomplètes ou incertaines. Trois sujets principaux sont à l'étude dans cette branche [7] : la prédiction de liens, la tomographie de graphes et la prédiction de la présence ou de l'absence des liens à partir d'observations. La prédiction de liens s'intéresse à détecter les faux positifs et les faux négatifs parmi les interactions observées d'un graphe, soit une forme de validation des observations [8, 9]. La tomographie de graphes prédit quels sont les liens et noeuds à l'intérieur du « périmètre » observé. Par exemple, en supposant que le graphe soit un arbre duquel la racine et les feuilles ont été observées, il est possible d'estimer quels sont les noeuds et liens internes non observés à l'aide de di□érents modèles d'inférence. Cette approche permet entre autres de déduire les noeuds et liens intermédiaires du réseau de connexions Internet [10, 11]. Enfin, la prédiction de l'existence ou de l'absence des liens à partir des observations, qu'on nomme *la reconstruction de graphes*, permet de déterminer le graphe d'un système duquel on ignore les interactions. C'est dans cette sous-division de l'inférence de la topologie des graphes que s'inscrit ce mémoire.

Une grande variété de méthodes de reconstruction de graphes ont vu le jour au cours des dernières décennies [12]. Par exemple, ce problème a été exploré en biologie avec une forêt d'arbres décisionnels [13], avec le coe□cient de corrélation de Pearson appliqué sur des fenêtres de séquences temporelles [14], avec des équations di□érentielles ordinaires [15] et avec un modèle d'inférence bayésienne [16]. La recherche sur ce sujet n'est toutefois pas limitée à la biologie : des méthodes ont été développées dans divers autres domaines comme en sciences sociales [17] et en neurosciences [18]. De plus, des approches plus générales ont été proposées afin de s'appliquer à une plus grande variété de données [9, 17, 19–22]. Ce domaine est en constante évolution étant donné les di□cultés qui y sont rattachées telles que le bruit dans les données et la justesse de la modélisation des interactions [12].

Le graphe n'est cependant pas toujours un modèle adéquat. Fondamentalement, les liens permettent uniquement d'encoder des relations par paires tandis que les composantes de certains systèmes peuvent interagir via des groupes plus grands. Par exemple, dans un réseau de collaborations, un article scientifique n'est pas le fruit de collaborations dyadiques d'auteurs, il s'agit plutôt d'un groupe d'individus qui ont travaillé collectivement sur un même projet. Afin de modéliser ces interactions, il est nécessaire d'introduire les *interactions d'ordre supérieur*, une généralisation du lien qui connecte plus de deux entités. Au cours des dernières années, il est devenu clair que ces connexions en groupe jouent un rôle important en modélisation [23]. Elles permettent par exemple de mieux expliquer la dynamique corticale du cerveau [24], les changements de phases « explosifs » [25], l'équilibre de la biodiversité d'un écosystème [26] et la collaboration dans le jeu des biens publics [27].

Malgré l'importance de la reconstruction de graphes et l'importance des interactions d'ordre supérieur, il existe à ce jour un nombre limité d'approches pour reconstruire la structure d'ordre supérieur : par exemple, Roy-Pomerleau [28] reconstruit hiérarchiquement les interactions d'ordre supérieur significatives à partir d'une observation de graphe biparti, Young et al. [29] reconstruisent la structure à partir d'un graphe observé grâce à un principe de parcimonie (défavorise un grand nombre d'interactions d'ordre supérieur), Santoro et al. [30] reconstruisent la structure de séries temporelles en s'appuyant sur la cote Z et Musciotto et al. [31] filtrent les interactions d'ordre supérieur à partir d'un modèle nul. Or, ces méthodes supposent l'exactitude absolue des mesures expérimentales malgré le fait qu'une portion importante de ces données proviennent d'expériences bruitées ou ces méthodes ne s'appliquent pas à des observations dyadiques, un format de données commun.

L'inférence bayésienne occupe une place importante en science des réseaux. Il s'agit d'une technique d'inférence statistique puissante permettant de déduire les paramètres d'un modèle en s'informant des données. Elle se distingue des autres approches en incorporant des lois de probabilité *a priori* sur les paramètres, ce qui lui permet de mieux performer sur des petits jeux de données et d'exprimer l'incertitude sur les résultats à l'aide de probabilités. C'est en raison de cette approche orientée sur les données que l'inférence bayésienne occupe

un rôle important dans la science des réseaux : elle permet par exemple d'e□ectuer de la détection de communauté [32–40], di□érents types d'inférence de la topologie des graphes [9, 16, 20, 22, 29, 35, 41–44] et l'inférence de la position des noeuds pour des modèles aléatoires géométriques [45, 46].

Dans cette optique, le projet présenté dans ce mémoire propose une approche d'inférence bayésienne pour reconstruire des hypergraphes, une structure d'interactions d'ordre supérieur, à partir d'observations par paires bruitées. En reconstruction, l'inférence bayésienne permet de prendre en compte toute l'information contenue dans les données et d'o□rir un ensemble de graphes possibles plutôt qu'un seul graphe, ce qui procure un résultat robuste, nuancé et flexible par rapport aux hypothèses supposées.

Le premier chapitre de ce mémoire est consacré à l'introduction des concepts de base nécessaires à l'élaboration d'une approche bayésienne de reconstruction. On y introduit les di□érentes représentations d'un réseau complexe, la théorie des probabilités ainsi que l'inférence bayésienne et les notions rattachées. Le chapitre est clos par la présentation du modèle de reconstruction de graphes de Young et al. [22] suivi des méthodes numériques requises pour l'appliquer en pratique. Au deuxième chapitre, sous forme d'un article scientifique, un modèle bayésien original basé sur les notions du chapitre précédent est développé pour e□ectuer la reconstruction d'hypergraphes à partir de mesures bruitées de ses interactions dyadiques. Ce modèle est alors comparé à un second modèle, analysé dans la Réf. [22], qui considère plutôt di□érents types d'interactions par paires. Le mémoire se termine par une conclusion qui résume les avantages et les inconvénients du modèle développé, puis qui propose des avenues d'exploration possibles pour bonifier la méthode.

Chapitre 1

Notions préliminaires

Les outils utilisés dans le projet de recherche sont présentés dans ce chapitre. Dans un premier temps, la section 1.1 se penche sur différentes représentations mathématiques d'un réseau complexe. Dans un deuxième temps, la section 1.2 présente un aperçu de la théorie des probabilités et les lois de probabilités nécessaires dans ce travail. Dans un troisième temps, la section 1.3 étend les lois de probabilités aux différentes représentations d'un réseau complexe. Dans un quatrième temps, la section 1.4 introduit le concept d'inférence bayésienne et la section 1.5 introduit le modèle bayésien de reconstruction de graphe de Young et al. [22]. Dans un cinquième et dernier temps, les sections 1.6 et 1.7 présentent les méthodes de Monte-Carlo utilisées dans ce travail pour échantillonner suivant des lois de probabilités quelconques et en estimer des statistiques. Ces algorithmes permettront notamment d'appliquer les modèles de reconstruction de la section 1.5 et du chapitre 2.

1.1 Les systèmes complexes et leur structure

Depuis le 17^{e} siècle, le réductionnisme est l'approche privilégiée en sciences : en divisant un problème difficile en plusieurs petits problèmes simples, on espère pouvoir en comprendre le portait global. Or, l'étude des domaines tels que la théorie du chaos, la biologie, les communications, les neurosciences révèle que le réductionnisme n'est pas une approche appropriée pour certains systèmes ; ils doivent être analysés comme un tout.

En science des réseaux, ces systèmes sont qualifiés de *complexes*. Mitchell [47] définit un système complexe comme un « *système dans lequel de vastes réseaux de composantes sans contrôle central obéissant à des règles de fonctionnement simples font apparaître un comportement collectif complexe, un traitement d'information sophistiqué et de l'adaptation provenant d'un apprentissage ou de l'évolution* »[1]. Le système complexe est donc fondamentalement *plus que la somme de ses parties* ; des propriétés globales *émergent*, elles ne s'observent pas dans les

[1] Traduction libre.

composantes [48]. Cependant, la définition d'un système complexe n'est pas universelle et, par conséquent, il est parfois di□cile de déterminer lesquels sont complexes. Néanmoins, plusieurs systèmes tels que les interactions entre protéines, les réseaux sociaux et le cerveau sont généralement reconnus comme étant complexes.

Afin de poursuivre une analyse quantitative d'un système complexe, celui-ci doit être doté d'une représentation mathématique. Toutefois, cette représentation peut être définie de nombreuses manières. Cette section a comme objectif de définir les représentations utilisées au cours de ce travail et d'en présenter leurs principales caractéristiques.

1.1.1 Le graphe

Parmi les di□érentes représentations mathématiques possibles, la plus simple et la plus intuitive est sans doute le graphe. Dans un graphe, les composantes du système sont représentées par des *noeuds* qui interagissent par paires via des *liens*. Cette abstraction mathématique permet de modéliser la structure d'interactions d'une large gamme de systèmes tels que les réseaux trophiques, les réseaux de transports et les cerveaux.

Dans le jargon, les termes « réseau » et « réseau complexe » sont régulièrement utilisés de manière interchangeable avec « graphe ». Toutefois, comme proposé par Crane [49], l'utilisation de « réseau » ou « réseau complexe » dans cet ouvrage fait référence à la structure d'interactions d'un système réel et le terme « graphe » est réservé à sa modélisation mathématique.

Mathématiquement, un *graphe* est un doublet $G := (V, E)$, où V est l'ensemble de noeuds (ou sommets) et où E est l'ensemble de liens (ou arêtes), les paires de noeuds connectées. Une manière intuitive de schématiser un graphe est d'illustrer chaque noeud par un point et chaque lien par un trait qui relie deux points (voir figure1.1b). Lorsqu'un noeud u est connecté à v par un lien (u, v) E, il est dit que v est un *voisin* de u. L'ensemble des voisins d'un noeud est appelé son *voisinage*. Dans ce mémoire, on note $n := |V|$ le nombre de noeuds dans un graphe.

Bien que précise et formelle, cette définition en termes d'ensembles est peu pratique en modélisation. Pour alléger la notation, les noeuds v V sont identifiés par un nombre entier entre 1 et n. On note que ceci suppose un ordre arbitraire sur l'ensemble V : les noeuds de V ne sont pas ordonnés tandis que les indices entiers associés le sont. Ainsi, il existe $n!$ manières équivalentes de sélectionner les indices. Toutefois, cette particularité n'est pas problématique dans ce mémoire.

Pour simplifier davantage la notation, les graphes sont représentés par leur *matrice d'adjacence* A. Dans la matrice d'adjacence, les lignes et les colonnes représentent les noeuds du graphe, ce qui en fait une matrice carrée de taille $n \times n$. Ses éléments, a_{ij}, indiquent si le lien (i, j)

existe :

$$a_{ij} := \mathbb{1}_E(i,j) = \begin{cases} 0 & \text{si } (i,j) \text{ n'est pas un lien, } (i,j) \notin E \\ 1 & \text{si } (i,j) \text{ est un lien, } (i,j) \in E \end{cases} \tag{1.1}$$

où $\mathbb{1}_E$ dénote la fonction indicatrice de l'ensemble E. On note que la matrice d'adjacence est une fonction du graphe G, mais son symbole « A » et ses éléments « a_{ij} » ne dépendent pas explicitement de G. Malgré ce léger manque de rigueur, le graphe associé à une matrice d'adjacence sera clair avec le contexte, car un seul graphe sera traité à la fois.

Dans le cadre de ce travail, seuls les graphes *simples* sont à l'étude, c'est-à-dire les graphes non orientés (un lien (u,v) est équivalent au lien (v,u)), sans boucles (liens de la forme (u,u)) et sans multiliens (plusieurs liens identiques entre deux noeuds). Par conséquent, le terme « graphe » signifie « graphe simple » dans ce document. Par ailleurs, on remarque que l'absence de multiliens était déjà sous-entendue dans la définition présentée du graphe puisque E est un ensemble et non un multiensemble.

1.1.2 Le graphe multiplexe

Il est parfois possible d'identifier plusieurs types de connexions entre les éléments d'un système. Dans un système de transport par exemple, il est possible de se déplacer par train entre les gares et par vol d'avion commercial entre différents aéroports des mêmes villes. Bien qu'on puisse utiliser deux graphes pour modéliser ce système, il est pratique de les joindre dans un seul objet mathématique étant donné que ces derniers partageraient les mêmes noeuds. Pour ce faire, on définit le *graphe multiplexe* (ou graphe *multicouches*) comme un triplet $G := (V, E, C)$, où V est l'ensemble les noeuds, E est l'ensemble des liens (u, v, c) connectant les noeuds u et v de la couche c et où C est l'ensemble des couches. Pour l'exemple du système de transport, les villes peuvent être représentées par des noeuds, et les chemins de fer et les vols d'avions par des liens de deux couches différentes.

1.1.3 Les représentations d'ordre supérieur

La représentation en graphe est parfois insuffisante pour conserver toute l'information sur la structure d'un système ; les réseaux de collaborations en sont un excellent exemple. Dans un réseau de collaborations, les auteurs « interagissent » par les articles auxquels ils ont collaboré. Puisque les articles peuvent être écrits par plus de deux auteurs, les collaborations ne peuvent pas être représentées directement par des relations dyadiques.

Afin de remédier à ce problème, les auteurs peuvent être connectés aux articles qu'ils ont écrits plutôt qu'être connectés entre eux. Les articles deviennent ainsi un autre « type » de noeud, noeuds qui sont ajoutés à l'ensemble V. Ce type de graphe, où il est possible de partitionner les noeuds de façon à ce que tous les liens connectent une partition à une autre, est appelé *graphe biparti*. La figure 1.1b illustre le graphe biparti associé à trois auteurs collaborant dans

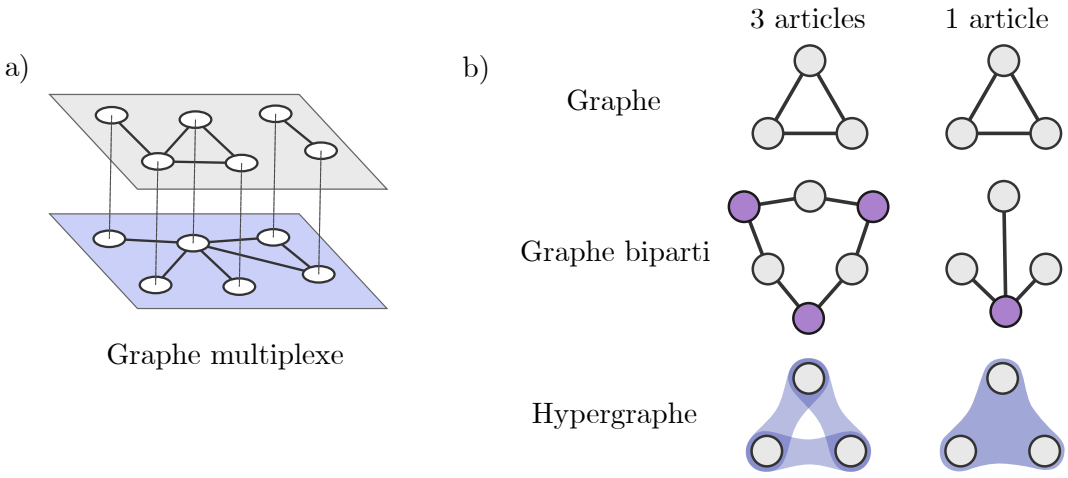


Figure 1.1 : (a) Graphe multiplexe. (b) Représentation en graphe, en graphe biparti et en hypergraphe d'un réseau de collaborations où trois auteurs ont collaboré au même article et d'un réseau où trois auteurs ont collaboré par paires à trois articles di□érents. Les graphes de la figure (b) sont le résultat d'une projection du graphe biparti sur les auteurs : deux auteurs sont connectés s'ils ont collaboré à au moins un article. Les noeuds en gris sont les auteurs et ceux en mauve illustrent les articles, des noeuds ajoutés dans le graphe biparti. Avec la représentation en graphe, un article écrit par trois auteurs et trois articles écrits par chaque paire d'auteurs mènent à la même projection. Il faut utiliser la représentation en graphe biparti ou en hypergraphe pour distinguer ces deux situations.

un même article et le graphe biparti représentant trois auteurs collaborant par paires dans trois articles di□érents.

Le désavantage de la représentation en graphe biparti dans ce contexte est que de nouveaux noeuds sont ajoutés au graphe. Puisque les noeuds n'ont plus la même signification, par exemple les articles et les auteurs, les outils d'analyse standards sur les graphes sont souvent inappropriés. C'est pourquoi il est fréquent d'e□ectuer une *projection* du graphe biparti sur un type de noeuds. Pour un réseau de collaborations par exemple, cette projection consiste à connecter les auteurs s'ils ont écrit au moins un article ensemble, de sorte que les noeuds du graphe résultant soient les auteurs. Toutefois, la figure 1.1b illustre qu'un article écrit par trois auteurs et que trois articles écrits par chaque paire d'auteurs mènent à la même projection. Ceci indique que cette approche entraîne une perte d'information sur la structure d'interactions du système.

Pour préserver l'information du système sans ajouter de noeuds, on introduit la notion d'hypergraphe. Un *hypergraphe* est défini comme un doublet $H = (V, E)$ où V est l'ensemble des noeuds et E est l'ensemble d'*hyperliens*, une généralisation du lien qui permet de regrouper un nombre quelconque de noeuds. Un hyperlien regroupant k noeuds est dénoté *k-lien*. À partir d'un hypergraphe, un article ayant k coauteurs se décrit par un k-lien tandis que trois articles écrits par des paires d'auteurs se modélisent par des 2-liens (voir figure1.1b). Par ailleurs, on remarque que la représentation à l'aide d'un graphe biparti est équivalente à celle utilisant

un hypergraphe : chaque noeud de type « article » correspond à un hyperlien connectant ses auteurs.

1.2 Notions préliminaires des probabilités

En tentant d'éviter un débat philosophique, on note que la notion d'« aléatoire » (au sens stochastique, sans causalité) est délicate en science. Elle vient à l'encontre du principe de causalité qui est au fondement de la grande majorité des modèles scientifiques. D'une certaine manière, la théorie de la dynamique non linéaire, du chaos et de la complexité a remis en question l'explication de phénomènes par l'aléatoire. Dans cette théorie, plusieurs systèmes obéissent à des règles déterministes simples, mais exhibent une sensibilité extrême aux conditions initiales. Il s'agit par ailleurs d'un ingrédient des systèmes complexes. Toutefois, dans ce travail, on ne distingue pas le « pseudo-aléatoire » (provenant du chaos) de l'aléatoire. On considère que l'aléatoire constitue un modèle qui décrit approximativement un système dont le comportement est difficile à prédire.

Dans cette section, les différents concepts des probabilités, la notation employée et différentes lois de probabilités communes sont présentés. Afin de cadrer rigoureusement les concepts en probabilités, la première sous-section les définissant se veut plus formelle en introduisant quelques notions de la théorie de la mesure. Or, cette théorie ajoute une couche d'abstraction qui complexifie l'interprétation du travail effectué et ne joue pas de rôle important à la formulation des modèles de reconstruction. C'est pourquoi les sections et chapitres subséquents font usage de quelques abus de notation pour alléger la lecture. On redirige le lecteur vers le manuel *Probability Theory* de Klenke [50] pour obtenir plus de détails sur la théorie de la mesure.

1.2.1 Définitions

En probabilité, on s'intéresse aux *expériences aléatoires* (non déterministes, stochastiques), phénomènes pour lesquels le résultat observé est impossible à prédire avec certitude. Le résultat d'une expérience aléatoire est appelé une *réalisation* (aussi nommé issue ou épreuve), un ensemble possible de réalisations est nommé un *événement*, et l'ensemble de toutes les réalisations est contenu dans l'ensemble Ω nommé *univers*. Un événement A est réalisé si la réalisation $\omega \in \Omega$ de l'expérience aléatoire est contenue dans celui-ci : $\omega \in A$. Par exemple, considérons l'expérience aléatoire de lancer un dé à six faces. L'univers de cette expérience est

$$\Omega = \{\text{« tombé sur 1 »}, \text{« tombé sur 2 »}, \text{« tombé sur 3 »},\\ \text{« tombé sur 4 »}, \text{« tombé sur 5 »}, \text{« tombé sur 6 »}\}.$$

Si la réalisation de l'expérience est ω = « tombé sur 2 », alors l'événement « tombé sur un nombre pair »

$$A = \{\text{« tombé sur 2 »}, \text{« tombé sur 4 »}, \text{« tombé sur 6 »}\} \tag{1.2}$$

est réalisé, car $\omega \quad A$.

En pratique, il est parfois impossible ou inutile d'obtenir la véritable issue de l'expérience aléatoire : on mesure plutôt une caractéristique observable de celle-ci par une *variable aléatoire*. La variable aléatoire est une fonction d'une réalisation qui retourne une quantité interprétable. Par exemple, dans une partie de fléchettes, la position exacte $\omega = (x, y)$ d'une fléchette tirée (dans $\Omega = \mathbb{R}^2$) est sans importance : seule la zone de la cible dans laquelle celle-ci se trouve détermine le pointage obtenu. De cette manière, on définit la variable aléatoire X comme la fonction retournant la zone à chaque position sur la cible. L'événement A associé à « fléchette dans la zone centrale » est alors l'ensemble des réalisations (x, y) de cette zone

$$\begin{aligned} A &= \{(x, y) \quad \Omega | X((x, y)) = \text{« fléchette dans la zone centrale »}\} \\ &= \{X = \text{« fléchette dans la zone centrale »}\} \\ &= X^{-1}(\text{« fléchette dans la zone centrale »}), \end{aligned}$$

où $\{X = B\}$ est une notation abrégée et où $X^{-1}(B)$ est appelé la *préimage* de B par X.

Afin de pouvoir modéliser et e□ectuer des prédictions du comportement de l'expérience aléatoire, on définit la *probabilité* de ses événements. E□ectivement, selon l'interprétation fréquentiste, la fréquence relative d'occurrence d'un événement tend vers sa probabilité dans la limite où l'expérience aléatoire est répétée une infinité de fois, et selon l'interprétation bayésienne, la probabilité quantifie la certitude de la réalisation d'un événement (voir section 1.4). La probabilité est obtenue à partir de la *mesure de probabilité*, une fonction $\mathbb{P}$ qui associe un scalaire dans l'intervalle $[0, 1]$ à chaque événement A. Par la définition d'une mesure de probabilité, la probabilité de l'univers est 1 et la probabilité de l'ensemble vide est nulle. La *distribution* ou *loi de probabilité* $\mathbb{P}_X$ d'une variable aléatoire X est la mesure de probabilité de la préimage de la quantité observée B (aussi appelée la mesure image de $\mathbb{P}$ par X) : $\mathbb{P}_X(B) := \mathbb{P}(X^{-1}(B)) = \mathbb{P}(\{X \quad B\})$. Le *support* d'une distribution, noté $\text{supp}(\mathbb{P}_X)$, est le plus petit ensemble de valeurs (de la variable aléatoire) en cardinalité de probabilité 1. Par abus de langage, ce travail évoque le « support » d'une variable aléatoire en faisant référence au support de sa distribution.

En probabilité, il existe deux grandes classes de variables aléatoires : les variables aléatoires discrètes et les variables aléatoires continues. Ce qui les distingue est la cardinalité de leur support. E□ectivement, les variables aléatoires discrètes ont un support au plus dénombrable tandis que les variables aléatoires continues ont un support infini indénombrable. Le résultat du tir d'une pièce de monnaie et la position dans l'espace d'une particule de gaz sont des exemples de variables aléatoires discrète et continue respectivement.

Afin de définir la loi de probabilité d'une variable aléatoire discrète X, il est suffisant de déterminer la probabilité de chaque valeur x prise par X, car pour un ensemble de valeurs B,

$$\mathbb{P}(\{X \in B\}) = \sum_{x \in B \cap S_X} \mathbb{P}(\{X = x\}) = \sum_{x \in B \cap S_X} \mathbb{P}_X(\{x\}) = \sum_{x \in B \cap S_X} f_X(x), \qquad (1.3)$$

où la propriété d'une mesure $\mathbb{P}(A_1 \cup A_2) = \mathbb{P}(A_1) + \mathbb{P}(A_2)$ avec $A_1 \cap A_2 = \varnothing$ a été utilisée, $S_X = \text{supp}(\mathbb{P}_X)$ et $f_X(x)$ est la *fonction de masse*, soit la probabilité que X vaille x. Étant donné que $\mathbb{P}(\Omega) = 1$ et que $\mathbb{P}(A) \in [0, 1]$, la fonction de masse doit respecter $f_X(x) \geq 0 \quad \forall x$ et $\sum_{x \in S_X} f_X(x) = 1$.

Pour une variable aléatoire continue X, il est impossible de définir sa loi de probabilité en fonction de la probabilité de chaque valeur individuelle puisqu'il existe un infini indénombrable de valeurs possibles pour X. Effectivement, la condition $\sum_{x \in S} f_X(x) = 1$ ne pourrait être respectée. La distribution est alors définie à partir de sa *densité de probabilité* $f_X(x)$, une fonction qui, intuitivement, attribue une probabilité à ce que X soit dans le voisinage de x. Mathématiquement, la mesure de probabilité associée à un ensemble B de valeurs est l'intégrale (théorème de Radon-Nikodym [50])

$$\mathbb{P}(\{X \in B\}) = \int_{B \cap S_X} f_X(x)\mathrm{d}x. \qquad (1.4)$$

L'intégrale de Riemann apparaît, car la distribution est supposée absolument continue par rapport à la mesure de Lebesgue (une mesure permettant d'évaluer la longueur d'intervalles réels). Afin de respecter les conditions $\mathbb{P}(\Omega) = 1$ et $\mathbb{P}(A) \in [0, 1]$, la fonction de densité doit respecter $f_X(x) \geq 0 \quad \forall x$ et $\int_{S_X} f_X(x)\mathrm{d}x = 1$.

Une manière alternative de caractériser la distribution d'une variable aléatoire dont les valeurs sont des scalaires réels est d'utiliser sa *fonction de répartition*. La fonction de répartition $\text{CDF}_X(x)$ (de l'anglais *cumulative distribution function*) correspond à la probabilité qu'une variable aléatoire X vaille au plus x. Ainsi, elle est définie par $\mathbb{P}(\{X \leq x\})$, soit

$$\text{CDF}_X(x) = \sum_{\substack{y \in \text{supp}(X) \\ y \leq x}} f_X(y), \quad \text{et} \qquad (1.5)$$

$$\text{CDF}_X(x) = \int_{\text{supp}(X)} f_X(y)\mathbb{1}_{\{y \leq x\}}(y)\mathrm{d}y = \int_{-\infty}^{x} f_X(y)\mathrm{d}y \qquad (1.6)$$

pour une variable aléatoire X discrète et continue respectivement.

La probabilité d'un événement A_1 conditionnel à un autre A_2 est définie comme

$$\mathbb{P}(A_1|A_2) := \frac{\mathbb{P}(A_1 \cap A_2)}{\mathbb{P}(A_2)} \qquad (1.7)$$

où $\mathbb{P}(A_2) > 0$ [51]. La probabilité conditionnelle est alors la probabilité renormalisée que l'événement A_1 soit réalisé pour un certain événement A_2 fixé. Elle s'interprète comme la

probabilité de mesurer A_1 *sachant* que l'événement A_2 a été réalisé. Il est donc raisonnable qu'elle ne soit pas définie si $\mathbb{P}(A_2) = 0$: A_2 ne peut pas être réalisé et conséquemment, aucune réalisation de A_1 n'est possible pour ce A_2.

Les événements A_1 et A_2 sont dits *indépendants* si leur probabilité conditionnelle est identique à la probabilité de l'événement $\mathbb{P}(A_1|A_2) = \mathbb{P}(A_1)$, c'est-à-dire $\mathbb{P}(A_1 \cap A_2) = \mathbb{P}(A_1)\mathbb{P}(A_2)$. Deux variables aléatoires X et Y sont *indépendantes* si les événements $\{X \leq x\}$ et $\{Y \leq y\}$ sont indépendants

$$\mathbb{P}(\{X \leq x\} \cap \{Y \leq y\}) = \mathbb{P}(\{X \leq x\})\mathbb{P}(\{Y \leq y\}), \quad \text{ou} \tag{1.8}$$

$$\mathrm{CDF}_{X,Y}(x, y) = \mathrm{CDF}_X(x)\,\mathrm{CDF}_Y(y), \tag{1.9}$$

où la probabilité $\mathbb{P}(\{X \leq x\} \cap \{Y \leq y\})$ est la *fonction de répartition conjointe* $\mathrm{CDF}_{X,Y}(x, y)$. Des variables aléatoires $(X_i)_{i=1}^N$ sont dites *indépendantes et identiquement distribuées* (*i.i.d.*) si elles admettent toutes la même loi de probabilité et sont deux à deux indépendantes.

La fonction de répartition conjointe de l'équation (1.9) est liée à la *loi conjointe* des variables aléatoires X et Y, loi définie par la fonction de masse $\mathbb{P}(\{X = x\} \cap \{Y = y\}) = f_{X,Y}(x, y)$ pour une variable aléatoire discrète et par la densité $f_{X,Y}(x, y)$ pour une variable aléatoire continue. À partir de la densité conjointe, on définit également la *densité conditionnelle* d'une variable aléatoire X à l'événement associé à la réalisation de l'autre $\{Y = y\}$

$$f_{X|Y=y}(x) = \frac{f_{X,Y}(x, y)}{f_Y(y)}. \tag{1.10}$$

À partir de la fonction de masse (densité) conjointe des variables aléatoires discrètes (continues) X et Y, il est possible de retrouver la fonction de masse (densité) *marginale* de la variable X en sommant (intégrant) sur toutes les valeurs possibles de Y

$$f_X(x) = \sum_{y \in \operatorname{supp} Y} f_{X,Y}(x, y) \tag{1.11}$$

$$f_X(x) = \int_{\operatorname{supp} Y} f_{X,Y}(x, y)\mathrm{d}y \ . \tag{1.12}$$

Cette opération est nommée marginalisation.

L'*espérance* d'une variable aléatoire discrète et celle d'une variable aléatoire continue X sont définies respectivement comme

$$\mathbb{E}[X] := \sum_{x \in \operatorname{supp}(X)} x f_X(x) \qquad \text{et} \qquad \mathbb{E}[X] := \int_{\operatorname{supp}(X)} x f_X(x)\mathrm{d}x \tag{1.13}$$

et la variance comme

$$\mathbb{V}[X] := \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \tag{1.14}$$

$$= \mathbb{E}\left[X^2\right] - 2\mathbb{E}[X\,\mathbb{E}[X]] + \mathbb{E}\left[(\mathbb{E}[X])^2\right]$$

$$= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2. \tag{1.15}$$

Grâce à la loi des grands nombres, l'espérance s'interprète comme la moyenne arithmétique attendue d'un grand nombre de réalisations de X. La variance évalue le niveau de dispersion des valeurs autour de l'espérance tel qu'écrit dans sa définition à l'équation (1.14).

L'espérance conditionnelle d'une variable aléatoire discrète et celle d'une variable aléatoire continue X par rapport à l'événement $\{Y = y\}$ se définissent respectivement comme

$$\mathbb{E}[X|Y = y] := \sum_{x \in \text{supp}(X|Y)} x f_{X|Y=y}(x) \quad \text{et} \tag{1.16}$$

$$\mathbb{E}[X|Y = y] := \int_{\text{supp}(X|Y)} x f_{X|Y=y}(x)\mathrm{d}x. \tag{1.17}$$

La variance conditionnelle est définie par

$$\mathbb{V}[X|Y = y] := \mathbb{E}\left[(X - \mathbb{E}[X|Y = y])^2 \middle| Y = y\right]. \tag{1.18}$$

Il est également possible de définir ces quantités en tant que variables aléatoires si la valeur de Y n'est pas fixée. Celles-ci, notées $\mathbb{E}[X|Y]$ et $\mathbb{V}[X|Y]$, sont effectivement des variables aléatoires puisqu'elles agissent comme des fonctions assignant un scalaire à une réalisation

$$\omega \to \mathbb{E}[X|Y = Y(\omega)] \quad \text{et} \quad \omega \to \mathbb{V}[X|Y = Y(\omega)]. \tag{1.19}$$

Une des propriétés de l'espérance conditionnelle est que

$$\mathbb{E}_X[X] = \mathbb{E}_Y[\mathbb{E}_X[X|Y]], \tag{1.20}$$

où les indices X et Y dénotent la variable aléatoire pour laquelle l'espérance est calculée. La preuve de cette propriété est détaillée à la Réf. [52].

Pour la suite de ce travail, le langage et la notation utilisés sont grandement assouplis et reflètent l'usage commun des probabilités en physique. La notation d'ensemble pour la mesure de probabilité sera parfois omise de sorte que $\mathbb{P}(\{X = x\})$ soit écrit $\mathbb{P}(X = x)$. Le symbole $\mathbb{P}$ exprimera autant la densité de probabilité que la fonction de masse plutôt que la mesure. De plus, la fonction de masse ou de densité ne sera pas accompagnée d'un indice précisant la variable aléatoire à laquelle elle est rattachée, car celle-ci sera claire avec l'argument donné à la fonction. Par exemple, si p est une valeur de paramètre et G est un graphe, $\mathbb{P}(p)$ est la probabilité que le paramètre vaille p et $\mathbb{P}(G)$ est la probabilité que le graphe soit G. Finalement, les termes « distribution » et « loi » feront référence à la densité de probabilité ou à la fonction de masse plutôt qu'à la mesure image de la variable aléatoire, et le terme « réalisation » désignera parfois la valeur associée à une variable aléatoire plutôt qu'à sa préimage.

1.2.2 Lois de probabilité univariées communes

Afin de modéliser une variété de variables aléatoires, différentes lois de probabilités sont nécessaires. Dans le cadre de ce travail cependant, seules quelques-unes sont utilisées. Le

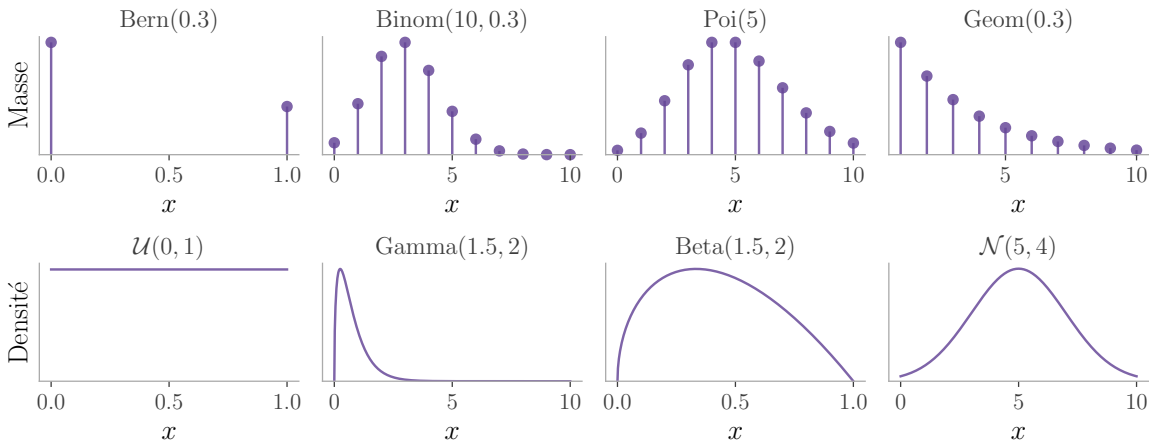


Figure 1.2 : Fonctions de masse et densités de probabilité des lois communes.

tableau 1.1 présente un résumé des caractéristiques de celles-ci et la figure 1.2 illustre l'allure de leur fonction de masse ou de leur fonction de densité. Ce document emploie également des lois tronquées et des lois de mélange, des distributions obtenues à partir d'autres lois. Bien que la définition de ces lois composites puisse s'appliquer aux lois continues, on présente ici uniquement leur version discrète qui est utilisée dans ce travail.

Comme son nom l'indique, la *loi tronquée* est le résultat d'une troncature du support d'une autre loi. La loi $f(x)$ tronquée sur l'intervalle $[a, b]$, notée $g(x)$, est donnée par

$$g(x) = \frac{f(x)}{\text{CDF}(b) - \text{CDF}(a)}, \qquad \text{supp}(g) = [a, b], \tag{1.21}$$

où CDF est la fonction de répartition associée à la variable aléatoire X.

Une *loi de mélange finie* (ou dénombrable) g est une combinaison convexe (coe□cients non négatifs qui somment à 1) de fonctions de masse f_i telles que

$$g = \sum_i w_i f_i, \qquad \text{supp}(g) = \bigcup_i \text{supp}(f_i) \tag{1.22}$$

avec $\sum_i w_i = 1$ et $w_i \geq 0 \quad i$. L'espérance d'une variable aléatoire X ayant comme distribution la loi de mélange g est

$$\begin{aligned} \mathbb{E}[X] &= \sum_{x \in \text{supp}(X)} x \sum_i w_i f_i(x) \\ &= \sum_i w_i \sum_{x \in \text{supp}(X)} x f_i(x) = \sum_i w_i \mathbb{E}[X_i], \end{aligned} \tag{1.23}$$

où X_i dénote une variable aléatoire avec la fonction de densité f_i. Un raisonnement similaire s'applique à une loi de mélange constituée de variables aléatoires continues et mène à des résultats analogues. La figure 1.3 illustre un exemple de loi de mélange constituée de deux lois de Poisson avec des paramètres λ di□érents.

Tableau 1.1 : Lois de probabilité communes utilisées dans ce document. Les densités et mesures de probabilités présentées ne sont définies que sur leur support ; il est donc sous-entendu que ces fonctions sont multipliées par la fonction indicatrice de leur support $\mathbb{1}_{\text{supp}(\mathbb{P})}$. On note que le support des lois continues peut être défini pour inclure ou exclure les bornes, car la mesure de probabilité d'un point est nulle. Pour la loi bêta par exemple, le support $[0, 1]$ peut être remplacé par l'intervalle $(0, 1)$.

Nom	Notation	Masse/densité	Paramètres	Support	Espérance	Variance
Bernoulli	$x \sim \text{Bern}(p)$	$p^x(1-p)^{1-x}$	$p \in [0,1]$	$\{0,1\}$	p	$p(1-p)$
Binomiale	$x \sim \text{Binom}(n,p)$	$\binom{n}{x} p^x(1-p)^{n-x}$	$p \in [0,1]$ $n \in \mathbb{Z}_+$	$\{0,\dots,n\}$	np	np
Poisson	$x \sim \text{Poi}(\lambda)$	$\frac{\lambda^x}{x!}e^{-\lambda}$	$\lambda \in \mathbb{R}_+$	$\mathbb{Z}_+$	λ	λ
Géométrique	$x \sim \text{Geom}(p)$	$(1-p)^{k-1}p$	$p \in [0,1]$	$\mathbb{Z}_+$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Uniforme (continue)	$x \sim U(a,b)$	$\frac{1}{b-a}$	$a,b \in \mathbb{R}$ $a < b$	$[a,b]$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Normale	$x \sim N(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\mu \in \mathbb{R}$ $\sigma^2 \in \mathbb{R}_+$	$\mathbb{R}$	μ	σ^2
Gamma	$x \sim \text{Gamma}(\alpha,\beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}$	$\alpha,\beta \in \mathbb{R}_+$	$\mathbb{R}_+$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$
Bêta	$x \sim \text{bêta}(\alpha,\beta)$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$	$\alpha,\beta \in \mathbb{R}_+$	$[0,1]$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

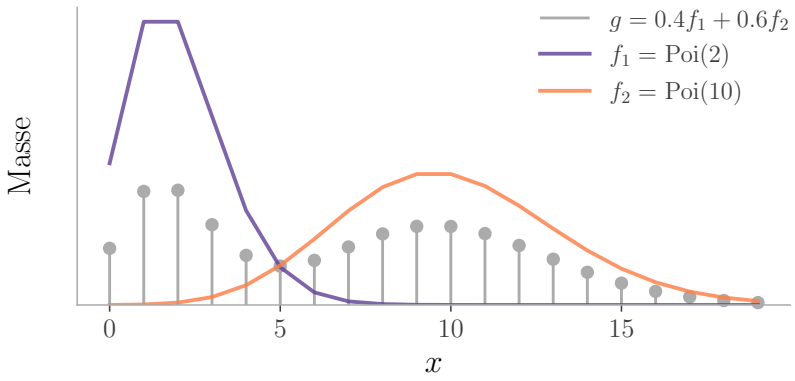


Figure 1.3 : Loi de mélange finie de deux lois de Poisson de poids $w_1 = 0.3$ et $w_2 = 0.7$ avec paramètres $\lambda_1 = 2$ et $\lambda_2 = 10$.

1.2.3 Théorème de la limite centrale

Un résultat important en théorie des probabilités est le *théorème de la limite centrale*. Ce théorème stipule que si les variables aléatoires i.i.d. $X_1, X_2, \ldots$ ont une espérance $\mu := \mathbb{E}[X_1]$ $\mathbb{R}$ et une variance finie $\sigma^2 := \mathbb{V}[X_1]$ $[0, \infty)$, alors dans la limite où $n \to \infty$, leur somme $\sum_{i=1}^{n} X_i$ suit une loi normale d'espérance $n\mu$ et de variance $n\sigma^2$ [50].

1.3 Graphes et hypergraphes aléatoires

Afin d'utiliser les graphes et les hypergraphes dans un contexte bayésien, le concept de variable aléatoire doit être défini sur ces objets. Cette tâche est relativement simple, car les graphes et hypergraphes aléatoires peuvent s'interpréter comme étant une loi conjointe sur des variables aléatoires binaires déterminant l'existence des liens et des hyperliens. Par exemple, la distribution d'un graphe aléatoire peut s'exprimer comme une loi conjointe sur les éléments a_{ij} du triangle inférieur de sa matrice d'adjacence.

Les graphes aléatoires constituent un outil important dans l'analyse de réseaux complexes. Ils permettent notamment d'étudier la propagation de maladies infectieuses par le retrait aléatoire de liens, appelé percolation [53], et de fournir des modèles nuls pour mettre en relief les particularités d'un graphe par rapport à l'ensemble aléatoire [54]. Comme les hypergraphes sont une généralisation des graphes, les hypergraphes aléatoires permettent de faire des analyses similaires dans le contexte d'interactions d'ordre supérieur. Dans le cadre de ce projet, les graphes et les hypergraphes aléatoires permettent de modéliser les interactions incertaines d'un système observé.

Bien qu'il existe de nombreuses manières d'assigner une probabilité à des graphes et à des hypergraphes, seules les quelques lois utilisées dans la suite de ce travail sont présentées. Le lecteur intéressé aux di□érentes lois de probabilité sur les graphes peut parcourir l'article

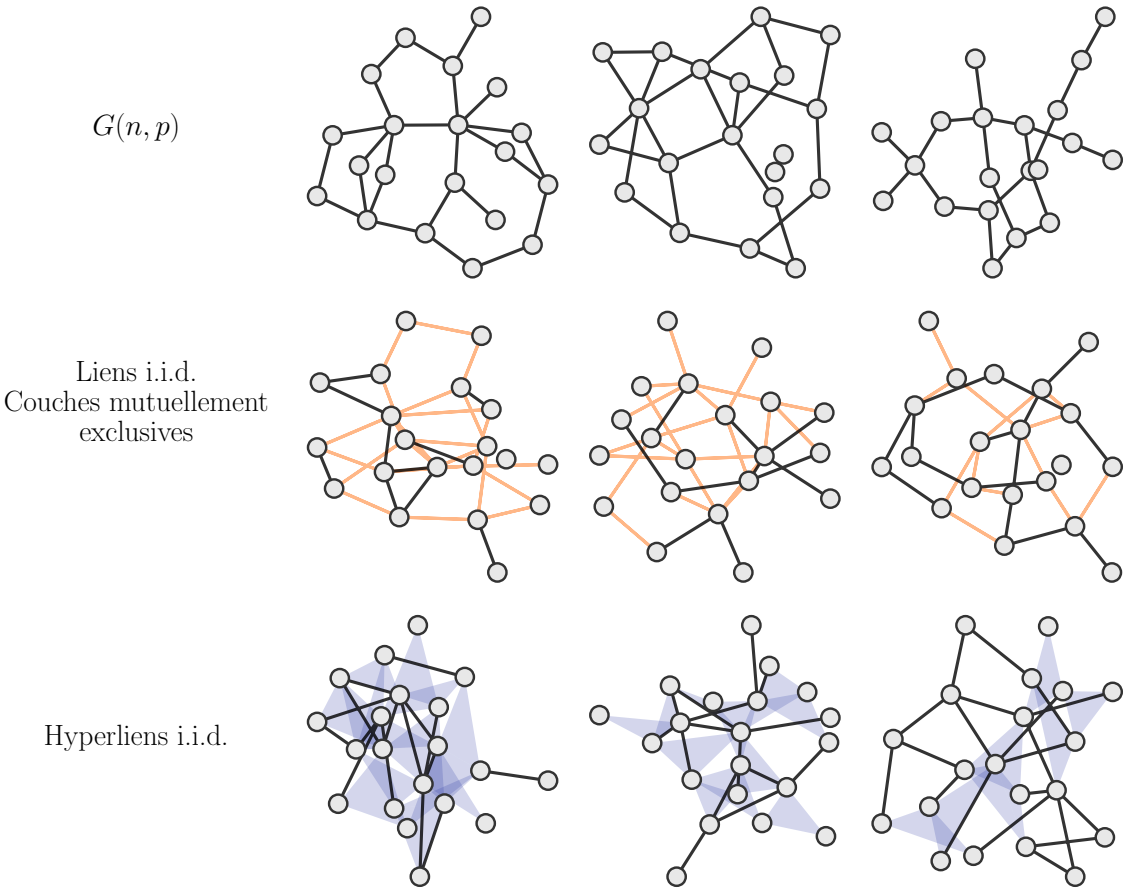


Figure 1.4 : Réalisations de graphes et d'hypergraphes aléatoires de 20 noeuds. La première ligne présente le graphe aléatoire de l'équation (1.24) avec $p = 0.1$, la deuxième ligne le graphe multiplexe aléatoire de l'équation (1.25) à deux couches avec $p_1 = p_2 = 0.1$ et la dernière ligne l'hypergraphe aléatoire de l'équation (1.27) avec $p_2 = 0.1$ et $p_3 = 0.01$. En noir sont représentés les liens, les liens de la première couche et les 2-liens ; en orange sont illustrés les liens de la deuxième couche ; en mauve sont présentés les 3-liens.

de revue de Drobyshevskiy et Turdakov [55] et l'article de revue de Goldenberg et al. [56] qui explorent une grande variété de modèles. Du côté des hypergraphes, plusieurs modèles aléatoires sont décrits dans l'article de revue de Battiston et al. [23].

1.3.1 Modèle $G(n,p)$

Le graphe aléatoire le plus simple est sans doute le modèle $G(n,p)$, aussi appelé le modèle de Gilbert [57]. Dans celui-ci, les éléments du triangle inférieur de la matrice d'adjacence A d'un graphe de n noeuds sont considérés i.i.d. selon une loi de Bernoulli $a_{ij} \quad \text{Bern}(p)$. De cette façon, la fonction de masse du graphe aléatoire est

$$\mathbb{P}(G) = \prod_{i<j} \mathbb{P}(a_{ij}) = \prod_{i<j} p^{a_{ij}}(1-p)^{1-a_{ij}} = p^m(1-p)^{\binom{n}{2}-m}, \tag{1.24}$$

où $m = \sum_{i<j} a_{ij}$ est le nombre de liens dans le graphe et où les indices $i < j$ sont les combinaisons de noeuds (i, j) telles que $1 \leq i < j \leq n$ (par exemple, $\sum_{i<j} = \sum_{i=1}^{n} \sum_{j=i+1}^{n}$). Puisqu'il existe $\binom{n}{2}$ combinaisons de paires de noeuds possibles, $\sum_{i<j}(1 - a_{ij}) = \binom{n}{2} - m$. La figure 1.4 présente des réalisations de ce graphe aléatoire.

1.3.2 Généralisation aux graphes multiplexes et aux hypergraphes

En considérant les différentes couches comme des graphes indépendants, le modèle $G(n, p)$ peut être directement généralisé aux graphes multiplexes. De cette manière, chaque couche c possède sa probabilité d'existence d'un lien p_c

$$\mathbb{P}(G) = \prod_{c} p^{m_c}(1 - p_c)^{\binom{n}{2} - m_c} \tag{1.25}$$

où m_c est le nombre de liens dans la couche c.

Dans une approche de reconstruction proposée au chapitre 2, les liens de couches différentes sont supposés mutuellement exclusifs. Pour satisfaire cette contrainte, la loi présentée peut être ajustée pour ne placer que des liens entre des paires non connectées dans les couches précédentes

$$\mathbb{P}(G) = \prod_{c} p_c^{m_c}(1 - p)^{\binom{n}{2} - \sum_{c'=1}^{c} m}, \tag{1.26}$$

où $a_{ij}^{(c)}$ dénote l'élément (i, j) de la matrice d'adjacence de la couche c. Cependant, les couches ne sont plus interchangeables dans ce modèle : une couche c peut former plus de liens qu'une couche supérieure $c' > c$. La figure 1.4 illustre des réalisations de ce graphe multiplexe aléatoire.

Du côté des hypergraphes, une généralisation du modèle $G(n, p)$ correspond à assigner une probabilité p_k indépendante aux k-liens. Puisque la taille maximale d'un hyperlien est n (aucun noeud redondant), on obtient en excluant les 1-liens que

$$\begin{aligned}\mathbb{P}(H) &= \prod_{k=2}^{n} \prod_{(i_1,\ldots,i_k) \in C_k^n} p_k^{\mathbb{1}_E((i_1,\ldots,i_k))}(1 - p_k)^{1 - \mathbb{1}_E((i_1,\ldots,i_k))} \\ &= \prod_{k=2}^{n} p_k^{h_k}(1 - p_k)^{\binom{n}{k} - h_k},\end{aligned} \tag{1.27}$$

où C_k^n est l'ensemble des combinaisons de taille k parmi les n noeuds et où h_k est le nombre de k-liens dans l'hypergraphe. La figure 1.4 illustre des réalisations de cet hypergraphe aléatoire.

1.3.3 Hypergraphes aléatoires plus structurés

Les connexions d'un système sont généralement plus structurées que de simples interactions distribuées indépendamment. Cette section présente brièvement quelques modèles d'hypergraphes plus réalistes qui seront analysés dans le chapitre 2.

Utilisé par Paul et al. [58], le modèle stochastique par blocs superposés ajoute une structure en communautés aux hypergraphes en assignant à chaque noeud i une communauté b_i. En supposant que des noeuds d'un même bloc r sont similaires, ceux-ci ont une probabilité $p_{k,r}$ d'être connectés par un k-lien tandis que des noeuds de communautés différentes ont une probabilité q_k d'être connectés. La probabilité d'un hypergraphe est alors

$$\mathbb{P}(H) = \prod_{k=2} \prod_{(i_1,\ldots,i_k)} \prod_{C_k^n} \rho_{k,r}^{\mathbb{1}_E((i_1,\ldots,i_k))} (1-\rho_{k,r})^{1-\mathbb{1}_E((i_1,\ldots,i_k))} \tag{1.28}$$

où

$$\rho_{k,r} = \begin{cases} p_{k,r} & \text{si } b_{i_1} = \cdots = b_{i_k} = r \\ q_k & \text{autrement.} \end{cases} \tag{1.29}$$

Un modèle des configurations d'hypergraphes, proposé par Miller et al. [59], fixe le nombre de k-liens $h_{i,k}$ contenant chaque noeud i. Pour générer un hypergraphe avec cette contrainte, $h_{i,k}$ embouts de k-liens sont créés pour chaque noeud i, puis les k-liens sont créés en sélectionnant uniformément k de ces embouts.

Présenté par Stasi et al. [60], le modèle β d'hypergraphe modélise la prédisposition d'un noeud à former des interactions. Dans cet hypergraphe aléatoire, chaque noeud i possède un scalaire $\beta_i^{(k)}$ contrôlant sa probabilité de former un k-lien

$$\mathbb{P}(H) = \prod_{k=2} \prod_{(i_1,\ldots,i_k)} \prod_{C_k^n} (p_{i_1,\ldots,i_k})^{\mathbb{1}_E((i_1,\ldots,i_k))} (1-p_{i_1,\ldots,i_k})^{1-\mathbb{1}_E((i_1,\ldots,i_k))} \tag{1.30}$$

où

$$p_{i_1,\ldots,i_k} = \frac{e^{\beta_{i_1}^{(k)}+\ldots\beta_{i_k}^{(k)}}}{1+e^{\beta_{i_1}^{(k)}+\ldots\beta_{i_k}^{(k)}}}.$$

1.4 Inférence bayésienne

En science, un des objectifs principaux est d'infirmer des théories avec données provenant d'un environnement contrôlé [61]. C'est dans ce cadre que s'inscrit l'inférence statistique, un domaine portant sur la déduction de caractéristiques d'une population en vue de l'obtention de conclusions à partir d'une sous-population limitée. Les méthodes développées dans ce champ de recherche sont d'une grande valeur pour les scientifiques en raison des nuances quantitatives qu'elles apportent aux conclusions.

L'inférence bayésienne, une approche d'inférence statistique, permettra dans ce mémoire d'effectuer la reconstruction de graphes et d'hypergraphes à partir de mesures bruitées. Cette section introduit les notions de base de l'inférence bayésienne de même que quelques concepts importants de modélisation qui sont nécessaires au projet de recherche.

1.4.1 Principe

Dans un modèle statistique, une expérience aléatoire est supposée avoir généré les données observées. Les observations pourraient donc être différentes à chaque mesure du système. Dans ce contexte, un modèle statistique est défini par la probabilité $\mathbb{P}(X|\theta)$ que la loi de probabilité paramétrée par θ ait généré les données X, une quantité appelée la *vraisemblance*. Si, par exemple, on souhaite analyser les tirs d'une pièce de monnaie $(X_i)_{i=1}^N$ où X_i est le résultat du i-ième lancer, on peut supposer que ceux-ci sont des réalisations de variables aléatoires i.i.d. d'une loi de Bernoulli de paramètre θ. En choisissant arbitrairement que $X_i = 0$ représente « face » et que $X_i = 1$ représente « pile », la vraisemblance s'écrit $\mathbb{P}(X|\theta) = \prod_i \theta^{X_i}(1-\theta)^{1-X_i}$. L'intérêt des modèles statistiques est que si le *vrai* paramètre θ de la pièce était connu, il serait possible de prédire la probabilité qu'un nouveau tir de la pièce tombe du côté « pile » ou du côté « face ».

Pour *estimer* la valeur de ce vrai paramètre θ, on utilise un *estimateur*. Un estimateur $\hat{\theta}$ est une fonction permettant d'évaluer un paramètre inconnu θ à partir de données (voir la prochaine section pour un exemple). En statistique classique, l'estimateur est habituellement le paramètre maximisant la vraisemblance, c'est-à-dire le paramètre le mieux adapté aux données observées. La statistique bayésienne n'est pas forcément limitée à déterminer un estimateur de θ : elle fournit la *distribution* des paramètres du modèle qui caractérisent les données.

En inférence bayésienne, le paramètre θ est supposé aléatoire, car il est inconnu. Bien qu'à première vue cette approche puisse sembler contre-intuitive, elle prend son sens lorsque la probabilité est interprétée comme une incertitude. Cette interprétation est en fait courante : on dit par exemple « il va pleuvoir avec 80% de *probabilité* » ou « je serai *probablement* disponible ». Dans ces contextes, la probabilité représente le degré de certitude par rapport à la réalisation d'un événement, la probabilité qu'une prédiction s'avère correcte.

Ce faisant, en inférence bayésienne, le degré de certitude de chaque valeur de paramètre possible est établi dans la *loi a priori* $\mathbb{P}(\theta)$. Cette loi encode, sous forme d'hypothèse, les valeurs de paramètre qui sont les mieux à même de bien modéliser les données. Toutefois, étant une hypothèse, la loi *a priori* ne doit généralement pas être déterminée directement des données. Dans l'exemple d'un tir de pièce de monnaie, on pourrait supposer *a priori* que θ est aux alentours de 0.5 ou encore que toutes les valeurs sont équiprobables.

La loi *a priori* est une des forces de l'inférence bayésienne par rapport à l'inférence fréquentiste. Comme il sera illustré dans l'exemple de la section 1.4.2, cette loi ajoute des données fictives au processus d'inférence, ce qui permet d'obtenir de meilleurs résultats pour des petits jeux de données (dans la mesure où les données fictives sont cohérentes avec les données).

Une fois la loi *a priori* formulée, elle est ajustée aux données X à partir de la vraisemblance.

Concrètement, la vraisemblance est combinée à la loi *a priori* à l'aide du théorème de Bayes

$$\underbrace{\mathbb{P}(\theta|X)}_{\text{loi } a \text{ posteriori}} = \frac{\overbrace{\mathbb{P}(X|\theta)}^{\text{vraisemblance}} \overbrace{\mathbb{P}(\theta)}^{\text{loi } a \text{ priori}}}{\underbrace{\mathbb{P}(X)}_{\text{évidence}}}, \tag{1.31}$$

où $\mathbb{P}(X)$ est la loi marginale des données, appelée *évidence*, qui apparaît comme constante de normalisation. Ce théorème se démontre aisément à partir de la définition de Kolmogorov de la probabilité conditionnelle (1.7).

Démonstration – Théorème de Bayes

Soit deux variables aléatoires discrètes Y et Z munies de la mesure de probabilité $\mathbb{P}$. De la définition (1.7),

$$\mathbb{P}(\{Y = y\}|\{Z = z\}) = \frac{\mathbb{P}(\{Y = y\} \cap \{Z = z\})}{\mathbb{P}(\{Z = z\})}.$$

Puisque $\mathbb{P}(A_1 \cap A_2) = \mathbb{P}(A_2 \cap A_1)$ où A_1 et A_2 sont deux événements,

$$\mathbb{P}(\{Y = y\} \cap \{Z = z\}) = \mathbb{P}(\{Z = z\}|\{Y = y\})\mathbb{P}(\{Y = y\}) \tag{1.32}$$

et

$$\mathbb{P}(\{Y = y\}|\{Z = z\}) = \frac{\mathbb{P}(\{Z = z\}|\{Y = y\})\mathbb{P}(\{Y = y\})}{\mathbb{P}(\{Z = z\})} \tag{1.33}$$

$$f_{Y|Z}(y) = \frac{f_{Z|Y}(z) f_Y(y)}{f_Z(z)}. \tag{1.34}$$

À partir de l'équation (1.10) et du fait que $f_{Y,Z} = f_{Z,Y}$, un raisonnement similaire peut être effectué pour des variables aléatoires continues afin d'obtenir l'équation (1.34) où les fonctions f sont des fonctions de densité.

Comme noté à l'équation (1.31), le résultat de l'inférence bayésienne est la *loi a posteriori* $\mathbb{P}(\theta|X)$. Effectuant un compromis entre les paramètres ajustés aux données et leur probabilité *a priori*, elle offre la probabilité que l'expérience aléatoire ait généré les données à partir des paramètres θ.

Une propriété intéressante de la distribution *a posteriori* est que sa variance conditionnelle est en moyenne plus petite que la variance de la loi *a priori*. En effet,

$$\begin{aligned}\mathbb{V}[\theta] &= \mathbb{E}\left[\theta^2\right] - \mathbb{E}[\theta]^2 \\ &= \mathbb{E}\left[\mathbb{E}\left[\theta^2|X\right]\right] - \mathbb{E}[\mathbb{E}[\theta|X]]^2\end{aligned} \tag{1.35}$$

par la propriété de l'espérance conditionnelle (1.20). Par la définition de la variance et par la linéarité de l'espérance,

$$\begin{aligned}\mathbb{V}[\theta] &= \mathbb{E}\left[\mathbb{V}[\theta|X]\right] + \mathbb{E}\left[\mathbb{E}[\theta|X]^2\right] - \mathbb{E}[\mathbb{E}[\theta|X]]^2 \\ &= \mathbb{E}[\mathbb{V}[\theta|X]] + \mathbb{E}\left[\mathbb{E}[\theta|X]^2\right] - \mathbb{E}[\mathbb{E}[\theta|X]]^2 \\ &= \mathbb{E}[\mathbb{V}[\theta|X]] + \mathbb{V}[\mathbb{E}[\theta|X]]. \end{aligned} \tag{1.36}$$

Puisque la variance est une quantité strictement positive, on obtient

$$\mathbb{E}[\mathbb{V}[\theta|X]] < \mathbb{V}[\theta]. \tag{1.37}$$

Ce résultat met en lumière l'importance de sélectionner une loi *a priori* appropriée. D'un côté, si la loi *a priori* a une faible variance et est bien choisie, alors l'information supplémentaire limite la variance *a posteriori*. De l'autre, si la loi *a priori* est mal adaptée aux données, alors la masse de probabilité se concentre autour de paramètres peu probables *a posteriori*.

1.4.2 Exemple : Modèle de Bernoulli

Supposons qu'on possède une séquence $(X_i)_{i=1}^N$ de tirs d'une même pièce de monnaie. Si la pièce i tombe du côté pile, X_i vaut 1, et si elle tombe du côté face, X_i vaut 0. En modélisant les tirs de la pièce comme des variables aléatoires de Bernoulli i.i.d. $X_i \sim \text{Bern}(p)$, la vraisemblance de la séquence est donnée par

$$\mathbb{P}(X_1, \ldots, X_N|p) = \prod_{i=1}^N \mathbb{P}(X_i|p) = \prod_{i=1}^N p^{X_i}(1-p)^{1-X_i} = p^K(1-p)^{N-K}, \tag{1.38}$$

où $K := \sum_{i=1}^N X_i$ est le nombre de tirs ayant comme résultat pile.

Comme énoncé à la section précédente, l'estimateur classique d'un paramètre en statistique correspond au maximum de vraisemblance

$$\hat{p}_{\text{emv}} = \operatorname*{argmax}_p \mathbb{P}(X|p) \tag{1.39}$$

(« emv » pour estimateur du maximum de vraisemblance). Étant donné que le logarithme est une fonction monotone croissante sur les réels positifs, le logarithme de la vraisemblance est maximisé

$$\ln p^K(1-p)^{N-K} = K \ln p + (N-K)\ln(1-p). \tag{1.40}$$

Ainsi,

$$\begin{aligned} 0 &= \frac{\partial}{\partial p}\Big[K\ln p + (N-K)\ln(1-p)\Big]_{p=\hat{p}_{\text{emv}}} \\ 0 &= K\frac{1}{\hat{p}_{\text{emv}}} - (N-K)\frac{1}{1-\hat{p}_{\text{emv}}} \\ 0 &= (\hat{p}_{\text{emv}}-1)K + \hat{p}_{\text{emv}}(N-K) \\ \hat{p}_{\text{emv}} &= \frac{K}{N} = \frac{1}{N}\sum_{i=1}^{N} X_i. \end{aligned} \tag{1.41}$$

Le point $\hat{p}_{\text{emv}}$ est un maximum, car

$$\frac{\partial^2}{\partial p^2}\ln \mathbb{P}(X_1, X_2, \ldots, X_N|p) = -K\frac{1}{p^2} - (N-K)\frac{1}{(1-p)^2} < 0 \tag{1.42}$$

où les deux termes sont toujours négatifs.

Pour poursuivre un traitement bayésien du problème, une loi *a priori* doit être attribuée au paramètre p. Dans cet exemple, on utilise une loi bêta(α, β)

$$\mathbb{P}(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1}, \tag{1.43}$$

où α et β sont des constantes fixées dans le modèle, appelées *hyperparamètres*. Bien qu'une autre distribution *a priori* puisse être choisie, la loi bêta est la loi conjuguée de ce modèle (concept expliqué à la section 1.4.3), ce qui simplifie les calculs explicites.

Ayant en main la vraisemblance et la loi *a priori*, la loi *a posteriori* est

$$\begin{aligned} \mathbb{P}(p|X) &= \frac{1}{\mathbb{P}(X)}p^K(1-p)^{N-K}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1} \\ &= \frac{1}{\mathbb{P}(X)}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{K+\alpha-1}(1-p)^{N-K+\beta-1}. \end{aligned} \tag{1.44}$$

Avec un léger travail algébrique, la constante de normalisation est

$$\begin{aligned} \mathbb{P}(X) &= \int \mathbb{P}(X|p)\mathbb{P}(p)\mathrm{d}p = \int p^K(1-p)^{N-K}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1}\mathrm{d}p \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\int p^{K+\alpha-1}(1-p)^{N-K+\beta-1}\mathrm{d}p \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(\alpha+K)\Gamma(\beta+N-K)}{\Gamma(\alpha+\beta+N)} \end{aligned} \tag{1.45}$$

La loi *a posteriori* est alors une loi bêta $p|X \quad$ bêta$(\alpha+K, \beta+N-K)$

$$\mathbb{P}(p|X) = \frac{\Gamma(\alpha+\beta+N)}{\Gamma(\alpha+K)\Gamma(\beta+N-K)}p^{K+\alpha-1}(1-p)^{N-K+\beta-1}. \tag{1.46}$$

Or, ce résultat se trouve directement à partir de l'équation (1.44). En effet, l'expression ayant déjà la forme d'une loi bêta, la constante de normalisation $\mathbb{P}(X)$ pouvait en être déduite.

Par ailleurs, il n'est généralement pas nécessaire de calculer $\mathbb{P}(X)$ lorsque la loi *a priori* est conjuguée.

Dans cet exemple on choisit l'espérance de la loi *a posteriori* comme estimateur bayésien. La particularité de cet estimateur est qu'il minimise l'erreur quadratique :

$$\begin{aligned} 0 &= \frac{\partial}{\partial \hat{p}} \mathbb{E}\left[(p-\hat{p})^2 | X\right]\Big|_{\hat{p}=\hat{p}_{\text{Bayes}}} \\ &= \int_{\text{supp}(p)} \frac{\partial}{\partial \hat{p}} (p-\hat{p})^2 \mathbb{P}(p|X)\Big|_{\hat{p}=\hat{p}_{\text{Bayes}}} \mathrm{d}p \\ &= -2\int_{\text{supp}(p)} (p-\hat{p}_{\text{Bayes}})\mathbb{P}(p|x)\mathrm{d}p, \\ &= -2\mathbb{E}[p|X] + 2\hat{p}_{\text{Bayes}} \\ \hat{p}_{\text{Bayes}} &= \mathbb{E}[p|X], \end{aligned} \tag{1.47}$$

ce qui est un minimum puisque

$$\frac{\partial^2}{\partial \hat{p}^2}\mathbb{E}\left[(p-\hat{p})^2|X\right] = 2 > 0. \tag{1.48}$$

Il existe d'autres estimateurs comme le mode et la médiane, mais on se limite à l'espérance dans cette section. L'estimateur pour cet exemple est alors

$$\hat{p}_{\text{Bayes}} = \mathbb{E}[p|X] = \frac{\alpha + K}{\alpha + \beta + N}. \tag{1.49}$$

En utilisant la paramétrisation $K_0 := \alpha$ et $N_0 := \alpha + \beta$, on remarque que ces nouveaux hyperparamètres s'interprètent comme des données fictives ajoutées aux données X

$$\hat{p}_{\text{Bayes}} = \frac{K_0 + K}{N_0 + N}. \tag{1.50}$$

K_0 est donc le nombre de tirs tombés sur pile et N_0 le nombre de tirs total *a priori*. De plus, $\hat{p}_{\text{Bayes}}$ peut s'interpréter comme étant une moyenne pondérée de l'estimateur des données fictives $\hat{p}_0 = \frac{K_0}{N_0}$ et l'estimateur de maximum de vraisemblance $\hat{p}_{\text{emv}}$

$$\hat{p}_{\text{Bayes}} = \frac{1}{N_0 + N}(N_0\hat{p}_0 + N\hat{p}_{\text{emv}}). \tag{1.51}$$

Ainsi, l'estimateur bayésien est un compromis entre la loi *a priori* et la vraisemblance des données. En d'autres mots, les données apportent une correction aux hypothèses. Dans la limite où $N \to \infty$, $\hat{p}_{\text{Bayes}} \to \hat{p}_{\text{emv}}$.

Une autre remarque intéressante est que si $\alpha = \beta = 1$, alors $\mathbb{P}(p)$ est une loi uniforme continue et $\hat{p}_0 = \frac{1}{2}$. Cet estimateur maximise l'entropie de Shannon, ce qui signifie qu'il minimise l'information supposée sur le paramètre p. En d'autres mots, il s'agit du seul estimateur *a priori* non biaisé [62].

On réitère ici l'importance du choix de la loi *a priori*. Il est préférable d'introduire moins d'information *a priori* (une plus grande variance) pour éviter que la distribution *a posteriori* soit déformée par un mauvais choix. Si $\hat{p}_0$ est une mauvaise estimation pour un grand N_0 (variance de $\hat{p}_0$ plus petite) par exemple, alors un grand nombre de données N est nécessaire pour obtenir un bon estimateur $\hat{p}_{\text{Bayes}}$.

1.4.3 Loi *a priori* conjuguée

Le calcul de loi *a posteriori* de l'exemple précédent n'est toutefois pas représentatif de la plupart des calculs en inférence bayésienne. Effectivement, la loi *a posteriori* ne prend généralement pas la forme d'une loi connue. Si on utilise une loi *a priori* normale tronquée sur l'intervalle $[0, 1]$ dans le modèle de la section précédente, alors

$$\mathbb{P}(p) = \frac{1}{\Phi(1) - \Phi(0)} \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(p-\mu)^2}{2\sigma^2}}, \tag{1.52}$$

où $\Phi(\cdot)$ est la fonction de répartition de la loi normale. La loi *a posteriori*

$$\mathbb{P}(p|X) = \frac{1}{\mathbb{P}(X)} \frac{1}{[\Phi(1) - \Phi(0)]} \sqrt{\frac{1}{2\pi\sigma^2}} p^K (1-p)^{N-K} e^{-\frac{(p-\mu)^2}{2\sigma^2}} \tag{1.53}$$

$$p^K (1-p)^{N-K} e^{-\frac{(p-\mu)^2}{2\sigma^2}}. \tag{1.54}$$

n'a ainsi plus une forme connue, ce qui rend moins direct le calcul de la constante de normalisation $\mathbb{P}(X) = \int \mathbb{P}(X|p)\mathbb{P}(p)\mathrm{d}p$. Il peut alors devenir très difficile, voire impossible, d'effectuer des calculs analytiques sur cette distribution (espérance, variance, quartiles, etc.).

Une manière d'éviter cette situation est de choisir une loi *a priori* qui se combine algébriquement à la vraisemblance. À cette fin, il existe les lois *conjuguées*, lois *a priori* pour lesquelles la loi *a posteriori* est de la même famille. À la section 1.4.2 par exemple, la loi conjuguée du paramètre p est une loi bêta, car la loi *a posteriori* est aussi une loi bêta. Une discussion plus détaillée sur les familles de lois *a priori* conjuguées se trouve dans le document *A Compendium of Conjugate Priors* [63].

Toutefois, il n'existe pas de loi *a priori* conjuguée connue pour les paramètres de toutes les vraisemblances. Ce faisant, l'évidence est souvent inconnue, et donc certains calculs analytiques sont très difficiles.

1.4.4 Loi *a priori* impropre

Dans de nombreux contextes, il est raisonnable de considérer une loi *a priori* uniforme sur un paramètre. En effet, elle ne favorise aucune valeur de paramètre, ce qui laisse les observations préciser lesquelles sont réalistes. Cependant, lorsque le support est infini, une loi de probabilité uniforme est mal définie.

Considérons un nouveau modèle où les observations sont supposées i.i.d. d'une loi de Poisson de paramètre λ, $X_i \sim \text{Poi}(\lambda) \quad i$

$$\mathbb{P}(X|\lambda) = \prod_{i=1}^{N} \frac{\lambda^{X_i}}{X_i!} e^{-\lambda}. \tag{1.55}$$

Ici, puisque le support du paramètre λ est infini $\lambda \in (0, \infty)$, une loi uniforme sur cet intervalle est impossible à normaliser. Une loi *a priori* dont la constante de normalisation est infinie est appelée une loi *impropre* et une loi dont la constante est finie est appelée une loi *propre*.

Contrairement à ce que les mathématiques semblent prescrire, une loi *a priori* impropre est parfois acceptable. Toutefois, lorsqu'une loi impropre est utilisée : la loi *a posteriori doit* être propre. Autrement, elle est également mal définie.

En supposant que λ est une loi uniforme impropre dans le modèle de l'équation (1.55), la loi *a posteriori* est

$$\mathbb{P}(\lambda|X) = \frac{\mathbb{P}(X|\lambda)\mathbb{P}(\lambda)}{\mathbb{P}(X)} \propto \prod_{i=1}^{N} \lambda^{X_i} e^{-\lambda} \quad \lambda^{\sum_{i=1}^{N} X_i} e^{-N\lambda}, \tag{1.56}$$

c'est-à-dire une loi Gamma($\sum_{i=1}^{N} X_i + 1, N$). Comme la loi Gamma est bien définie pour ces paramètres, la loi *a posteriori* est propre et la loi uniforme sur λ est acceptable.

On remarque qu'une loi *a priori* impropre est souvent une forme limite d'une loi conjuguée. Dans ce modèle de loi de Poisson, la loi *a priori* conjuguée du paramètre λ est une loi Gamma

$$\mathbb{P}(\lambda) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}. \tag{1.57}$$

Avec cette loi *a priori*, la loi *a posteriori* est

$$\mathbb{P}(\lambda|X) = \frac{\mathbb{P}(X|\lambda)\mathbb{P}(\lambda)}{\mathbb{P}(X)} \propto \left[\prod_{i=1}^{N} \lambda^{X_i} e^{-\lambda}\right] \left[\lambda^{a-1} e^{-b\lambda}\right] \lambda^{\sum_{i=1}^{N} X_i + a - 1} e^{-(N+b)\lambda}, \tag{1.58}$$

soit une loi Gamma($\sum_{i=1}^{N} X_i + a, N + b$). Par comparaison, la loi *a priori* uniforme correspond aux hyperparamètres $a = 1$ et $b = 0$, une paramétrisation non permise de la loi Gamma (b est strictement positif).

Pour certains modèles, il est difficile de vérifier analytiquement que la loi *a posteriori* est propre. Certains choisissent de négliger cet aspect, mais il est plus judicieux d'utiliser une autre loi *a priori*. Par ailleurs, un comportement similaire à une loi uniforme s'obtient avec une loi *a priori* propre : il suffit de supposer une très grande variance *a priori*. Par exemple puisque la variance de la loi Gamma est a/b^2, on peut choisir un grand a ou un petit b. Pour un traitement numérique, un petit b est préférable à un grand a, car réduire b accroît la variance plus rapidement que l'espérance $\mathbb{E}[\lambda] = a/b$.

1.4.5 Indépendance des paramètres

Une autre hypothèse commune dans les modèles bayésiens suppose l'indépendance *a priori* des paramètres. Dans un modèle de loi normale par exemple, il est raisonnable de supposer l'indépendance entre la moyenne μ et la variance σ^2. Or, est-ce que l'hypothèse d'indépendance *a priori* implique l'indépendance des paramètres *a posteriori* ?

Considérons un modèle bayésien quelconque de paramètres θ_1 et θ_2 qui modélise des données X. En supposant θ_1 et θ_2 indépendants *a priori*, la loi *a posteriori* est

$$\mathbb{P}(\theta_1, \theta_2|X) = \frac{\mathbb{P}(X|\theta_1, \theta_2)\mathbb{P}(\theta_1, \theta_2)}{\mathbb{P}(X)},$$
$$\mathbb{P}(X|\theta_1, \theta_2)\mathbb{P}(\theta_1)\mathbb{P}(\theta_2). \tag{1.59}$$

L'équation (1.59) illustre que θ_1 et θ_2 ne sont pas forcément indépendants *a posteriori*. Plus précisément, les paramètres sont indépendants *a posteriori* seulement si la vraisemblance se sépare en un produit de deux fonctions $P(X|\theta_1, \theta_2) = f_1(X, \theta_1)f_2(X, \theta_2)$

$$\mathbb{P}(\theta_1, \theta_2|X) \quad f_1(X, \theta_1)\mathbb{P}(\theta_1) \quad f_2(X, \theta_2)\mathbb{P}(\theta_2)$$

$$= \quad \mathbb{P}(\theta_1, \theta_2|X) = \mathbb{P}(\theta_1|X)\mathbb{P}(\theta_2|X). \tag{1.60}$$

1.4.6 Modèle hiérarchique

Dans les modèles bayésiens, les lois *a priori* sont sensibles aux hyperparamètres choisis et parfois, certains hyperparamètres sont mieux décrits par d'autres quantités physiques. En vue d'améliorer la robustesse et l'interprétabilité de l'inférence, on introduit les modèles hiérarchiques.

Un modèle *hiérarchique* est un modèle dont les hyperparamètres sont aléatoires plutôt que fixés. Dans l'exemple de la pièce de monnaie de la section 1.4.2, les hyperparamètres α et β de la loi *a priori* sur p pourraient être supposés aléatoires, de sorte que la loi s'écrive $\mathbb{P}(p|\beta, \alpha)$. De plus, puisque les hyperparamètres sont aléatoires, une loi *a priori* $\mathbb{P}(\alpha, \beta)$ leur est également attribuée. De cette manière, le modèle bayésien devient

$$\mathbb{P}(p, \alpha, \beta|X) = \frac{\mathbb{P}(X|\theta)\mathbb{P}(\theta|\alpha, \beta)\mathbb{P}(\alpha, \beta)}{\mathbb{P}(X)}. \tag{1.61}$$

Étant donné que la loi *a priori* sur les hyperparamètres dépend aussi d'hyperparamètres, il est possible de poursuivre cette hiérarchie (un nombre fini de fois cela dit) en considérant comme aléatoires ces nouveaux hyperparamètres.

D'un côté, en supposant une loi *a priori* sur un hyperparamètre, celui-ci n'a plus besoin d'être fixé, ce qui permet d'« adoucir » la loi. De l'autre, si l'hyperparamètre peut s'exprimer par un ou une combinaison d'autres hyperparamètres physiques, ces derniers sont également inférés par le modèle bayésien, ce qui facilite l'interprétation des résultats.

1.4.7 Modèle de mélange et symétries *a posteriori*

Une des classes de modèles polyvalents en inférence est le *modèle de mélange*. Ceux-ci permettent de classifier une séquence de données $(X_i)_{i=1}^N$. Dans un modèle de mélange, X_i est distribué selon une certaine loi $f_1(X_i)$ si celui-ci est de la classe C_1 tandis qu'il est distribué selon une loi $f_2(X_i)$ s'il appartient à la classe C_2. Mathématiquement, la vraisemblance d'une donnée X_i est

$$\mathbb{P}(X_i|c_i,\theta) = \mathbb{1}_{\{c_i=1\}}(c_i)f_1(X_i) + \mathbb{1}_{\{c_i=2\}}(c_i)f_2(X_i) \tag{1.62}$$

où θ sont les paramètres de f_1 et f_2 et où c_i est la classe de X_i. De manière générale, la vraisemblance d'une donnée est

$$\mathbb{P}(X_i|c_i,\theta) = \sum_{k=1}^{K} \mathbb{1}_{\{c_i=k\}}(c_i)f_k(X_i) \tag{1.63}$$

pour un modèle à K classes. En l'absence d'information précise sur la classe de chaque donnée, cette dernière est marginalisée de sorte que les X_i observés sont distribués selon la loi de mélange

$$\mathbb{P}(X_i|\theta) = \sum_{k=1}^{K} \mathbb{P}(c_i = k)f_k(X_i). \tag{1.64}$$

Avec ce type de modèle, la loi *a posteriori* obtenue nous renseigne sur la probabilité de chacune des classes. Cette forme de classification procure l'avantage d'être nuancée contrairement à l'approche de maximisation de la vraisemblance.

Cependant, cette famille de modèles possède un problème intrinsèque : la symétrie de la loi *a posteriori*. En effet, si plusieurs distributions f_i sont identiques (des lois de Poisson, des lois normales, etc.), les étiquettes des classes sont indiscernables. Si, par exemple, un modèle admet une classe A avec f_A = Poisson(λ_A) et une classe B avec f_B = Poisson(λ_B), alors on peut échanger λ_A et λ_B (voir figure 1.5). Cette symétrie est problématique, car elle rend la loi *a posteriori* multimodale, de sorte que les statistiques de tendance centrale (espérance, médiane et mode) n'aient plus de sens.

Une solution à ce problème consiste à imposer une relation d'ordre entre les paramètres. En effet, sans perte de généralité, on peut supposer que $\lambda_A < \lambda_B$. Mathématiquement, ceci est effectué en tronquant le support d'un paramètre par rapport à l'autre dans la loi *a priori*

$$\mathbb{P}(\lambda_A|\lambda_B) \quad g(\lambda_A)\mathbb{1}_{\{\lambda_A<\lambda_B\}}, \tag{1.65}$$

où g est une densité *a priori* quelconque.

1.4.8 Distribution prédictive *a posteriori*

Une des qualités d'un modèle est son habileté à prédire les comportements à venir. En inférence bayésienne, ces prédictions sont effectuées avec la *distribution prédictive a posteriori*. Cette

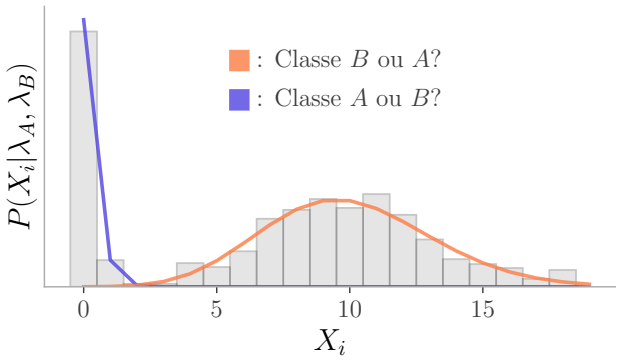


Figure 1.5 : Modèle de mélange comportant une symétrie sur les paramètres. La vraisemblance est un mélange de deux lois de Poisson avec paramètres 0.5 et 10. Il est impossible de déterminer si $\lambda_A = 0.5$ et $\lambda_B = 10$ ou $\lambda_A = 10$ et $\lambda_B = 0.5$.

distribution, notée $\mathbb{P}(\tilde{X}|X)$, est la probabilité de générer les données $\tilde{X}$ conditionnellement aux données mesurées X

$$\mathbb{P}(\tilde{X}|X) = \int \mathbb{P}(\tilde{X}, \theta|X)\mathrm{d}\theta = \int \mathbb{P}(\tilde{X}|\theta)\mathbb{P}(\theta|X)\mathrm{d}\theta, \tag{1.66}$$

où $\mathbb{P}(\tilde{X}|\theta, X) = \mathbb{P}(\tilde{X}|\theta)$ est la vraisemblance. La variable aléatoire $\tilde{X}$ est conditionnellement indépendante à X, car θ contient l'information nécessaire au modèle pour générer les données. Pour obtenir un échantillon de $\mathbb{P}(\tilde{X}|X)$, il suffit de générer des données $\tilde{X}^{(i)}|\theta^{(i)}$ avec la vraisemblance conditionnée sur des paramètres tirés de la loi *a posteriori* $\theta^{(i)}|X$. La génération numérique d'un échantillon est abordée aux sections 1.6 et 1.7.

Dans ce projet de recherche, la distribution prédictive *a posteriori* est utilisée comme une forme de validation des résultats obtenus : on s'attend à ce que $\tilde{X}$ ressemble à X suite au processus d'inférence. Il s'agit d'une approche privilégiée dans l'ouvrage *Bayesian Data Analysis* [64].

1.5 Reconstruction de graphes par inférence bayésienne

Les outils présentés dans les sections précédentes nous permettent maintenant de définir un modèle bayésien de reconstruction de graphes. Cette section présente la méthode de Young et al. [22].

1.5.1 Modélisation bayésienne

Typiquement, les interactions d'un système ne peuvent pas être observées directement comme ceci a été illustré à l'introduction avec le régime alimentaire d'animaux. Dans plusieurs systèmes, une manière de mesurer l'importance d'une interaction entre deux éléments consiste à compter le nombre de fois que ceux-ci interagissent dans une période de temps donnée. Ces

observations sont regroupées dans une matrice X dont les éléments x_{ij} contiennent le nombre d'interactions mesurées entre les noeuds i et j. En supposant qu'il existe un graphe d'interactions sous-jacent au système, l'idée de Young et al. est de considérer que si deux noeuds interagissent ($a_{ij} = 1$), leurs interactions sont mesurées plus souvent que s'ils n'interagissent pas ($a_{ij} = 0$). L'objectif est ainsi de déduire d'une mesure x_{ij} la probabilité que deux éléments i et j du système soient connectés par un lien.

Pour ce faire, les auteurs modélisent les mesures x_{ij} d'une paire (i, j) par un modèle de mélange dont la classe est déterminée par la présence ou l'absence d'un lien dans le graphe

$$\mathbb{P}(x_{ij}|a_{ij}, \theta) = (1 - a_{ij})f_0(x_{ij}) + a_{ij}f_1(x_{ij}). \tag{1.67}$$

Puisque la loi Poisson se prête bien aux processus de comptage, les lois f_0 et f_1 sont supposées comme des lois de Poisson de paramètres λ_0 et λ_1 respectivement

$$\mathbb{P}(x_{ij}|a_{ij}, \lambda_0, \lambda_1) = (1 - a_{ij})\frac{\lambda_0^{x_{ij}}}{x_{ij}!}e^{-\lambda_0} + a_{ij}\frac{\lambda_1^{x_{ij}}}{x_{ij}!}e^{-\lambda_1} = \frac{\lambda_{a_{ij}}^{x_{ij}}}{x_{ij}!}e^{-\lambda_{a_{ij}}}. \tag{1.68}$$

Dans ce modèle, si le graphe sous-jacent du système est connu, il est raisonnable de considérer que les observations mesurées expérimentalement entre les noeuds i et j sont indépendantes de celles mesurées entre deux autres noeuds k et . Avec cette supposition, la vraisemblance est

$$\mathbb{P}(X|G, \lambda_0, \lambda_1) = \prod_{i<j} \mathbb{P}(x_{ij}|a_{ij}, \lambda_0, \lambda_1). \tag{1.69}$$

Étant inconnu, le graphe est supposé aléatoire dans l'interprétation bayésienne. Young et al. supposent qu'*a priori*, le graphe est obtenu du modèle $G(n, p)$ présenté à la section 1.3.1

$$\mathbb{P}(G|p) = \prod_{i<j} p^{a_{ij}}(1 - p)^{1-a_{ij}}. \tag{1.70}$$

Puisqu'elles dépendent du contexte des données, les lois *a priori* ne sont pas spécifiées. Cependant, il est important de traiter la symétrie $\lambda_0 \leftrightarrow \lambda_1$, ce qui peut être e□ectué en imposant *a priori* que $\lambda_0 < \lambda_1$. Ceci encode l'hypothèse qu'un lien implique un plus grand nombre de mesures.

En réunissant la vraisemblance et les lois *a priori* de ce modèle hiérarchique, la loi *a posteriori* est la suivante

$$\begin{aligned}\mathbb{P}(G, \lambda_0, \lambda_1, p|X) &= \frac{\mathbb{P}(X|G, \lambda_0, \lambda_1)\mathbb{P}(G|p)\mathbb{P}(\lambda_0, \lambda_1, p)}{\mathbb{P}(X)} \\ &= \frac{\mathbb{P}(\lambda_0, \lambda_1, p)}{\mathbb{P}(X)} \prod_{i<j} \frac{1}{x_{ij}!} \prod_{i<j} \lambda_{a_{ij}}^{x_{ij}} e^{-\lambda_{a_{ij}}} \prod_{i<j} p^{a_{ij}}(1 - p)^{1-a_{ij}}\end{aligned} \tag{1.71}$$

Si les lois *a priori* conjuguées sont retenues pour le modèle, c'est-à-dire que p suit une loi bêta et que λ_0 et λ_1 suivent une loi Gamma, alors une forme fermée s'obtient pour cette loi. Or, malgré l'utilisation d'une loi *a priori* conjuguée, la loi *a posteriori* est d'une forme inconnue et il est di□cile d'évaluer analytiquement ses statistiques.

1.5.2 Estimation de la probabilité des liens

Une manière d'exprimer la reconstruction du graphe est de déterminer la probabilité *a posteriori* d'existence de chacun de ses liens

$$\begin{aligned}\mathbb{E}[a_{ij}|X] &= 0 \cdot \mathbb{P}(a_{ij} = 0|X) + 1 \cdot \mathbb{P}(a_{ij} = 1|X) = \mathbb{P}(a_{ij} = 1|X) \\ &= \sum_G a_{ij}\mathbb{P}(G|X),\end{aligned} \tag{1.72}$$

où la somme énumère l'ensemble des graphes de n noeuds.

Puisqu'il existe au total $2^{\binom{n}{2}}$ graphes de n noeuds, cette somme ne peut généralement pas s'évaluer autant numériquement qu'analytiquement. De plus, pour obtenir la loi $\mathbb{P}(G|X)$, la loi *a posteriori* doit être marginalisée

$$\mathbb{P}(G|X) = \int_0^1 \int_0^\infty \int_0^\infty \mathbb{P}(G, \lambda_0, \lambda_1, p|X)\mathrm{d}\lambda_0\mathrm{d}\lambda_1\mathrm{d}p. \tag{1.73}$$

Bien que ce calcul soit possible analytiquement en choisissant des lois *a priori* conjuguées, la distribution sur les graphes résultante n'a pas la forme d'une loi connue. Il est néanmoins possible d'utiliser cette approche pour estimer l'espérance (1.72) grâce aux méthodes numériques développées à la section 1.7. Elles requièrent cependant un coût computationnel important pouvant être évité grâce à l'approche employée par Young et al.

Dans leur article, les auteurs décomposent la loi *a posteriori* d'une manière alternative

$$\mathbb{P}(G, \theta|X) = \mathbb{P}(G|X, \theta)\mathbb{P}(\theta|X). \tag{1.74}$$

Pour obtenir la loi de $\theta|X$, la loi *a posteriori* est marginalisée sur les graphes

$$\begin{aligned}\mathbb{P}(\theta|X) &= \sum_G \mathbb{P}(G, \theta|X) \\ &= \frac{\mathbb{P}(\theta)}{\mathbb{P}(X)} \prod_{i<j} \frac{1}{x_{ij}!} \sum_G \prod_{i<j} \lambda_{a_{ij}}^{x_{ij}} e^{-\lambda_{a_{ij}}} \prod_{i<j} p^{a_{ij}}(1-p)^{1-a_{ij}} \\ &= \frac{\mathbb{P}(\theta)}{\mathbb{P}(X)} \prod_{i<j} \frac{1}{x_{ij}!} \sum_G \prod_{i<j} p\lambda_0^{x_{ij}} e^{-\lambda_0}{}^{a_{ij}} (1-p)\lambda_1^{x_{ij}} e^{-\lambda_1}{}^{1-a_{ij}}.\end{aligned} \tag{1.75}$$

À partir d'une simple manipulation algébrique, une expression fermée de la somme sur tous les graphes s'obtient. En effet, si b_{ij} représente la quantité associée à un non-lien et c_{ij} représente la quantité associée à un lien, alors le produit du binôme $(a_{ij} + b_{ij})$ permet d'énumérer tous les produits possibles de la forme $(b_{ij})^{a_{ij}}(c_{ij})^{1-a_{ij}}$ d'un graphe

$$\begin{aligned}\sum_G \prod_{i<j} (b_{ij})^{a_{ij}}(c_{ij})^{1-a_{ij}} &= \prod_{i<j} (b_{ij} + c_{ij}) \\ &= (b_{00}b_{01}\cdots) + (c_{00}b_{01}\cdots) + (b_{00}c_{01}\cdots) + (c_{00}c_{01}\cdots) + \ldots.\end{aligned} \tag{1.76}$$

En utilisant cette relation,

$$\mathbb{P}(\theta|X) = \frac{\mathbb{P}(\theta)}{\mathbb{P}(X)} \prod_{i<j} \frac{1}{x_{ij}!} \prod_{i<j} p\lambda_0^{x_{ij}} e^{-\lambda_0} + (1-p)\lambda_1^{x_{ij}} e^{-\lambda_1}. \tag{1.77}$$

En appliquant le théorème de Bayes, la loi de $G|X,\theta$ s'obtient également sous une forme fermée

$$\begin{aligned}
\mathbb{P}(G|X,\theta) &= \frac{\mathbb{P}(X|G,\theta)\mathbb{P}(G|\theta)}{\mathbb{P}(X|\theta)} \\
&= \frac{\mathbb{P}(X|G,\theta)\mathbb{P}(G|\theta)\mathbb{P}(\theta)}{\mathbb{P}(\theta|X)\mathbb{P}(X)} \\
&= \frac{\prod_{i<j} p\lambda_0^{x_{ij}} e^{-\lambda_0}{}^{a_{ij}} (1-p)\lambda_1^{x_{ij}} e^{-\lambda_1}{}^{1-a_{ij}}}{\prod_{i<j} p\lambda_0^{x_{ij}} e^{-\lambda_0} + (1-p)\lambda_1^{x_{ij}} e^{-\lambda_1}} \\
&= \prod_{i<j} Q_{ij}^{a_{ij}} (1-Q_{ij})^{1-a_{ij}}
\end{aligned} \tag{1.78}$$

où

$$Q_{ij} = \frac{p\lambda_0^{x_{ij}} e^{-\lambda_0}}{p\lambda_0^{x_{ij}} e^{-\lambda_0} + (1-p)\lambda_1^{x_{ij}} e^{-\lambda_1}}. \tag{1.79}$$

Les liens du graphe aléatoire $G|X,\theta$ sont alors conditionnellement indépendants et existent avec probabilité Q_{ij}. Comme le graphe aléatoire est une séquence de variables aléatoires de Bernoulli, en générer un échantillon s'effectue à un coût computationnel moindre à partir de méthodes de Monte-Carlo (présentées à la section 1.6). Cette approche accélère donc nettement les calculs numériques.

Avec cette décomposition, les variables aléatoires sous leur forme conjointe $G,\theta|X$ sont équivalentes aux variables aléatoires séparées $\theta|X$ et $G|X,\theta$. Afin de générer un échantillon de la loi *a posteriori*, il suffit alors de générer un échantillon de $G|X,\theta$ et d'appliquer les techniques numériques de la section 1.7 pour obtenir un échantillon de la loi $\mathbb{P}(\theta|X)$.

1.6 Méthodes de Monte-Carlo

Comme illustré dans les sections précédentes, les calculs analytiques sont difficiles pour de nombreux modèles bayésiens. Ces calculs peuvent toutefois être approchés numériquement : en générant un échantillon de la distribution *a posteriori*, il est possible d'estimer l'espérance $\mathbb{E}[f(\theta)|X]$, la variance $\mathbb{V}[f(\theta)|X]$ ou encore un centile d'une fonction f des paramètres *a posteriori*. Des algorithmes pour générer un échantillon d'une distribution quelconque doivent alors être élaborés. À cette fin, les méthodes de type Monte-Carlo sont bien adaptées.

Les méthodes de Monte-Carlo forment une famille de méthodes permettant d'obtenir des résultats numériques à l'aide de nombres aléatoires. Parmi leurs nombreuses applications, ces

algorithmes permettent notamment d'estimer des intégrales de haute dimension, d'optimiser des fonctions de manière robuste et de générer un échantillon d'une distribution quelconque.

Cette section présente deux algorithmes pour générer des échantillons numériquement selon différentes distributions, puis présente un exemple d'utilisation de ces échantillons dans le cadre d'un calcul d'intégrale.

1.6.1 Générer de l'aléatoire

Autrefois, il était commun d'utiliser une table (provenant d'une expérience physique par exemple) pour obtenir des nombres aléatoires. Cependant, dans le cadre du calcul par Monte-Carlo, un grand nombre de variables aléatoires est généralement requis pour obtenir une bonne approximation comme il sera illustré à la section 1.6.4. Ces tables sont donc peu pratiques dans ce type d'application.

Pour régler ce problème, les *générateurs de nombres pseudo aléatoires* (PRNG, de l'anglais *pseudo random number generator*) ont été développés. Ces algorithmes basés sur des applications chaotiques génèrent des séquences ayant une taille quelconque de nombres aléatoires approximativement i.i.d d'une loi uniforme continue dans l'intervalle $[0, 1]$. Le générateur congruentiel linéaire, par exemple, produit une séquence à partir de la relation de récurrence

$$x_{n+1} = (ax_n + c) \mod m. \tag{1.80}$$

Un nombre réel sur l'intervalle $[0, 1]$ est obtenu en divisant les x_n par $m - 1$. Il existe une grande variété de PRNG qui créent de meilleurs échantillons que le générateur présenté, mais ce mémoire n'entre pas dans les détails de ceux-ci. Ces sujets sont explorés dans le livre *Random Number Generation and Monte Carlo Methods* [65]. Pour la suite de ce document, l'accès à un PRNG adéquat est tenu pour acquis.

1.6.2 Méthode de la transformée inverse

Pour échantillonner une loi f quelconque, l'algorithme conceptuellement le plus simple est sans doute la *méthode de la transformée inverse*. Dans cette méthode, la transformation appropriée g est appliquée sur une variable aléatoire $u \sim U(0, 1)$ de sorte que $g(u) \sim f$. Ce faisant, une séquence d'un PRNG est directement traduite à la distribution f. Cette transformation g est la fonction de répartition inverse de f, ce qu'on démontre ici pour une loi f univariée. L'algorithme 1 présente en pseudo-code cette méthode.

> **Démonstration – Une variable aléatoire uniforme transformée par la fonction de répartition inverse suit la loi f.**
>
> Soit $U \sim U(0, 1)$ et la fonction de répartition CDF de la loi f. Si le domaine d'intégration de la CDF est restreint au support de f, alors la CDF est une fonction monotone strictement

```
fonction TransforméeInverse
    U ← Uniforme(0, 1)
    retourner CDF^{-1}(U)
fin fonction
```

Algorithme 1 : Pseudo-code de la méthode de la transformée inverse.

croissante et est donc bijective (sa fonction inverse est bien définie). Puisque les inégalités sont préservées sous l'application d'une fonction monotone croissante,

$$\mathbb{P}\left(\{\mathrm{CDF}^{-1}(U) \leq x\}\right) = \mathbb{P}(\{U \leq \mathrm{CDF}(x)\}).$$

$\mathbb{P}(\{U \leq u\})$ est la définition de la fonction de répartition de U

$$\mathbb{P}(\{U \leq u\}) = \int_0^u \mathrm{d}x = u.$$

Ainsi,

$$\mathbb{P}\left(\{\mathrm{CDF}^{-1}(U) \leq x\}\right) = \mathrm{CDF}(x).$$

La fonction de répartition inverse de la variable aléatoire $\mathrm{CDF}^{-1}(U)$ est donc la fonction de répartition de f, ce qui implique que les deux variables aléatoires sont identiques.

À titre d'exemple, trouvons l'inverse de la fonction de répartition pour générer des nombres aléatoires de la loi exponentielle $f(x) = \lambda e^{-\lambda x}$. La fonction de répartition est

$$\mathrm{CDF}(x) = \int_0^x \lambda e^{-\lambda y}\mathrm{d}y = -e^{-\lambda y}\Big|_0^x = 1 - e^{-\lambda x} \tag{1.81}$$

et la fonction de répartition inverse est

$$u = 1 - e^{-\lambda x} \qquad x = \frac{\ln(1-u)}{\lambda} = \mathrm{CDF}^{-1}(u). \tag{1.82}$$

D'autres exemples sont fournis en annexe : l'annexe A.1 applique cette méthode pour échantillonner une loi géométrique tronquée et l'annexe A.2.1 l'applique pour la loi Gamma tronquée en utilisant une estimation numérique de la fonction de répartition inverse.

1.6.3 Méthode du rejet

Malheureusement, la fonction de répartition inverse de la majorité des distributions n'a pas de forme fermée connue. Bien qu'elle puisse être évaluée numériquement, il est parfois plus simple d'utiliser une méthode alternative comme la *méthode du rejet*.

Dans la méthode du rejet, une réalisation x est obtenue numériquement d'une *loi de proposition* g. En « acceptant » la réalisation x avec probabilité

$$\mathbb{P}(\text{Accepter } x|x) = \frac{f(x)}{Mg(x)}, \tag{1.83}$$

où M est une constante finie telle que $Mg(x) \geq f(x)$ pour tout x, la distribution des x acceptés $\mathbb{P}(x|x \text{ accepté})$ est la loi cible f. L'algorithme 2 décrit en pseudo-code la méthode du rejet.

```
fonction MéthodeRejet
    faire
        S ← g(S)
        U ← Uniforme(0, 1)
    tant que U ≥ f(S) / (M g(S))
    retourner S
fin fonction
```

Algorithme 2 : Pseudo-code de la méthode du rejet. En pratique, il est possible de lancer une erreur après un nombre maximal d'itérations pour éviter une boucle infinie ou trop longue.

Démonstration – La loi des propositions acceptées est la loi f.

Soit la variable aléatoire X continue de densité f à échantillonner, la variable aléatoire continue Y suivant la distribution de proposition g pour laquelle $\text{supp}(g) \quad \text{supp}(f)$, une constante finie M choisie telle que $Mg(y) \geq f(y)$ pour tout y, la variable aléatoire $U \quad U(0,1)$ et l'événement $A = \{U \leq \frac{f(y)}{Mg(y)}\} \cap \{Y = y\}$ d'une proposition acceptée. Les variables aléatoires U et Y sont indépendantes. La fonction de répartition de la variable aléatoire Y conditionnelle à l'événement A est, de la définition (1.7),

$$\begin{aligned}
\mathbb{P}(\{Y \leq x\}|A) &= \frac{\mathbb{P}(\{Y \leq x\} \cap A)}{\mathbb{P}(A)} = \frac{\int_{-\infty}^{x} g(y) \int_{0}^{\frac{f(y)}{Mg(y)}} du dy}{\int_{-\infty}^{\infty} g(y) \int_{0}^{\frac{f(y)}{Mg(y)}} du dy} \\
&= \frac{\int_{-\infty}^{x} g(y) \frac{f(y)}{Mg(y)} dy}{\int_{-\infty}^{\infty} g(y) \frac{f(y)}{Mg(y)} dy} \\
&= \frac{\int_{-\infty}^{x} f(y) dy}{\int_{-\infty}^{\infty} f(y) dy} = \mathbb{P}(\{X \leq x\}).
\end{aligned}$$

La fonction de répartition de la loi des propositions acceptées est la fonction de répartition de la variable aléatoire X. Par conséquent, les variables aléatoires $Y|A$ et X sont identiques et ont la même distribution.

Il est obtenu dans la démonstration que $\mathbb{P}(A) = 1/M$, ce qui indique qu'une proposition est acceptée avec probabilité $1/M$. De cette façon, la variable aléatoire du nombre d'itérations

requis avant d'obtenir une valeur acceptée suit une loi géométrique Geom($1/M$). Ce faisant, l'algorithme prend en moyenne M itérations pour générer une variable aléatoire de f à partir de g.

Il faut ainsi minimiser la constante M pour maximiser l'efficacité de l'algorithme. La constante M idéale est donc

$$M = \sup_{x \in \operatorname{supp} f} \frac{f(x)}{g(x)}. \tag{1.84}$$

La grandeur M peut s'interpréter comme la ressemblance entre g et f. En effet, plus f et g sont semblables, plus le supremum du ratio $f(x)/g(x)$ est petit. Par ailleurs, si $M = 1$, alors $f = g$ et la proposition g est parfaite, de sorte que les x sont acceptés avec probabilité 1 (mais ceci est sans intérêt pratique).

Le principal désavantage de cette méthode est qu'obtenir une loi g convenable est parfois difficile. Notamment, afin que M soit finie, la condition $\lim_{x \to \infty} f(x)/g(x) < \infty$ doit être respectée. Ceci signifie que la queue de g doit être plus épaisse que celle de f si le support de f est infini. De plus, il faut que $\operatorname{supp}(f) \subseteq \operatorname{supp}(g)$. Enfin, si la complexité algorithmique requise pour échantillonner g est grande, cette loi de proposition est possiblement inappropriée, car la méthode du rejet requiert la génération de plusieurs réalisations de g. Un exemple de méthode du rejet est présenté dans l'annexe A.2.2 pour échantillonner une loi Gamma tronquée.

1.6.4 Application : intégration par Monte-Carlo

Une des principales applications du calcul par Monte-Carlo est l'estimation numérique d'intégrales de haute dimension. Par exemple, pour le modèle bayésien de reconstruction, les intégrales sur la loi *a posteriori* permettent de déterminer les caractéristiques des graphes et des paramètres pour un certain jeu de données. On peut, par exemple, désirer calculer l'espérance d'une fonction $g(G, \theta)$

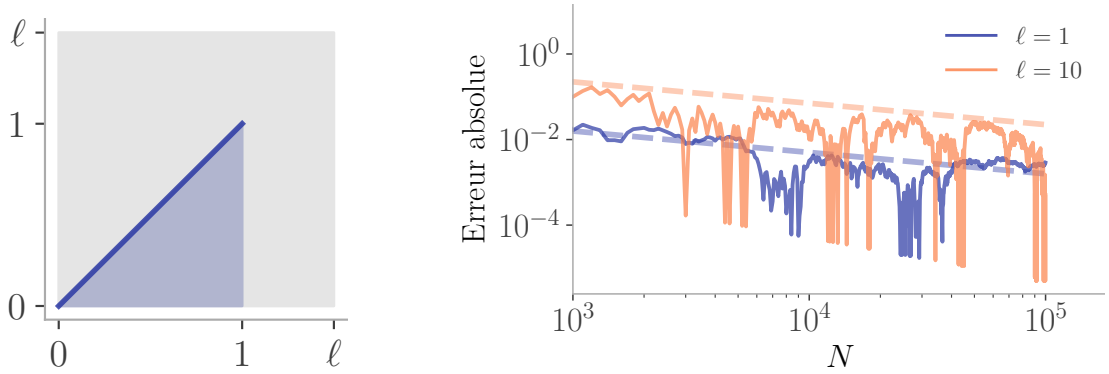
$$\mathbb{E}[g(G, \theta)|X] = \sum_{G} \int_{\mathbb{R}^k} g(G, \theta)\mathbb{P}(G, \theta|X)\mathrm{d}\theta, \tag{1.85}$$

où k dénote le nombre de paramètres dans θ et où la somme sur G énumère tous les graphes possibles. Un exemple de fonction g intéressante dans le contexte de reconstruction est le nombre de liens correctement identifiés dans la loi *a posteriori* (supposant que le graphe ayant généré X est connu).

Avant de s'intéresser à l'estimation de l'équation (1.85), on explore un exemple simple pour illustrer la méthode. Supposons qu'on cherche à estimer l'intégrale suivante

$$I = \int_0^1 x\mathrm{d}x. \tag{1.86}$$

Analytiquement, cette intégrale est trivialement 1/2. Développons une méthode Monte-Carlo pour estimer ce résultat uniquement à partir de l'intégrande et des bornes.



(a) (b)

Figure 1.6 : Calcul Monte-Carlo de l'intégrale (1.86). (a) Aire sous la courbe de l'intégrande (bleu) et carré de longueur qui recouvre l'aire sous la courbe (gris). (b) Erreur relative de l'estimateur $\hat{I}$ en fonction de la taille de l'échantillon pour = 1 et = 10. L'écart-type calculé à l'aide de l'équation (1.92) est tracé en pointillé.

L'intégrale (1.86) correspond à l'aire sous la courbe d'un triangle aux côtés de longueur $(1, 1, \sqrt{2})$. Cette région peut être couverte par un carré de côté ≥ 1 (voir figure 1.6). Puisque l'aire de ce carré est 2^2 fois plus grande que l'aire du triangle, un point uniformément tiré dans le carré a une probabilité $p = 1/(2^2)$ d'être à l'intérieur du triangle. Or, dans une application typique, le résultat de l'intégrale I est inconnu et la probabilité p ne peut donc pas être évaluée.

C'est ici qu'entre en jeu le calcul par Monte-Carlo : le résultat de l'intégrale I peut être déduit d'une estimation de la probabilité p. Cette estimation est e□ectuée en calculant la proportion de points générés uniformément dans le carré qui sont à l'intérieur du triangle. Mathématiquement, avec N points, l'estimateur $\hat{p}$ est

$$\hat{p} := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\Delta}(U_i), \tag{1.87}$$

où U_i sont les points uniformément distribués dans le carré et où $\mathbb{1}_{\Delta}(U_i)$ vaut 1 si le point est dans le triangle et est 0 sinon. L'estimateur de l'intégrale est alors le produit entre l'aire du carré et la probabilité qu'un point soit dans le triangle

$$\hat{I} = {}^2\hat{p}. \tag{1.88}$$

D'un point de vue général, l'aire d'une région R est estimée à partir de l'aire d'une région M qui la recouvre. La probabilité p qu'un point U uniformément distribué dans M soit à l'intérieur de R est

$$p = \mathbb{E}[\mathbb{1}_R(U)] = \frac{A(R)}{A(M)} \tag{1.89}$$

où $A(\cdot)$ dénote l'aire d'une région. En choisissant R comme étant la région entre l'intégrande et 0, l'aire $A(R) = I$ est le résultat de l'intégrale. Grâce à la loi des grands nombres, l'espérance(1.85) est estimée par la moyenne d'échantillon de variables aléatoires U_i i.i.d. uniformément dans la région M

$$\mathbb{E}[\mathbb{1}_R(U)] \quad \hat{p} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_R(U_i). \tag{1.90}$$

En joignant les équations (1.89) et (1.90), on peut estimer l'intégrale avec l'estimateur

$$\hat{I} := \frac{A(M)}{N} \sum_{i=1}^{N} \mathbb{1}_R(U_i). \tag{1.91}$$

On note qu'un estimateur est une variable aléatoire et une estimation est sa réalisation. Comme mentionné dans la section 1.2, la distinction entre ces deux concepts n'est pas effectuée.

Par définition, les variables aléatoires $Y_i := \mathbb{1}_R(U_i)$ sont des variables aléatoire i.i.d. d'une loi Bern$(A(R)/A(M))$. Par conséquent, la somme $Z := \sum_i Y_i$ est une variable aléatoire binomiale [66]. Comme la variance de Z est $Np(1-p)$, l'écart-type SD de l'estimateur $\hat{I}$ est

$$\begin{aligned} \mathrm{SD}\left[\hat{I}\right] &= \sqrt{\mathbb{V}\left[\frac{A(M)}{N} Z\right]} = \sqrt{\left(\frac{A(M)}{N}\right)^2 \mathbb{V}[Z]} \\ &= \sqrt{\left(\frac{A(M)}{N}\right)^2 N \frac{A(R)}{A(M)} \left(1 - \frac{A(R)}{A(M)}\right)} \\ &= \sqrt{\frac{A(R)}{N}(A(M) - A(R))} \quad \frac{1}{\sqrt{N}}. \end{aligned} \tag{1.92}$$

Une démarche alternative pour arriver à ce résultat fait appel au théorème de la limite centrale. En effet, puisque les Y_i ont des variances finies, le théorème de la limite centrale stipule que Z suit une loi normale de moyenne Np et de variance $Np(1-p)$ dans la limite où $N \to \infty$.

La figure 1.6b présente la décroissance de l'erreur relative entre I et son estimateur $\hat{I}$ au cours d'une simulation numérique Monte-Carlo pour l'intégrale (1.86). On remarque que l'erreur d'estimation ne décroît pas de façon monotone. En effet, $\hat{I}$ est une variable aléatoire et c'est sa variance qui est une fonction monotone décroissante de la taille d'échantillon, comme illustré par l'écart-type sur la figure. Dans cet exemple, l'écart-type peut être calculé exactement puisque I est connu, mais ceci n'est typiquement pas le cas. Enfin, la figure 1.6b met en lumière l'importance du choix de la région M : l'erreur est sensible au choix du paramètre contrôlant la taille de M.

Ce qui fait de l'intégration par Monte-Carlo un outil intéressant est sa généralité. En effet, la méthode présentée se généralise directement à des intégrales plus complexes et de plus haute dimension : il suffit d'ajuster les régions R et M et d'utiliser la mesure de volume

appropriée pour l'espace. La convergence en $N^{-1/2}$ tient également dans le cas général tant que les conditions du théorème de la limite centrale sont respectées. Toutefois, la constante de proportionnalité devant l'erreur dépend de l'aire (ou volume) de M, ce qui indique que l'erreur devient exponentiellement plus grande lorsque des dimensions sont ajoutées. En revanche, d'autres approches de type Monte-Carlo comme l'échantillonnage préférentiel permettent de réduire cette variance (voir Ref. [66]).

Avec cette technique d'intégration, l'espérance de l'équation (1.85) s'estime grâce à

$$\mathbb{E}[g(G,\theta)|X] \quad \frac{1}{N}\sum_{i=1}^{N} g(G_i,\theta_i)\mathbb{P}(G_i,\theta_i|X), \tag{1.93}$$

où G_i sont des graphes aléatoires de n noeuds uniformément distribués et où θ_i sont des variables aléatoires distribuées uniformément dans le support de $\mathbb{P}(\theta)$. Il s'agit donc de la moyenne de g pondérée par la probabilité *a posteriori* $\mathbb{P}(G_i,\theta_i|X)$.

1.7 Monte-Carlo par chaînes de Markov

En théorie, l'approche Monte-Carlo conventionnelle présentée est suffisante pour évaluer une espérance de la forme (1.85). En pratique toutefois, plusieurs problèmes majeurs surviennent à l'équation (1.93) :

- l'évidence $\mathbb{P}(X)$ doit être connue pour évaluer $\mathbb{P}(G_i,\theta_i|X)$, ce qui nécessite un autre calcul Monte-Carlo ;
- la grandeur de l'espace des paramètres et des graphes engendre une grande variance sur l'estimateur ;
- la loi *a posteriori* a une petite variance, de sorte qu'un très grand échantillon de la distribution uniforme est requis pour obtenir des contributions $\mathbb{P}(G_i,\theta_i|X)$ non négligeables.

Il est donc préférable d'utiliser une approche différente pour estimer les caractéristiques de la distribution *a posteriori* : les algorithmes de Monte-Carlo par chaînes de Markov (MCMC de l'anglais *Markov Chain Monte Carlo*).

Les algorithmes de MCMC forment une famille de techniques pouvant échantillonner une distribution f quelconque à l'aide d'une chaîne de Markov. Intuitivement, un algorithme de MCMC parcourt le support en favorisant les réalisations qui ont une plus grande masse de probabilité. Il s'agit d'une alternative à la méthode de la transformée inverse et à la méthode du rejet. Son avantage est qu'il permet d'échantillonner aisément des distributions plus complexes que les autres techniques présentées.

Afin d'estimer l'espérance de l'équation (1.85), la moyenne d'un échantillon de la loi *a posteriori* généré par l'algorithme de MCMC est calculée. Mathématiquement, en faisant appel à la loi des grands nombres,

$$\mathbb{E}[g(G,\theta)|X] \quad \frac{1}{N}\sum_{i=1}^{N} g(G_i,\theta_i), \qquad G_i,\theta_i \quad \mathbb{P}(G_i,\theta_i|X). \tag{1.94}$$

L'intérêt de l'estimation par MCMC est qu'elle pallie les problèmes soulevés pour l'estimation par Monte-Carlo conventionnelle :

- l'évidence ne doit plus être évaluée en utilisant l'algorithme de Metropolis-Hastings (expliqué à la sous-section 1.7.3) ;
- la variance n'est plus proportionnelle à la taille de l'espace, mais elle dépend de l'autocorrélation de la chaîne de Markov générée [67] ;
- tous les éléments de l'échantillon contribuent également à l'estimateur, ce qui assure une plus grande stabilité du résultat comparé à une loi uniforme.

De surcroît, générer un échantillon de la distribution est avantageux puisque ceci permet d'obtenir aisément d'autres statistiques comme les centiles et la variance.

Cette section introduit, dans un premier temps, les chaînes de Markov et leurs propriétés afin de comprendre le fonctionnement des algorithmes de MCMC. Dans un deuxième temps, les deux algorithmes de MCMC utilisés dans ce travail sont présentés : l'algorithme de Metropolis-Hastings et l'échantillonneur de Gibbs. Le contenu est inspiré des livres *Markov Chains and Mixing Times (Second Edition)* [68] et *Monte Carlo Statistical Methods* [69].

1.7.1 Chaînes de Markov

Une *chaîne de Markov* est une séquence ordonnée de variables aléatoires (X_t) dont chaque X_{t+1}, nommé *état*, dépend uniquement de l'état précédent X_t. Le support des variables aléatoires X_t est l'*espace des états*. Dans la chaîne de Markov, le passage d'un état $X_t = x_i$ au suivant $X_{t+1} = x_j$ est appelé une *transition*. L'évolution de la distribution des états est dictée par ses *probabilités de transition* (ou *noyau de transition*) $\mathbb{P}(X_{t+1} = x_j|X_t = x_i)$. Si ses probabilités de transition sont constantes dans le temps

$$\mathbb{P}(x_j|x_i) := \mathbb{P}(X_{t+2} = x_j|X_{t+1} = x_i) = \mathbb{P}(X_{t+1} = x_j|X_t = x_i) \qquad t, \tag{1.95}$$

la chaîne de Markov est dite *homogène*. Comme tous les algorithmes de MCMC présentés utilisent des chaînes de Markov homogènes, les probabilités de transitions sont notées $\mathbb{P}(x_j|x_i)$.

La distribution des états à un temps $t+1$, notée π_{t+1} s'obtient de la marginalisation du noyau de transition avec la distribution des états précédente π_t

$$\pi_{t+1}(x_j) = \sum_i \mathbb{P}(x_j|x_i)\pi_t(x_i) \qquad j. \tag{1.96}$$

La chaîne a donc une distribution initiale des états π_0, de manière similaire à un système physique qui possède des conditions initiales (énergie, vitesse, position, etc.).

Si une distribution des états π est invariante sous l'évolution temporelle de la chaîne

$$\pi(x_j) = \sum_i \mathbb{P}(x_j|x_i)\pi(x_i) \qquad j, \tag{1.97}$$

elle est dite *stationnaire*. De plus, s'il existe une distribution des états π telle que l'évolution est symétrique par rapport au temps

$$\mathbb{P}(x_j|x_i)\pi(x_i) = \mathbb{P}(x_i|x_j)\pi(x_j), \tag{1.98}$$

la chaîne de Markov est *réversible*. La loi π est stationnaire par construction

$$\sum_i \mathbb{P}(x_j|x_i)\pi(x_i) = \sum_i \mathbb{P}(x_i|x_j)\pi(x_j) = \pi(x_j) \sum_i \mathbb{P}(x_i|x_j) = \pi(x_j). \tag{1.99}$$

Lorsqu'il existe une séquence finie de transitions de probabilités non nulles permettant le passage de l'état x_i à l'état x_j et de l'état x_j à l'état x_i, les états x_i et x_j *communiquent*. La chaîne de Markov est *irréductible* lorsque toutes les paires d'états communiquent.

Finalement, pour une chaîne de Markov irréductible, $T(x_i)$ dénote l'ensemble des tailles t de séquences pour lesquelles le passage d'un état x_i vers lui-même a une probabilité non nulle (en passant ou non par d'autres états). La *période* de l'état x_i est le plus grand commun diviseur de $T(x_i)$; une chaîne est dite *apériodique* si tous ses états ont une période de 1.

1.7.2 Convergence vers la distribution stationnaire

Lorsque son espace des états est fini, une chaîne de Markov admet une distribution stationnaire unique si elle est irréductible et apériodique. De plus, la distribution des états de la chaîne converge en distribution vers la loi stationnaire selon la variation totale dans la limite où $t \to \infty$ pour toute distribution initiale π_0 [68]. L'existence et l'unicité de la distribution stationnaire peuvent être démontrées pour des chaînes de Markov avec un espace des états infini, mais ceci requiert d'autres concepts et propriétés qui ne seront pas abordés dans ce document (voir Ref. [69] pour plus de détails).

Ces propriétés permettent de comprendre le principe des algorithmes de MCMC : en créant une chaîne de Markov dont la distribution stationnaire est f, la distribution des états tend vers f dans la limite où $t \to \infty$. Toutefois, puisqu'une chaîne de Markov simulée numériquement

ne peut pas évoluer pendant un temps infini, quand a-t-elle atteint sa loi stationnaire dans une simulation ?

En fait, comme notée par Robert et Casella [69], la convergence en distribution n'est généralement pas problématique pour les simulations par MCMC. Admettant que la loi cible f attribue une probabilité significative à l'état initial X_0, la chaîne de Markov se comporte comme si elle était à l'équilibre. Les caractéristiques à surveiller dans une simulation sont plutôt la corrélation entre les états X_t ainsi que la vitesse d'exploration de l'espace.

1.7.3 Algorithme de Metropolis-Hastings

La principale difficulté des algorithmes de MCMC est de construire la chaîne de Markov qui converge vers la distribution stationnaire recherchée. Heureusement, il existe un algorithme simple et très général qui permet de construire cette chaîne aisément : l'algorithme de Metropolis-Hastings (MH) [70]. Dans le MH, une chaîne de Markov est construite de sorte qu'à chaque temps t, un état y est proposé selon une *distribution de proposition* $q(y|X_t)$ conditionnelle à l'état courant X_t. L'état y n'est toutefois pas automatiquement assigné à X_{t+1} : il est accepté selon la *probabilité d'acceptation*

$$\alpha(y, X) = \min\left[1, \frac{f(y)q(X|y)}{f(X)q(y|X)}\right]. \tag{1.100}$$

L'algorithme 3 présente en pseudo-code la méthode. Comme f apparaît sous forme de ratio, il n'est pas nécessaire de calculer sa constante de normalisation. En effet, $f(y)/f(X_t) = g(y)/g(X_t)$ pour une fonction $g \propto f$. Il s'agit d'un avantage significatif en inférence bayésienne puisque la constante de normalisation du modèle est typiquement difficile à calculer.

```
fonction MetropolisHastings(X_0, N)
    pour tout t ∈ {1, 2, ..., N} faire
        Y ← q(Y|X)
        U ← Uniforme(0, 1)
        si U ≤ α(Y, X) alors
            X_t ← Y
        sinon
            X_t ← X_{t−1}
        fin si
    fin pour
    retourner (X_i)_{i=1}^N
fin fonction
```

Algorithme 3 : Pseudo-code de l'algorithme de Metropolis-Hastings.

Selon cet algorithme d'acceptation-rejet, les probabilités de transition de la chaîne de Markov

(X_t) sont

$$\mathbb{P}(x_j|x_i) = \begin{cases} \alpha(x_j, x_i)q(x_j|x_i) & \text{si } i = j, \\ 1 - \sum_{k=i} \alpha(x_k, x_i)q(x_k|x_i) & \text{autrement,} \end{cases} \tag{1.101}$$

où la probabilité que l'état $X_{t+1} = x_j$ soit le même état que $X_t = x_i$ correspond à la somme des probabilités qu'un état proposé y soit refusé.

Grâce à la manière dont elle est construite, la chaîne de Markov de MH est réversible pour la loi f, ce qui assure que la loi f est une distribution stationnaire. Ceci est validé en vérifiant que l'équation (1.98) est respectée. Si $i = j$, alors

$$\mathbb{P}(x_j|x_i)f(x_i) = \min\left(1, \frac{f(x_j)q(x_i|x_j)}{f(x_i)q(x_j|x_i)}\right) q(x_j|x_i)f(x_i) = \min\left(q(x_j|x_i)f(x_i),\ q(x_i|x_j)f(x_j)\right),$$
$$\mathbb{P}(x_i|x_j)f(x_j) = \min\left(1, \frac{f(x_i)q(x_j|x_i)}{f(x_j)q(x_i|x_j)}\right) q(x_i|x_j)f(x_j) = \min\left(q(x_i|x_j)f(x_j),\ q(x_j|x_i)f(x_i)\right),$$

alors que l'équation est trivialement respectée si $i = j$.

Afin que l'algorithme de MCMC fonctionne, la distribution de proposition q doit satisfaire deux conditions. Premièrement, la loi stationnaire est seulement si la chaîne est irréductible, signifiant que la distribution de proposition doit permettre d'explorer l'espace complet. Deuxièmement, la chaîne de Markov doit être apériodique pour qu'une loi stationnaire existe. Toutefois, ceci est rarement problématique parce qu'il existe généralement au moins un état proposé qui a une probabilité non nulle d'être refusé $\mathbb{P}(x_i|x_i) > 0$. Il est également possible de proposer des états identiques pour forcer une probabilité $\mathbb{P}(x_i|x_i)$ non nulle. Si la chaîne est irréductible et $\mathbb{P}(x_i|x_i) > 0$, alors la chaîne est apériodique puisque le temps de la chaîne peut être « décalé » pour atteindre des états qui ne serait autrement qu'accessibles à un certain multiple $k > 1$ de t.

Au moment de concevoir et d'ajuster un algorithme de MH, il est important de s'assurer que celui-ci parcourt adéquatement l'espace. D'un côté, une distribution q qui propose des états trop peu probables risque d'engendrer un taux d'acceptation trop bas. De l'autre, une loi q proposant des états trop probables risque d'explorer trop lentement le support. On note également que les algorithmes de MCMC ont généralement de la difficulté à explorer les différents modes d'une distribution multimodale. Toutefois, des algorithmes ont été développés afin d'amoindrir ce problème [71, 72].

Pour valider le bon comportement et la qualité de l'exploration de la chaîne de Markov simulée, il existe plusieurs diagnostics empiriques comme le facteur de réduction potentielle d'échelle (*potential scale reduction factor*) [73] ou encore la taille effective d'échantillon [67].

1.7.4 Échantillonneur de Gibbs

L'échantillonneur de Gibbs est un cas particulier de l'algorithme de MH qui permet d'échantillonner une loi conjointe à partir de ses lois conditionnelles. Supposons maintenant que f est loi conjointe de paramètres $(\theta_i)_{i=1}^{p}$. Pour échantillonner f selon cet algorithme, un ordre quelconque des indices i est choisi à chaque temps t, puis chaque paramètre est échantillonné séquentiellement dans cet ordre selon la loi conditionnelle $\mathbb{P}(\theta_i|\theta_{-i})$, où θ_{-i} dénote l'ensemble des paramètres excluant θ_i. L'algorithme 4 présente en pseudo-code la méthode.

```
fonction Gibbs(θ^(0), N)
    θ ← θ^(0)
    S ← [ ]
    pour tout t ∈ {1, 2, ..., N} faire
        r ← (1, 2, ..., p) réarrangé aléatoirement
        pour tout i dans r faire
            θ_i ∼ ℙ(θ_i|θ_{-i})
        fin pour
        S[t] ← θ
    fin pour
    retourner S
fin fonction
```

Algorithme 4 : Pseudo-code d'un échantillonneur de Gibbs pour une loi conjointe de p paramètres $(\theta_i)_{i=1}^{p}$. $\theta^{(0)}$ est l'état initial de la chaîne de Markov et N est la taille de l'échantillon désiré.

On montre que les mises à jour $\theta_i \sim \mathbb{P}(\theta_i|\theta_{-i})$ sont en fait des propositions de type MH acceptées avec probabilité 1. La distribution de proposition d'un paramètre θ_i est

$$q_i(\theta'|\theta^t) = \begin{cases} \mathbb{P}(\theta_i'|\theta^t_{-i}) & \text{si } \theta'_{-i} = \theta^t_{-i} \\ 0 & \text{autrement} \end{cases} \tag{1.102}$$

où θ^t dénote les paramètres θ au temps t de la chaîne de Markov et où $\mathbb{P}(\theta_i|\theta^t_{-i})$ est la probabilité de proposer l'état θ_i'. La condition $\theta'_{-i} = \theta^t_{-i}$ est imposée, car seul le paramètre θ_i est proposé par la loi q_i au temps t. La probabilité d'accepter θ' est alors

$$\begin{aligned} \alpha(\theta', \theta^t) &= \frac{f(\theta')}{f(\theta^t)} \frac{q_i(\theta^t|\theta')}{q_i(\theta'|\theta^t)} \\ &= \frac{f(\theta')}{f(\theta^t)} \frac{\mathbb{P}(\theta_i^t|\theta^t_{-i})}{\mathbb{P}(\theta_i'|\theta^t_{-i})} \\ &= \frac{\mathbb{P}(\theta^t_{-i})\mathbb{P}(\theta_i'|\theta^t_{-i})}{\mathbb{P}(\theta^t_{-i})\mathbb{P}(\theta_i^t|\theta^t_{-i})} \frac{\mathbb{P}(\theta_i^t|\theta^t_{-i})}{\mathbb{P}(\theta_i'|\theta^t_{-i})} \\ &= 1 \end{aligned} \tag{1.103}$$

où $f(\theta) = \mathbb{P}(\theta_{-i})\mathbb{P}(\theta_i|\theta_{-i})$. Cette courte démonstration provient du livre *Bayesian data analysis* [64].

En pratique, un ordre fixe d'échantillonnage des paramètres est souvent prédéterminé pour simplifier et accélérer légèrement l'algorithme numérique. Cependant, les garanties statistiques de convergence supposent un ordre uniformément distribué. He et al. [74] montrent que le temps de convergence peut varier d'un facteur polynomial entre les deux approches. Ils illustrent pour différents modèles qu'un ordre particulier de paramètres permet d'obtenir une convergence plus rapide et que la convergence est significativement plus lente pour d'autres ordres. Il semble donc judicieux de choisir un ordre aléatoire pour obtenir une convergence rapide et robuste.

Cet algorithme est au coeur de l'aspect numérique du projet de recherche décrit au chapitre 2. Un exemple détaillé d'application de cet algorithme se trouve à l'annexe 2.9.

Chapitre 2

Reconstruction d'hypergraphes par inférence bayésienne

Hypergraph reconstruction from noisy pairwise observations

Simon Lizotte[1, 2], Jean-Gabriel Young[1, 3, 4], Antoine Allard[1, 2, 4]

[1] Département de physique, de génie physique et d'optique,
Université Laval, Québec (Québec), Canada G1V 0A6
[2] Centre interdisciplinaire en modélisation mathématique,
Université Laval, Québec (Québec), Canada G1V 0A6
[3] Department of Mathematics and Statistics,
University of Vermont, Burlington, VT 05405, USA
[4] Vermont Complex Systems Center,
University of Vermont, Burlington, VT 05405, USA

2.1 Avant-propos

Dans le chapitre 1, les hypergraphes, les principes de l'inférence bayésienne et plusieurs techniques numériques d'échantillonnage ont été présentés. En construisant sur ces concepts, ce chapitre introduit un nouveau modèle de reconstruction d'hypergraphes par inférence bayésienne.

2.2 Résumé

La tâche de reconstruction de réseau vise à estimer la structure d'un système complexe à partir de données de diverses sources telles que des séries temporelles, des mesures « instantanées » prises dans un court laps de temps ou des comptes d'interactions. Des travaux récents ont examiné ce problème dans le contexte de graphes où toutes les relations impliquent précisément deux entités. Dans cet article, nous étudions le problème général de reconstruction d'un réseau dans lequel des interactions d'ordre supérieur existent également. Nous nous concentrons sur un exemple minimal de ce problème, les hypergraphes ayant des interactions entre des paires (liens) et des triplets (triangles) de noeuds mesurées de manière imparfaite et indirecte. Nous développons un algorithme de *Metropolis-Hastings-within-Gibbs* pour ce modèle et l'utilisons pour mettre en lumière les défis uniques qui se présentent avec les modèles d'ordre supérieur. Finalement, nous montrons que pour la plupart des réseaux réels et synthétiques, ce nouveau modèle reconstruit la structure avec une plus grande précision qu'un modèle de graphe avec deux types de liens.

2.3 Abstract

The network reconstruction task aims to estimate a complex system's structure from various data sources such as time series, snapshots, or interaction counts. Recent work has examined this problem in networks whose relationships involve precisely two entities—the pairwise case. Here we investigate the general problem of reconstructing a network in which higher-order interactions are also present. We study a minimal example of this problem, focusing on the case of hypergraphs with interactions between pairs and triplets of vertices, measured imperfectly and indirectly. We derive a Metropolis-Hastings-within-Gibbs algorithm for this model to highlight the unique challenges that come with estimating higher-order models. We show that this approach tends to reconstruct empirical and synthetic networks more accurately than an equivalent graph model without higher-order interactions.

2.4 Introduction

Networks are a convenient model for the intricate structure of complex systems, in which interactions between any pair of the system's constituting elements can be directly interpreted

as edges between the corresponding vertices of a graph. In typical network analyses, these pairwise interactions will initially be unknown as we cannot observe them directly; one must instead define a model of what is and is not an interaction and put this model to the data to identify the relevant network. For instance, we might define a pollinator and a plant species as interacting if a pollinator prefers a particular species over others. This definition will then let us infer a plant-pollinator interaction network by observing how often each pollinator visits each plant and processing the data with an appropriate statistical model [44, 75].

Numerous methods have been proposed to perform this critical step of the network analysis process, commonly called graph reconstruction. They span a broad range of statistical and machine learning techniques and are often tailored to the specific field for which they have been developed [12]. Gene regulatory networks, for instance, have been reconstructed with methods ranging from random forests [13] to methods based on Pearson correlation in temporal windows [14] or in ordinary differential equations [15]. Bayesian frameworks based on genomic features [16] or random-walk-based algorithms [76] have been used to estimate protein-protein interaction networks; while brain networks have been measured with a vast range of methods like cross-frequency phase synchronization [77], Granger causality [78], and matrix-regularized network learning frameworks [79]. More general methods have also been developed to reconstruct diverse datasets [9, 19–22].

While convenient, graphs are fundamentally limited to encoding dyadic connections and higher-order interactions aren't always reducible to a set of pairwise ties [25, 25, 80]. For example, empirical evidence shows that accounting for such higher-order interactions can enhance models of cortical dynamics [24], of biodiversity [26, 81, 82], and of social group formation [83]. If they are to reap the benefits of such representations, network science methods should be able to handle higher-order interactions whenever dyadic relationships are insufficient.

There has been significant recent progress in adapting the network science methods to higher-order representations [23], but the higher-order reconstruction problem has no fully satisfactory solution to date. For instance, direct generalizations of dyadic algorithms are impractical due to the extraordinary amount of data they require to function [23]—a network of n vertices can support up to 2^n hyperedges so naively measuring every edge becomes rapidly infeasible with growing n. Recent work has addressed this issue partially by using pairwise data to make inferences about possible higher-order structures [29] or filters on incomplete hyperedge data [31]. However, no method to date can simultaneously handle reconstruction and noise in the pairwise measurements.

This paper introduces a Bayesian framework to infer higher-order structural interactions from imperfect pairwise measurements. We study a minimal example of this problem, focusing on the case of hypergraphs with interactions between pairs and triplets of vertices, measured

imperfectly and indirectly. Instead of providing a point estimate, this framework offers a distribution of the possible hypergraphs compatible with all the available observations. The range of structures provided by this distribution allows us to compute error bars for various network measurements and the outcomes of network processes. We also present a network modeling approach that encodes the projection of hyperedges as different types of pairwise interactions to analyze the importance of the correlation induced by these higher-order interactions. Finally, we compare the reconstruction accuracy of these two frameworks on synthetic observations generated from various synthetic and empirical hypergraphs.

2.5 Methods

Let us assume that we possess some measurements $X = [x_{ij}]_{i,j=1,\dots,n}$ of the pairwise interactions of the units of a complex system composed of n elements. In general reconstruction problems, these observations could take on many forms, such as time series correlation of brain regions [84] or the direct observation of the presence (or absence) of edges in a networked system [9], to name only two examples. To keep our presentation of the methods concrete, we will focus on the case where x_{ij} is an integer number of observed interactions for vertices i and j. Our objective is to infer the interactions in a hidden latent structure S under the assumption that these interactions shape the observed behavior of the system (i.e., the measurements). This latent structure could be any type of structural representation such as graphs, simplicial complexes, or hypergraphs.

We expect the observation data to be noisy, meaning that remeasuring the system could lead to different values X. We also expect that similar (different) interactions in S could lead to very different (similar) measurements. For instance, two pairwise observations x_{ij} and x_{rs} could be identical even if the pair (i, j) interacts in S while (r, s) does not. To account for these fluctuations, we develop a Bayesian inference framework, a fully probabilistic approach producing a probability distribution over the different structures S compatible with the data X.

2.5.1 Data model

We first specify the likelihood $P(X|S, \mu)$, which expresses how the observations X are related to the latent structure S and any additional parameters of the observation processes μ. We assume that the structure S encodes three types of symmetrical interactions: each pair (i, j) can interact weakly ($_{ij} = 1$), interact strongly ($_{ij} = 2$) or not interact ($_{ij} = 0$). For instance, measurements X of a social network could be the number of conversations recorded between acquaintances ($_{ij} = 1$), friends ($_{ij} = 2$) or strangers ($_{ij} = 0$).

For a fairly broad range of measurement processes, it will often be reasonable to model the observed number of interactions x_{ij} between vertices i and j with conditionally independent

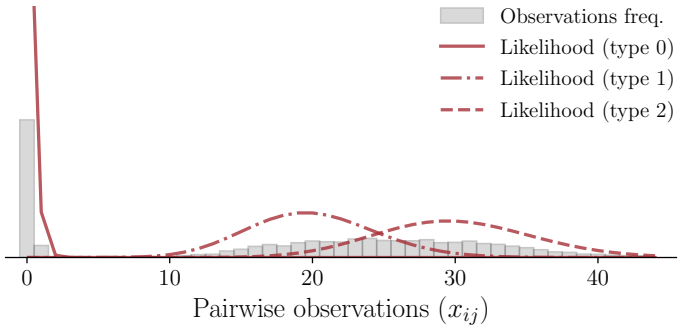


Figure 2.1: Illustration of a typical distribution of pairwise interactions X produced by the data model. The frequencies of the pairwise interactions are shown in gray. The contribution of each type of interaction to the likelihood is shown in red.

Poisson random variables. Hence, the observation x_{ij} is only determined by its associated type of interaction $_{ij}$ and average μ_{ij}, leading to the likelihood

$$P(X|S,\mu) = \prod_{i<j} \frac{\mu_{ij}^{x_{ij}}}{x_{ij}!} e^{-\mu_{ij}}, \tag{2.1}$$

where $\mu = (\mu_0, \mu_1, \mu_2)$. Figure 2.1 illustrates the distribution of pairwise observations modeled by Eq. (2.1). This model will only be appropriate if the errors on two distinct measurements x_{ij} and x_{rs} are not correlated, and every x_{ij} is the outcome of numerous independent observations of an ongoing measurement process with constant success rate. We make these assumptions to provide a simple illustration of our inference framework, but we stress that it is general enough to account for more general and diverse types of data, distributions, and structures.

2.5.2 Structural models

The next step is to specify the latent structural model $P(S|\varphi)$, which is a prior probability on each interaction $_{ij}$ conditioned on some additional parameters collectively denoted by φ. This distribution encodes our hypothesis on the structure of interactions of the system before we make any measurements. For instance, we might expect person i to be more likely to develop a friendship with person j than with person k because i and j live in the same neighborhood.

To highlight the role of latent higher-order interactions in the reconstruction procedure (or lack thereof), we consider two models for the structure S: a hypergraph model ($S = H$) and a categorical-edge model with a graph structure ($S = G$).

Hypergraph model

We define the hypergraph structure $H = (V, E, T)$ as a set of vertices V with 2-edges E and 3-edges T. We limit the size of the hyperedges to 3 for the sake of simplicity, although larger hyperedges could easily be considered by adapting the data model in Eq. (2.1) accordingly. We opt for a simple hypergraph model in which the existence of each hyperedge is conditionally independent from the others. Denoting as p (q) the probability of existence of 3-edges (2-edges), the probability of H is

$$P(H|\varphi_H) = q^{h_1}(1-q)^{\binom{n}{2}-h_1}p^{h_2}(1-p)^{\binom{n}{3}-h_2}, \tag{2.2}$$

where $\varphi_H = \{p, q\}$ are the parameters, $h_1 = |E|$ is the number of 2-edges and $h_2 = |T|$ is the number of 3-edges.

We connect this structure to the data model by assigning a type $_{ij}$ to each pair of vertices as

$$_{ij} = \begin{cases} 2 & \text{if } (i,j) \quad \Delta, \\ 1 & \text{if } (i,j) \quad E \text{ and if } (i,j) \quad \Delta, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

where Δ is the set of pairs covered by a 3-edge

$$\Delta = \{(i,j) \mid \quad k \text{ s.t. } (i,j,k) \quad T\}. \tag{2.4}$$

To make further progress, we must make a few arbitrary choices since the full model—the joint distribution of the data and latent structure—can be re-parametrized in ways that do not a□ect the distribution over labels and, therefore, over data. These symmetries will cause identifiability problems when we use the model to make inferences about latent hypergraphs, so we address them immediately.

First, since the mapping from hypergraph to labels is lossy, the presence of some edges can be *hidden* by others. For example, if vertices i and j are connected by both a 2-edge and a 3-edge, then the interaction will be considered of type $_{ij} = 2$, as if the 2-edge did not exist—removing them does not a□ect the interaction type and consequently does not change the value of the likelihood given at Eq. (2.1). 3-edges can also hide other 3-edges, as depicted in Fig. 2.2. Hence, we must bear in mind that we will only be able to make inferences about "visible" edges.

Second, the full model is susceptible to label-switching and thus needs additional adjustments. Indeed, while a non-interacting pair ($_{ij} = 0$) and a pair of vertices connected by a 2-edge ($_{ij} = 1$) are associated with di□erent distributions of observations because they have distinct means μ_0 and μ_1, it is possible to change the structure H and the parameters μ in a way

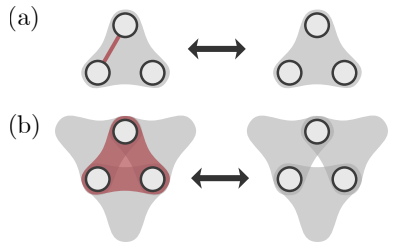


Figure 2.2: Examples of structural configurations with hidden edges. The presence or absence of the (a) 2-edge and (b) 3-edge shown in red does not alter the type of interaction $_{ij}$ of the vertices, which is the same for all configurations. Hence, the likelihood in Eq. (2.1) has the same value, and we say that these red hyperedges are *hidden* by the other 3-edges.

that will not a□ect the overall likelihood of a dataset X. This can be done by replacing every non-interacting pair of H by a 2-edge and vice versa while also swapping the value of μ_0 and μ_1. We address this label-switching symmetry it in the standard way by imposing that $\mu_0 < \mu_1$ or, equivalently, by thinking of non-interacting pairs as associated with a smaller expected number of interactions than interacting pairs.

The label $_{ij} = 2$ can also technically be exchanged with the labels $_{ij} = 0$ and $_{ij} = 1$, but because they are inherited from a latent hypergraph that correlates multiple pairs of vertices, the problem will only manifest itself in very specific situations. Namely, every 2-edge has to belong to at least one triangle formed by two other 2-edges or projected 3-edges (this worst-case hypergraph is described in section 2.6.3). Since a vanishing fraction of hypergraphs exhibit this specific configuration, imposing $\mu_1 < \mu_2$ is unnecessary to disambiguate most configurations. That said, in practice, we found it useful to impose $\mu_0 < \mu_2$. Type 1 and type 2 interactions are typically sparse, which means that type 0 interactions are dense. Non-interacting pairs could therefore seem to form many triangles and could be interpreted as the projection of 3-edges. Imposing $\mu_0 < \mu_2$ avoids any confusion.

Categorical-edge model

Our second model involves graphs with categorical edges $G = (V, E_1, E_2)$ defined as a set of vertices V, of *weak* edges E_1, and of *strong* edges E_2. The types of interaction are then

$$_{ij} = \begin{cases} 2 & \text{if } (i,j) \quad E_2, \\ 1 & \text{if } (i,j) \quad E_1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

Much like in the hypergraph case, we adopt an agnostic model and assume *a priori* that the categorical edges are placed randomly according to a simple two-step generative process: strong edges are created independently with probability q_2 and weak edges are created independently in the remaining unconnected pairs with probability q_1

$$P(G|\varphi_G) = q_1^{m_1}(1-q_1)^{\binom{n}{2}-m_1-m_2} q_2^{m_2}(1-q_2)^{\binom{n}{2}-m_2}, \tag{2.6}$$

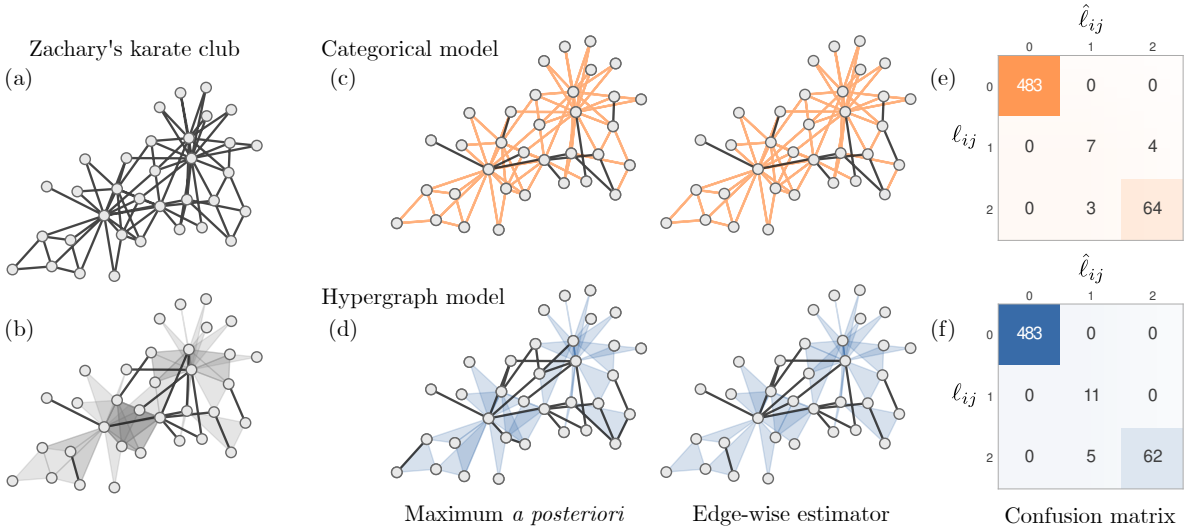


Figure 2.3: Inference process on a small dataset. (a) Original network of Zachary's karate club [85]. (b) Hypergraph representation of the Zachary's karate club, see main text. (c) Illustration of the structure corresponding to the estimators $\hat{S}_{\mathrm{MAP}}$ and $\hat{S}_{\mathrm{EW}}$ for the categorical-edge model. Strong edges are shown in orange. (d) Same as (c) but using the hypergraph model. (e) Confusion matrix built using the $\hat{S}_{\mathrm{MM}}$ estimators for the interaction types. (f) Same as (e) but using the hypergraph model. The inference was done on synthetic observations generated using $\mu = (0.01, 20, 30)$. The Maximum *a posteriori* (MAP) structure maximizes the posterior distributions [Eqs. (2.8) and (2.9)], while the edge-wise structure contains the edges and hyperedges that exist in at least half of the samples of the posterior distributions. The estimator for the type of interaction, noted $\hat{\ell}_{ij}$, is used to build the confusion matrix. It corresponds to the most likely type of interaction for vertices i and j.

where $\varphi_G = \{q_1, q_2\}$, $m_1 = |E_1|$ and $m_2 = |E_2|$ are the number of weak edges and strong edges respectively.

There are no hidden edges in this model but the label switching problem is now three-fold: $\ell_{ij} = 0$ can be swapped with $\ell_{ij} = 1$, but also $\ell_{ij} = 0$ with $\ell_{ij} = 2$ and $\ell_{ij} = 1$ with $\ell_{ij} = 2$. As for the hypergraph model, we address this issue by imposing $\mu_0 < \mu_1 < \mu_2$. Hence, we suppose that non-interaction pairs are less frequently measured than interactions and that weak interactions are less frequently measured than the strong ones.

2.5.3 Posterior distributions

Combining the quantities defined above, the Bayes formula yields the posterior distribution $P(S, \mu, \varphi|X)$ of each structural model

$$P(S, \mu, \varphi|X) = \frac{P(X|S, \mu)P(S|\varphi)P(\mu, \varphi)}{P(X)}, \tag{2.7}$$

where $P(\mu, \varphi)$ is a conjugate prior distribution (see Appendix 2.8 for details) and $P(X)$ is a normalization factor that needs not be specified.

Table 2.1: Properties of the synthetic and empirical hypergraph datasets. The table shows the number of vertices (n), the number of type-1 and type-2 interactions, the relative reconstruction error ( ) for both the categorical-edges graph model and the hypergraph model, as well as the fraction of 2-edges that are hidden under a 3-edge (E_Δ). The relative reconstruction error ( ) shown here is the median of the relative reconstruction error for 10 observation matrices generated with $\mu = (0.01, 40, 50)$.

Hypergraph	n	interaction				E_Δ
		type 1	type 2	Categor.	Hyper.	
Zachary's karate club [85]	34	11	67	0.13	**0.11**	0
Crimes [86]	202	57	209	0.11	**0.05**	0
Sexual contacts [87]	159	47	108	0.11	**0.05**	0
Plant-pollinator [88]	57	51	128	0.11	0.11	0.80
Languages [89]	150	30	242	0.07	**0.03**	0
Hypergraph SBM [58]	100	60	76	0.44	**0.03**	0.20
Triangle-edge CM [59]	100	107	89	0.52	**0.09**	0.32
β-model [60]	100	61	56	0.51	**0.33**	0.70
Best-case	100	92	93	0.38	**0.01**	0
Worst-case	100	100	100	**0.36**	0.50	1

Combining Eqs. (2.2) and (2.6) with (2.7) yields the following posterior distributions

$$P(H, \mu, \varphi_H | X) = \frac{P(\mu, \varphi)}{P(X)} q^{h_1} (1-q)^{\binom{n}{2} - h_1} p^{h_2} (1-p)^{\binom{n}{3} - h_2} \prod_{i<j} \frac{(\mu_{ij})^{x_{ij}}}{x_{ij}!} e^{-\mu_{ij}} \tag{2.8}$$

and

$$P(G, \mu, \varphi_G | X) = \frac{P(\mu, \varphi)}{P(X)} q_1^{m_1} (1-q_1)^{\binom{n}{2} - m_1 - m_2} q_2^{m_2} (1-q_2)^{\binom{n}{2} - m_2} \prod_{i<j} \frac{(\mu_{ij})^{x_{ij}}}{x_{ij}!} e^{-\mu_{ij}}, \tag{2.9}$$

which both weight every structure-parameters tuple (S, μ, φ) according to their compatibility with the observations X and their prior probabilities.

Equations (2.8)–(2.9) are not closed forms of known distributions, with the main complication being due to the presence of edge labels $_{ij}$ in the likelihood. Hence, any meaningful use of these posterior distributions will require the generation of samples from it, which in turn will be used to estimate statistics such as percentiles, the average and the variance of various functions $f(S, \mu, \varphi)$. To this end, we have derived a Metropolis-within-Gibbs algorithm whose details are discussed in Appendix 2.9. A C++/Python implementation is available at https://github.com/DynamicaLab/hypergraph-bayesian-reconstruction. The algorithm returns a series of tuples $\{(S_t, \mu_t, \varphi_t)\}_{t=1,\dots N}$ sampled in proportion to Eq. (2.7), for either structural models.

2.6 Results

2.6.1 Case study: Zachary's Karate Club

We first illustrate the framework with a simple case study based on Zachary's Karate Club [85]. Our goal will be to recover the latent structure of this system, encoded as a hypergraph H, given synthetic data X generated with the likelihood of Eq. (2.1) and $\mu = (0.01, 20, 30)$. This μ makes it fairly easy to discern non-interacting pairs but leads to some overlap between the two other types of interactions, which will allow us to highlight the influence of higher-order interactions on the accuracy on the inference (see Fig. 2.1 which illustrates the distribution of pairwise measurement for this choice of parameters). The structure of the original Karate Club only contains dyadic observations which makes for an uninteresting test of our method, so we add the 3-edges that are found by a separate hypergraph inference technique [29]. (We break down any hyperedge larger than 3 vertices into multiple 3-edges.) We show the original graph and associated hypergraph in Figs. 2.3a and 2.3b—we use the latter throughout our case study.

With this hypergraph structure fixed, we generate a synthetic dataset X and approximate its associated posterior distribution using samples generated with the Markov chain Monte Carlo (MCMC) algorithms described in Appendix 2.9. From these samples, we derive two estimators of the structure: the maximum *a posteriori* (MAP) estimator

$$\hat{S}_{\mathrm{MAP}} = \operatorname*{argmax}_{S} P(S|X), \tag{2.10}$$

corresponding to the latent structure that maximizes the posterior distribution, and the edge-wise estimator $\hat{S}_{\mathrm{EW}}$ that only contains the weak/strong edges or 2-edges/3-edges with a marginal posterior probability above 0.5, e.g., for the hypergraph model

$$\hat{S}_{\mathrm{EW}} = \{e|\ e \quad E \quad T, P(e|X) > 0.5\}, \tag{2.11}$$

where $P(e|X)$ is the marginal probability that edge e is present. We complement these structural estimators with an estimator of the type of each pairwise interaction, the maximum marginal estimator

$$\hat{S}_{\mathrm{MM}} = \{\hat{}_{ij} \mid i, j \quad V\}, \tag{2.12a}$$

where

$$\hat{}_{ij} = \operatorname*{argmax}_{ij \quad \{0,1,2\}} P(\ _{ij}|X) \tag{2.12b}$$

is the most likely type of interaction type for vertices i and j (ties are broken by choosing a type at random).

Figures 2.3c and 2.3d show $\hat{S}_{\mathrm{MAP}}$ and $\hat{S}_{\mathrm{EW}}$ for both models fitted to the same realization of the data X. In both cases, we see that our inference framework reconstructs the original

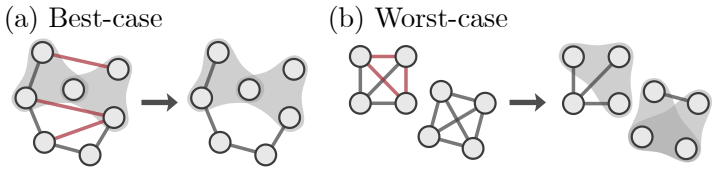


Figure 2.4: Illustration of the generation of the best-case and worst-case hypergraphs. (a) The best-case hypergraphs are obtained by first generating a random hypergraph using Eq. (2.2) and then removing any 2-edge that creates triangle when projecting the hypergraph onto the pairwise interactions. (b) The worst-case hypergraphs are generated from a graph with cliques of 2-edges and in which each triangle can be promoted to a 3-edge with a given probability.

structure quite accurately, though both estimators miss a few 3-edges. While some of them are genuine errors, quite a few missing 3-edges are simply hidden and thus unrecoverable (as defined in section 2.5.2).

Thankfully, these missing hidden 3-edges have little impact on the accuracy of our framework when it comes to predicting the interaction label $_{ij}$. Figures 2.3e and 2.3f show this with confusion matrices, a generalization of statistical errors (type I and type II errors) for multiple classes. The element c_{rs} of this matrix denotes the number of times a pairwise interaction of type $_{ij} = r$ has been predicted as $\hat{}_{ij} = s$ by the maximum marginal estimator $\hat{S}_{\text{MM}}$. Hence, a perfect reconstruction corresponds to a diagonal matrix. The major difference between both confusion matrices is that the categorical-edges model uses weak edges and strong edges somewhat interchangeably, which results in reconstruction errors that go both ways. In contrast, the hypergraph model has no false positive 3-edges. This is due to the restrictive nature of 3-edges: each type-2 pairwise interaction must be associated with at least two other type-2 pairwise interactions (as long as the 3-edge is not hidden). As a result, our framework will err on a more conservative side when assigning larger hyperedges: The framework will assign $_{ij} = 1$ unless there is sufficient evidence in the neighborhood of vertices i and j that supports a 3-edge. This additional neighborhood information is what allows the hypergraph model to have a smaller sum of off-diagonal elements in the confusion matrix, meaning that it more accurately retrieves the interaction types.

2.6.2 Expanded dataset

Next, we study the performance of our inference framework on a broader collection of synthetic and empirical hypergraphs. For the empirical datasets, we select a network of crimes [86], a network of sexual contacts [87], a plant–pollinator network [88] and a network of languages [89]. The original datasets are all bipartite, so we again adapt them to our purpose by interpreting one of the two vertex types as hyperedges: individuals are vertices and crimes are hyperedges, sex workers are vertices and hyperedges are their clients, pollinators are the vertices and the plants they pollinate are hyperedges, vertices are countries and hyperedges are languages spoken. We drop hyperedges with more than five vertices to keep a sufficient

number of 2-edges in the hypergraph, and we also remove any isolated vertex. We also re-use the hypergraph derived from Zachary's Karate Club above.

We complement these empirical datasets with hypergraphs generated using the three computer models, namely (i) the superimposed stochastic block model [58] (two unequal communities of 30 and 70 vertices with connection probabilities of $q_{11} = 0.05$, $q_{12} = q_{21} = 0.001$ and $q_{22} = 0.02$ for 2-edges, and of $p_1 = 0.005$ and $p_2 = 0.0001$ for 3-edges inside communities and $p_{\text{out}} = 0.00001$ outside communities), (ii) a triangle-edge configuration model of 100 vertices [59] (with degrees drawn from independent geometric distributions of means 2 and 3 for 2-edges and 3-edges, respectively), and (iii) the β-model for layered hypergraphs [60] (with vertex propensities of 2-edges and 3-edges drawn from normal distributions of averages -4.5 and -5 and of standard deviations 2.5 and 2, respectively).

As before, we generate a series of synthetic observations with the likelihood in Eq. (2.1) and $\mu = (0.01, 40, 50)$, and then sample the posterior distribution to compute the confusion matrices of both models. We summarize our results using the fraction of misclassified type-1 and type-2 interactions, a quantity we call the *relative reconstruction error*

$$= \frac{c_{10} + c_{12} + c_{20} + c_{21}}{c_{10} + c_{11} + c_{12} + c_{20} + c_{21} + c_{22}}, \tag{2.13}$$

where c_{rs} are the elements of the confusion matrix. The results are reported in Table 2.1 where we see that the hypergraph model performs at least as well as the categorical-edge model. The following section explores the factors influencing the performance of the hypergraph model.

2.6.3 When are the hyperedges most relevant

To gain better insights on the factors influencing the performance of the hypergraph model, we consider two extreme cases: a "best-case hypergraph' and a "worst-case hypergraph".

In the best-case hypergraphs, groups of 3 vertices can only be connected by a 3-edge. This means that vertices (i, j, k) can form a triangle in projected pairwise interactions only if $_{ij} = _{ik} = _{jk} = 2$. As a result, there is no ambiguity on whether or not triangles are a mix of 2-edges and projected 3-edges, and 3-edges can be distinguished from triangles of non-interacting pairs since they have greater pairwise measurements. This e□ectively makes the neighborhood of any pair of vertices very informative on its type of interaction. We generate such hypergraphs by removing the 2-edges that do not respect the imposed constraint from a hypergraph generated with the prior (2.2) (see Fig. 2.4).

The worst-case hypergraphs only contain 2-edges if they form a triangle in the projection. In other words, $_{ij} = 1$ is only possible if there exists another vertex k such that $_{ik}\ _{jk} > 0$. As a result, there is no longer a di□erence in the observations between a 3-edge and a triangle comprised of a mixture of 2-edges and projected 3-edges; the neighborhood of a pairwise

observation is uninformative. To produce these worst-case hypergraphs, we generate graphs with cliques of 2-edges where each triangle is promoted randomly to a 3-edge (see Fig. 2.4).

To estimate how much a given hypergraph resembles the best-case or the worst-case, we compute the proportion of 2-edges inside projected triangles

$$E_\Delta = \frac{1}{h_1} \sum_{(i,j) \in E} \left(1 - \sum_{k \in V} \mathbb{1}[(i,k),(j,k) \in \Delta \in E]\right). \tag{2.14}$$

The closer E_Δ is to 0, the closer the hypergraph is to a best-case hypergraph, and the closer the E_Δ is to 1, the closer the hypergraph is to a worst-case hypergraph.

Revisiting Table 2.1, we see that E_Δ is related to the error and that errors for each hypergraph range between the best-case and the worst-case. However, the proportions ρ_k of pairs predicted as type k, defined as

$$\rho_k = \frac{c_{0k} + c_{1k} + c_{2k}}{\binom{n}{2}}, \tag{2.15}$$

also play a role in : when a type of interaction is being observed at a similar rate to another, models will most likely favor the type with the largest proportion as it leads to a better fit.

Table 2.1 also shows that empirical hypergraphs are generally closer to a best-case hypergraph than to a worst-case. This is due to the sparsity of interactions of empirical complex systems: we expect that most 2-edges are not part of projected triangles. For that reason, the hypergraph model works better than the categorical-edges graph model for the majority of systems. And when the hypergraph model errs, both models tend to err as confirmed by the last two lines of Table 2.1.

2.6.4 Impact of data means

To complete our analysis, we study the impact of the parameters μ on the reconstruction by varying μ_1 while keeping $\mu_0 = 0.05$ and $\mu_2 = 50$ fixed, for the two families of extreme hypergraphs described above (with $n = 100$ vertices). Doing so allows us to identify the regimes in which the hypergraph model displays a better performance. In addition to the relative reconstruction error , we also consider two additional summary statistics: the entropy S of the label distribution, and the sums of residuals R_k.

We define the entropy of the label distribution as

$$S = -\sum_{k=0}^{2} \rho_k \log_3 \rho_k. \tag{2.16}$$

This statistic measures the e ective number of interactions predicted by the models: it is 0 if only one type of interaction exists and it is 1 if $\rho_0 = \rho_1 = \rho_2 = \frac{1}{3}$. Because the empirical datasets we consider are sparse, most pairs of vertices do not interact, meaning that S is

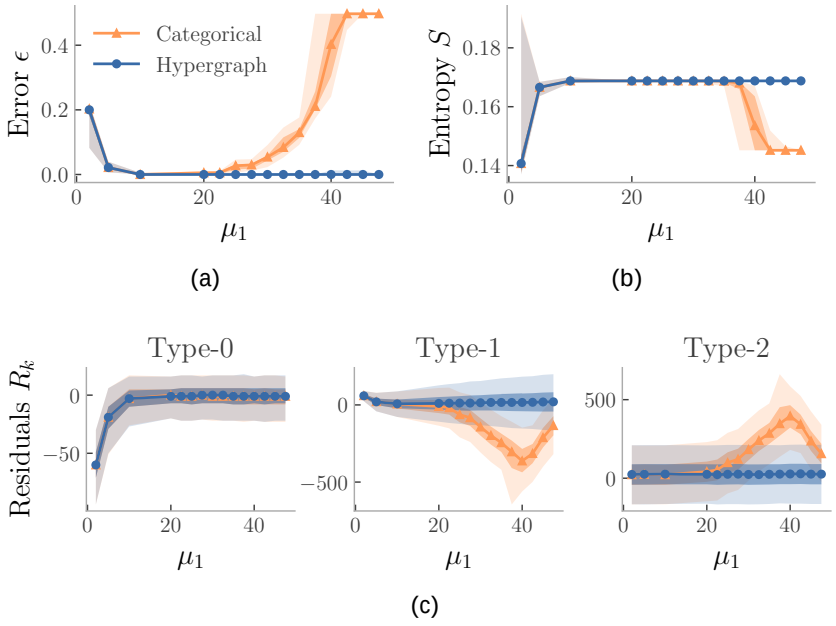


Figure 2.5: Impact of the measurement rate (μ_1) of type-1 interactions on the reconstruction of a best-case hypergraph. (a) Relative reconstruction error . (b) Entropy S. (c) Sums of residuals R_k. The observations were generated with $\mu_0 = 0.01$, $\mu_2 = 50$ and various μ_1 using the hypergraph model (blue) and the categorical-edges graph model (orange). The hypergraph model displays (a) a smaller reconstruction error (b) a larger entropy and (c) lower residuals than the categorical-edges graph model, which indicates a better reconstruction. Symbols represent the median, light colored shadings are percentiles 2.5 and 97.5 and dark colored shadings are percentiles 25 and 75 of the metrics for 200 synthetic observations. Residuals were evaluated using 200 predictive observation matrices and the best-case hypergraph was generated using $p = 0.00017$ and $q = 0.019$.

small. Nevertheless, comparing entropy values allows us to detect when a model completely ignores a type of interaction.

The sums of residuals R_k are defined as

$$R_k = \sum_{i<j} (x_{ij} - \tilde{x}_{ij})\delta_{k,\,ij}, \tag{2.17}$$

where $\tilde{X} = [\tilde{x}_{ij}]_{i,j=1,\dots,n}$ is an observation matrix generated synthetically from the posterior-predictive distribution [22, 90]. For each sample point $\tilde{S}, \tilde{\mu} \sim P(S, \mu|X)$, we generate predictive matrices $\tilde{X}$ from the likelihood (2.1). This is known as a form of *posterior–predictive check*, and it quantifies the goodness of fit of a model by checking that the fitted model can adequately reproduce the original data. The statistics R_k will reveal biases in the fitted model, with $R_k \approx 0$ only when the predicted pairwise observations $\tilde{x}_{ij}$ are on average equal to the pairwise observations x_{ij} for the interactions of type k.

Figures 2.5 and 2.6 show that the relative reconstruction error generally increases as μ_1

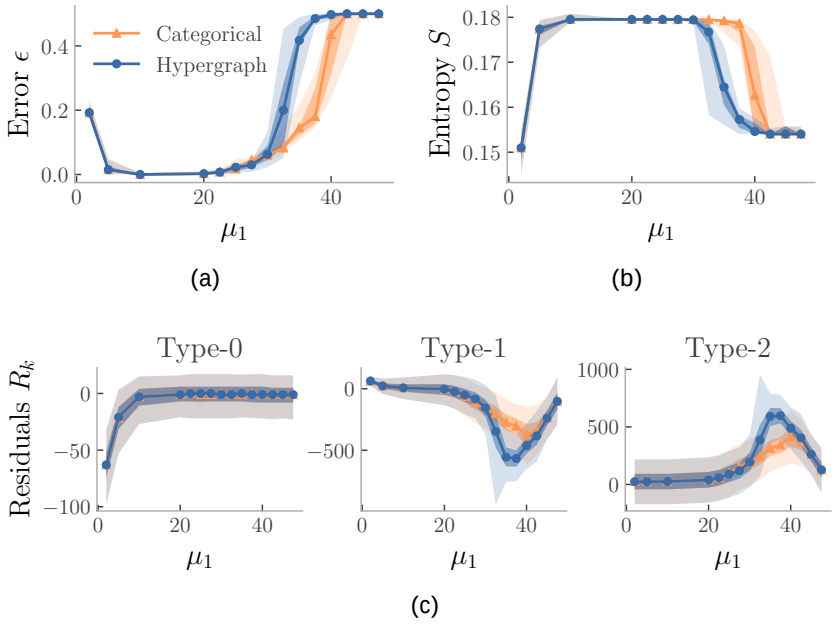


Figure 2.6: Impact of the measurement rate (μ_1) of type-1 interactions on the reconstruction of a worst-case hypergraph (see Fig. 2.5 for details). While the categorical-edges graph has a similar performance to the best-case hypergraph (Fig. 2.5), the hypergraph model cannot distinguish 3-edges from 2-edges with triangles, which results in a worse reconstruction. This is seen with (a) a larger reconstruction error (b) a smaller entropy and (c) larger residuals. The worst-case hypergraph was constructed from 20 5-cliques in which triangles were promoted to 3-edges with probability 0.19.

approaches μ_0 or μ_2. This behavior is expected because there is a greater overlap between the corresponding Poisson distributions in the observations X. When this overlap is large, interaction types are represented similarly in the observations X, which makes them di□cult to infer. Figures 2.5 and 2.6 also show that the entropy generally decreases and stabilizes to a lower plateau as μ_1 approaches μ_2. This is due to a similar phenomenon: with the increasing overlap, models favor one type of interaction over the other to the point where one type of interaction disappears. Once the interaction types have "merged", the entropy remains constant.

For the best-case hypergraph, we clearly see in Fig. 2.5 that the hypergraph model overall outperforms the categorical-edges graph model. Figure 2.5a shows that the hypergraph model makes very little reconstruction errors for all sets of parameters. This translates to a higher entropy, as seen in Fig. 2.5b, and to a smaller predictive bias in Fig. 2.5c. We conclude that the worse performance observed for the categorical-edges graph model is explained by weak and strong edges ending up being interchangeable because of their pairwise nature. Without the information from the neighborhood that 3-edges imply, the interaction type of a pair $_{ij}$ must be deduced from its observation x_{ij} alone.

For the worst-case hypergraph, Fig. 2.6 illustrates that the categorical-edges graph model slightly outperforms the hypergraph model. We believe this is due to the prior distribution of the 3-edge probability p: because there are $\binom{n}{3}$ possible 3-edges compared to $\binom{n}{2}$ possible 2-edges, there is a much larger number of 3-edges than strong edges for the same probability. In this worst-case setting, 3-edges are almost indistinguishable from 2-edges since triangles are mixture of 2-edges and projected 3-edges. Thus, there is no improvement brought by the hypergraph model, which suggest that this hypergraph representation is not appropriate.

2.7 Conclusion

Mounting evidence collected in recent years support that the behavior of many complex systems require taking into account high-order interactions. However, many of the tools of this rapidly expanding field have yet to find practical applications still as measurements of higher-order systems remains challenging to this day.

We presented a minimal Bayesian inference framework that makes progress in this direction, by reconstructing hypergraphs from noisy observations of their pairwise projection. Using synthetic and empirical datasets, we illustrated the impact that taking into account high-order interactions has on the accuracy of the reconstruction. Notably, we identified the regimes where high-order interactions yield fewer reconstruction errors, due to the fact that hyperedges require the use of local information contained in the neighborhood of vertices.

Although the inference framework introduced here is fairly general, we illustrated it using simple data and hypergraph models to avoid obfuscating its presentation unnecessarily. Thus, future work should be done to apply our framework to hypergraphs with hyperedges larger than 3-edges, and to non-Poissonian data models. Doing so will require to treat carefully the way higher-order interactions are assumed to be encoded in the pairwise observation data; as we have shown, hidden hyperedges can hinder high quality reconstruction. A possible solution worth investigating involves the use of simplicial complexes, a more restricted higher-order structure in which a hyperedge of size k implies every hyperedge of size $k - 1$. Yet, how to connect pairwise interactions to such higher-order interactions remains an open question and is a testament to the bright future Bayesian inference of higher-order interactions has over the coming years.

2.8 Appendix A: Prior distributions

We use the conjugate priors for each parameter in the model, which correspond to Beta distributions

$$q_1 \quad \text{Beta}(\xi, \zeta) \tag{2.18a}$$
$$q_2 \quad \text{Beta}(\xi, \zeta) \tag{2.18b}$$
$$p \quad \text{Beta}(\xi, \zeta) \tag{2.18c}$$
$$q \quad \text{Beta}(\xi, \zeta). \tag{2.18d}$$

In all experiments we set $\xi = 1.1$ and $\zeta = 5$ which encourages sparsity while discouraging the complete removal of a interaction types (a null probability).

As discussed in the main text, we address the potential label switching problem of edge types by imposing an order for the parameters $\mu = (\mu_0, \mu_1, \mu_2)$, which can be viewed as a prior on these parameters [91]. For the categorical-edge model, we impose a total ordering $\mu_0 < \mu_1 < \mu_2$ while the correlations produced by the triangles of the hypergraph model allow us to only assume the partial order $\mu_0 < \mu_1$ and $\mu_0 < \mu_2$ under the assumption that the difference of hyperedge size is sufficient to break symmetries. These considerations translate into the following conjugate distributions for the categorical-edges model

$$\mu_0 \quad \text{Gamma}(\alpha_0, \beta_0) \tag{2.19a}$$
$$\mu_1|\mu_0 \quad \text{TruncGamma}_{(\mu_0, \infty)}(\alpha_1, \beta_1) \tag{2.19b}$$
$$\mu_2|\mu_1 \quad \text{TruncGamma}_{(\mu_1, \infty)}(\alpha_2, \beta_2), \tag{2.19c}$$

and for the hypergraph model we have

$$\mu_0 \quad \text{Gamma}(\alpha_0, \beta_0), \tag{2.20a}$$
$$\mu_1|\mu_0 \quad \text{TruncGamma}_{(\mu_0, \infty)}(\alpha_1, \beta_1) \tag{2.20b}$$
$$\mu_2|\mu_0 \quad \text{TruncGamma}_{(\mu_0, \infty)}(\alpha_2, \beta_2). \tag{2.20c}$$

We use the following probability density functions for $x \quad \text{Gamma}(\alpha, \beta)$ and $y \quad \text{TruncGamma}_{(c,d)}(\alpha, \beta)$:

$$f(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{2.21}$$
$$g(y) = \frac{\mathbb{1}_{(c,d)}(y)}{\gamma(d, \alpha) - \gamma(c, \alpha)} \frac{1}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y}, \tag{2.22}$$

where γ is the lower incomplete gamma function and $\mathbb{1}$ is the indicator function. In all our numerical experiments, we set the priors $\alpha_0 = \alpha_1 = \alpha_2 = 1.05$ and $\beta_0 = \beta_1 = \beta_2 = 0.5$. In another inference setting, these should be adjusted to reflect prior knowledge about the dataset X.

2.9 Appendix B: Sampling algorithms

We use a Gibbs sampler to sample the joint posterior distribution $P(S, \theta|X)$, where $\theta = \{\mu, \varphi\}$. This class of algorithms allows us to sample from arbitrary joint distributions by sampling from each of its conditional distributions in alternance, here the parameter distribution $P(\theta|S, X)$ and the structural distribution $P(S|X, \theta)$. In what follows, we derive these sampling distributions and determine algorithms that generate samples from them.

2.9.1 Sampling the parameters

We break down the sampling of the parameters θ in sequential sampling steps for each of the individual parameters, meaning that when sampling from $P(\theta|S, X)$, each parameter is conditionally independent to the others. This marginal distribution is noted $P(\theta\ |\theta_{-\theta}, S, X)$ where $\theta_{-\theta}$ represents all parameters excluding θ. Using Bayes' formula, one can see that this distribution is proportional to the posterior distribution

$$P(\theta\ |\theta_{-\theta}, S, X) = \frac{P(S, \theta|X)}{P(S|X)} \quad P(S, \theta|X). \tag{2.23}$$

Using Eqs. (2.2), (2.6) and (2.18), we directly find that

$$q_1|\theta_{-q_1}, G, X \quad \text{Beta}(m_1 + \xi, \tbinom{n}{2} - m_1 - m_2 + \zeta) \tag{2.24a}$$
$$q_2|\theta_{-q_2}, G, X \quad \text{Beta}(m_2 + \xi, \tbinom{n}{2} - m_2 + \zeta) \tag{2.24b}$$
$$q|\theta_{-q}, H, X \quad \text{Beta}(h_1 + \xi, \tbinom{n}{2} - h_1 + \zeta) \tag{2.24c}$$
$$p|\theta_{-p}, H, X \quad \text{Beta}(h_2 + \xi, \tbinom{n}{3} - h_2 + \zeta), \tag{2.24d}$$

which are all beta distributions. A random variable z Beta(a, b) can be sampled rapidly with standard univariate sampling methods available in most statistical software packages, for example as $z = x/(x + y)$ where x Gamma(a) and y Gamma(b) [92].

To sample the parameters $\mu = (\mu_0, \mu_1, \mu_2)$, we rearrange the product inside Eq. (2.1) as

$$P(X|S, \theta) = \prod_{i<j} \frac{1}{x_{ij}!} \prod_{k=0}^{2} \mu_k^{X^{(k)}} e^{-\mu_k L^{(k)}} \tag{2.25}$$

where

$$X^{(k)} = \sum_{i<j} x_{ij} \delta_{k,\ ij} \tag{2.26}$$
$$L^{(k)} = \sum_{i<j} \delta_{k,\ ij} \tag{2.27}$$

are, respectively, the sum of observations with label k and the number of pairs with label k, and where δ being the Kronecker delta. Combining Eqs. (2.6) and (2.23) yields for the

categorical-edges graph model

$$\mu_0|\theta_{-\mu_0}, G, X \quad \text{TruncGamma}_{(0,\mu_1)}(X^{(0)}+\alpha_0, L^{(0)}+\beta_0) \tag{2.28a}$$
$$\mu_1|\theta_{-\mu_1}, G, X \quad \text{TruncGamma}_{(\mu_0,\mu_2)}(X^{(1)}+\alpha_1, L^{(1)}+\beta_1) \tag{2.28b}$$
$$\mu_2|\theta_{-\mu_2}, G, X \quad \text{TruncGamma}_{(\mu_1,\infty)}(X^{(2)}+\alpha_2, L^{(2)}+\beta_2). \tag{2.28c}$$

Combining Eqs. (2.2) and (2.23) yields for the hypergraph model

$$\mu_0|\theta_{-\mu_0}, H, X \quad \text{TruncGamma}_{(0,\mu_-)}(X^{(0)}+\alpha_0, L^{(0)}+\beta_0) \tag{2.29a}$$
$$\mu_1|\theta_{-\mu_1}, H, X \quad \text{TruncGamma}_{(\mu_0,\infty)}(X^{(1)}+\alpha_1, L^{(1)}+\beta_1) \tag{2.29b}$$
$$\mu_2|\theta_{-\mu_2}, H, X \quad \text{TruncGamma}_{(\mu_0,\infty)}(X^{(2)}+\alpha_2, L^{(2)}+\beta_2) \tag{2.29c}$$

where $\mu_- = \min\{\mu_1, \mu_2\}$.

Since this step is revisited often by our algorithm, we combine three sampling methods to ensure rapid and accurate sampling in all cases [93]: rejection sampling using a gamma distribution if the rejection probability is low, a more costly inverse transform sampling using incomplete gamma inverse function, and rejection sampling with an adjusted "linear distribution" if all other methods fail. The main interest in using the linear distribution is that it provides a good approximation of the density for small intervals. The inverse transform sampling often works, but can suffer from numerical instabilities especially for small truncation intervals.

We define the linear probability density function as

$$f(x) = \frac{1 + cx}{2}, \quad x, c \quad [-1, 1] \tag{2.30}$$

where c is the slope. A sample from this distribution is obtained using its inverse cumulative distribution function

$$\text{CDF}^{-1}(u) = \frac{\sqrt{c^2 - 2c + 4cu + 1} - 1}{c} \tag{2.31}$$

where u is a continuous random variable uniformly distributed on $[0, 1]$.

In the rejection sampling algorithm, the support of this distribution is adjusted to match the truncated gamma distribution and c is the slope of a line connecting the truncated gamma density evaluated at the lower bound to the density evaluated at the upper bound.

2.9.2 Sampling graphs with categorical edges

The distribution used to sample the categorical-edges graph model is derived by following a similar reasoning as for Eq. (2.23). We first observe that

$$P(S|\theta, X) = \frac{P(S, \theta|X)}{P(\theta|X)} \quad P(S, \theta|X). \tag{2.32}$$

Combining this expression with Eqs. (2.9) and (2.18) yields

$$P(G|\theta, X) \quad q_1^{m_1+\xi-1}(1-q_1)^{\binom{n}{2}-m_1-m_2+\zeta-1} q_2^{m_2+\xi-1}(1-q_2)^{\binom{n}{2}-m_2+\zeta-1} \times \prod_{i<j} \frac{(\mu_{ij})^{x_{ij}}}{x_{ij}!} e^{-\mu_{ij}}. \tag{2.33}$$

The edge labels $_{ij}$ induce complicated interactions between the parameters, so we turn to a Metropolis-Hastings (MH) algorithm to generate samples from this distribution as it does not appear to correspond to a well known closed-form distribution.

The MH algorithm is initialized at the ground truth hypergraph projection and the ground truth parameters except in Table 2.1 where it is initialized at a graph with no strong edges and weak edges wherever $x_{ij} > 0$ and at parameters μ and φ_G set to the maximum likelihood estimator obtained from a Poisson mixture model. At each iteration, we propose to increment a interaction type with probability η and to decrement a interaction type with probability $1 - \eta$. We use $\eta = 0.5$ in our numerical simulations.

If the algorithm reaches a point where the graph is fully connected with strong edges (or empty), than we propose to decrement (or increment) a type with probability 1. The pair (i, j) whose type is to be decremented is chosen uniformly among all pairs whose type is not zero. The pair (i, j) whose type is to be incremented is chosen proportionally to the weight

$$w_{ij} = \begin{cases} x_{ij} + 1 & \text{if } _{ij} < 2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.34}$$

The proposal probability of a new graph G conditioned on the current graph G is

$$Q(G \,|G, X) = a \frac{\eta w_{ij}}{\sum_{i<j} w_{ij}} + (1-a)\frac{1-\eta}{m_1 + m_2} \tag{2.35}$$

where $a = 1$ if the label is to be incremented and $a = 0$ if it is to be decremented. Finally, the proposal is accepted with probability

$$\alpha(G \,|G) = \min\left[1, \frac{P(G\,, \theta|X)Q(G|G\,, X)}{P(G, \theta|X)Q(G\,|G, X)}\right] \tag{2.36}$$

where $Q(G|G\,, X)$ is the probability of reverting the proposed move.

2.9.3 Sampling hypergraphs

Combining Eqs. (2.8), (2.18) and (2.32), we find

$$P(H|\theta, X) \quad \frac{P(\theta)}{P(X)} q^{h_1+\xi-1}(1-q)^{\binom{n}{2}-h_1+\zeta-1} p^{h_2+\xi-1}(1-p)^{\binom{n}{3}-h_2+\zeta-1} \times \prod_{i<j} \frac{(\mu_{ij})^{x_{ij}}}{x_{ij}!} e^{-\mu_{ij}}, \tag{2.37}$$

which, again, is not a standard distribution. Hence we use a MH algorithm to generate samples of it in a similar fashion as for the categorical-edges graph model.

The MH algorithm is initialized at the ground truth hypergraph and parameters except in Table 2.1 where it is initialized at a hypergraph with no 3-edge and 2-edges wherever $x_{ij} > 0$ and at parameters μ and φ_H set to the maximum likelihood estimator obtained from a Poisson mixture model. At each iteration, one of six possible moves is proposed:

1. add ($a = 1$) a 2-edge with probability $\nu_2\eta$;
2. remove ($a = 0$) a 2-edge with probability $\nu_2(1-\eta)$;
3. add ($a = 1$) a 3-edge with probability $\nu_3\eta$;
4. remove ($a = 0$) a 3-edge with probability $\nu_3(1-\eta)$;
5. add ($a = 1$) hidden 2-edges with probability $(1-\nu_2-\nu_3)\eta$;
6. remove ($a = 0$) hidden 2-edges with probability $(1-\nu_2-\nu_3)(1-\eta)$.

We use $\eta = 0.5$ and $\nu_2 = \nu_3 = 0.4999$.

If the algorithm reaches a point where either no 2-edge or 3-edge can be added (removed), then one is removed (added) with probability 1. If a move in which hidden 2-edges should be added/removed has been picked and that move is not possible (e.g. there are no hidden 2-edge to be removed), a completely new move is randomly chosen.

The proposed move, that would transform the hypergraph H into a new one H', is accepted with probability

$$\alpha(H'|H) = \min\left(1, \frac{P(H', \theta|X)Q(H|H', X)}{P(H, \theta|X)Q(H'|H, X)}\right). \tag{2.38}$$

We now detail the proposal probability ratio $\frac{Q(H|H',X)}{Q(H'|H,X)}$ for each of the 6 possible moves.

When a 3-edge is to be removed, it is chosen uniformly among the existing 3-edges. When a 3-edge is to be added, the three vertices (i, j, k) are chosen in three steps: pick $i \sim P(i)$, pick $j \sim P(j|i)$ and pick $k \sim P(k|i)$ where

$$P(i) = \frac{\prod_{l=i}(x_{il}+1)}{\prod_{r}\prod_{s=r}(x_{rs}+1)} \tag{2.39a}$$

$$P(j|i) = \frac{x_{ij}+1}{\prod_{l=i}(x_{il}+1)}. \tag{2.39b}$$

Since the order in which vertices are chosen does not matter, the probability that triplet (i, j, k) is chosen is

$$P(i,j,k) = 2P(i)P(j|i)P(k|i) + 2P(j)P(i|j)P(k|j) + 2P(k)P(i|k)P(j|k). \tag{2.40}$$

If this selection process results in the triplet (i, j, j) or chooses an existing 3-edge, then the proposed move is automatically rejected since the distribution is only supported on *simple* hypergraphs. Altogether, the proposal probability ratio for moves involving 3-edges can be summarized as

$$\frac{Q(H|H\ ,X)}{Q(H\ |H,X)} = \left[\frac{1}{\eta P(i,j,k)}\frac{1-\eta}{h_2+a}\right]^{2a-1}. \tag{2.41}$$

When a 2-edge needs to be removed, it is chosen uniformly among the existing 2-edges. When a 2-edge (i, j) needs to be added, it is chosen proportionally to the weight

$$\omega_{ij} = \begin{cases} x_{ij}+1 & \text{if } (i,j) \quad E \\ 0 & \text{otherwise.} \end{cases} \tag{2.42}$$

Altogether, the proposal probability ratio for moves involving 2-edges can be summarized as

$$\frac{Q(H|H\ ,X)}{Q(H\ |H,X)} = \left[\frac{\sum_{r<s}\omega_{rs}+a(x_{ij}+1)}{\eta(x_{ij}+1)}\frac{1-\eta}{h_1+a}\right]^{2a-1} \tag{2.43}$$

Our definition of the types of interactions $_{ij}$ [Eq. (2.3)] implies that hidden 2-edges do not contribute to the likelihood [Eq. (2.1)]; their addition/removal depends solely on the hypergraph model. However, the *cost* of removing a 3-edge depends on the number of hidden 2-edges underneath. Because of this asymmetry, we found that running the MH algorithm with the four previous moves tend to get stuck with certain configurations of hidden 2-edges. Our solution has been to propose two additional moves specifically targeting hidden 2-edges.

To propose the addition/removal of hidden 2-edges, we first regroup every existing hidden 2-edges into a set C_0 and every "nonexistent" hidden 2-edges into a set C_1. (These nonexistent hidden 2-edges are interactions of type 2 for which the corresponding 2-edge does not exist.) We then draw the number m of 2-edges to add/remove from a truncated geometric distribution of parameter χ_a on the interval $[2, |C_a|]$. If $|C_a| < 2$, a new move is picked randomly as the chosen one cannot be performed. We force $m \geq 2$ to ensure these two additional moves do not overlap with the previous two moves involving 2-edges in the MH algorithm acceptance probabilities. Finally, we choose uniformly m hidden 2-edges in C_a and store them in the set e; their addition/removal consist in the proposed move. The probability of a given set e is

$$P(e|C_a,\chi_a) = \frac{(1-\chi_a)^{|e|-2}\chi_a}{1-(1-\chi_a)^{|C_a|-1}}\left[\frac{|C_a|}{|e|}\right]^{-1}, \tag{2.44}$$

and the proposal probability ratio for moves involving hidden 2-edges only is

$$\frac{Q(H|H\ ,X)}{Q(H\ |H,X)} = \left[\frac{1-\eta}{\eta}\right]^{2a-1}\frac{P(e|C_{1-a}\ e,\chi_{1-a})}{P(e|C_a,\chi_a)}. \tag{2.45}$$

In the simulations, we use $\chi_0 = 0.99$ and $\chi_1 = 0.01$ as we want to remove more frequently than add hidden 2-edges.

2.9.4 Convergence

We stop the two previous MH algorithms whenever the likelihood stabilizes, meaning the chains have reached stationarity. We consider that this has happened when the relative change in the average likelihood of the last W iterations is smaller than a tolerance parameter δ. We use $W = 20000$ and $\delta = 0.02$ in our simulations.

Further, to ensure the MH algorithm runs long enough but not too long, we set a minimum $I_{\min}$ and maximum $I_{\max}$ number of iterations. We adjust these values empirically with a test run, but they are roughly $I_{\min} = 10^5$ and $I_{\max} = 10^6$. Finally, for each posterior distribution sample, we run four chains and keep the one with the highest average likelihood.

2.10 Appendix C: Regime $\mu_1 > \mu_2$ and confusion matrices

We mentioned in Sec. 2.5.2 that conditions such as $\mu_1 < \mu_2$ or $\mu_1 > \mu_2$ need not be imposed in the prior distributions since 2-edges and 3-edges are fundamentally different. As a complement to the analysis presented in Sec. 2.6.3, we investigate the case where μ_2 is varied between $\mu_0 = 0.05$ and $\mu_1 = 50$.

Comparison between Figs. 2.5 and 2.6 and Figs. 2.7 and 2.8 suggests that both scenarios are quite similar, as expected. In particular, note that the apparent swap in the sums of residuals for the categorical-edges graph model is simply due to the redefinition of $_{ij}$ to accommodate the restriction that $\mu_1 < \mu_2$ in the model. Indeed we redefine

$$_{ij} = \begin{cases} 1 & (i,j) \quad E_2, \\ 2 & (i,j) \quad E_1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.46}$$

The only noteworthy difference between the two sets of simulations occurs when μ_2 approaches μ_0. We observe the same phenomenon than when μ_1 approaches μ_2 from the left: the information in the neighborhood used by the hypergraph model allows for a more accurate reconstruction (i.e., smaller , larger S). Interestingly, this effect is also apparent in the worst-case hypergraphs.

Figures 2.9, 2.10, 2.7d and 2.8d show the normalized confusion matrix for the best-case and worst-case hypergraphs. The entries of the normalized confusion matrix $\tilde{c}_{rs}$ are the proportion of interactions of type $_{ij} = r$ that were predicted as $\hat{}_{ij} = s$ by the model

$$\tilde{c}_{rs} = \frac{c_{rs}}{c_{r0} + c_{r1} + c_{r2}}. \tag{2.47}$$

For instance, the element $\tilde{c}_{21}$ is the proportion of projected 3-edges predicted as 2-edges in the hypergraph model.

When the categorical-edges graph model ends up inferring only one type of interaction, there are two equivalent reconstructed graphs: all interactions are weak edges or all interactions are strong edges. Noting that in less extreme cases, the model naturally favors strong edges due to the larger associated variance in the likelihood, we set all interactions to strong edges whenever it labels them all as a weak edges.

We see that for the best-case structure in Figs. 2.9 and 2.7d, the hypergraph model makes little to no error. As we increase μ_1, we also observe a gradual increase of the number of misclassified weak edges for the categorical-edges model. For the worst-case structure, the results in Figs. 2.10 and 2.8d show the the hypergraph model favors 3-edges and that the categorical-edges model favors strong edges.

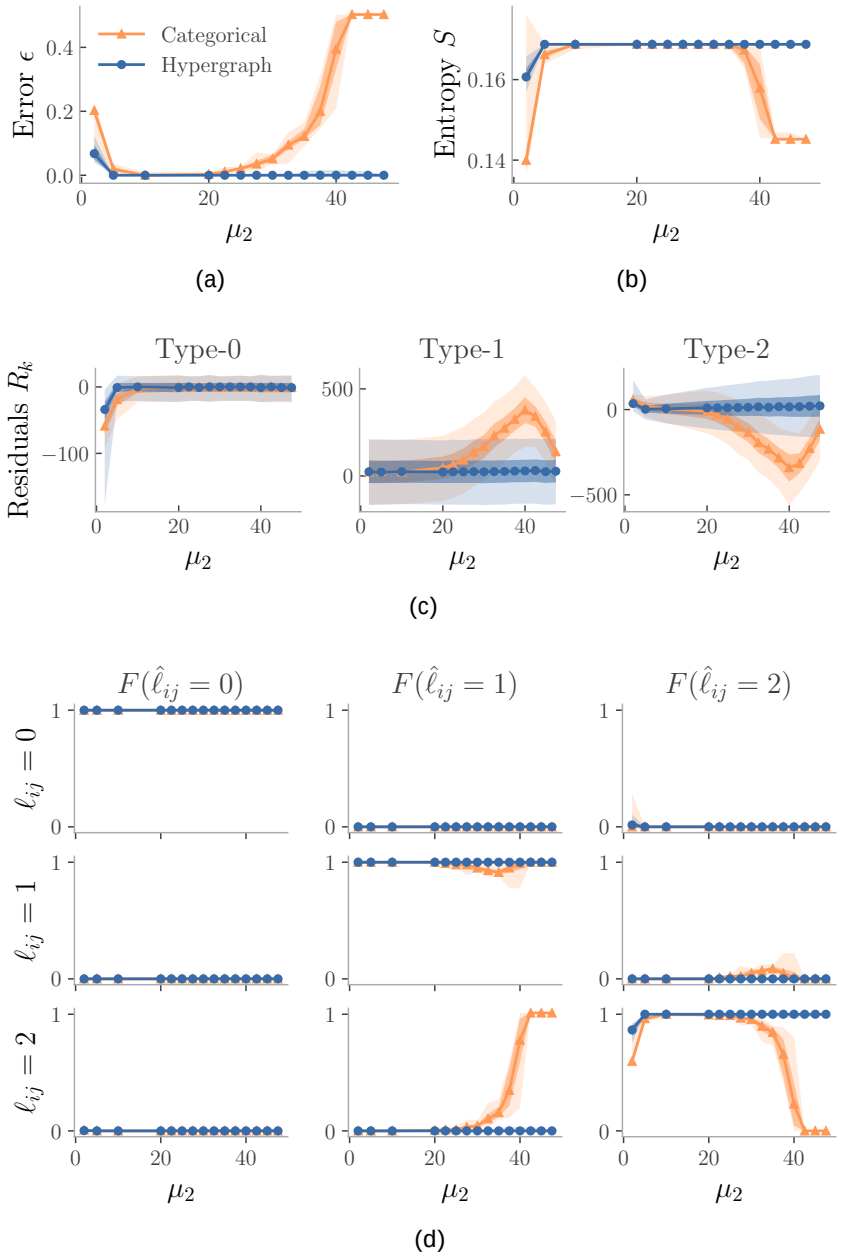


Figure 2.7: Impact of the measurement rate (μ_2) of type-2 interactions on the reconstruction of a best-case hypergraph. (a) Relative reconstruction error . (b) Entropy S. (c) Sums of residuals R_k. (d) Normalized confusion matrix. The observations were generated with $\mu_0 = 0.01$, $\mu_1 = 50$ and various μ_2 using the hypergraph model (blue) and the categorical-edges graph model (orange). The hypergraph model displays (a, d) less reconstruction errors (b) a larger entropy (c) lower residuals than the categorical-edges graph model, which indicates a better reconstruction. See the caption of Fig. 2.5 for details on the numerical experiment.

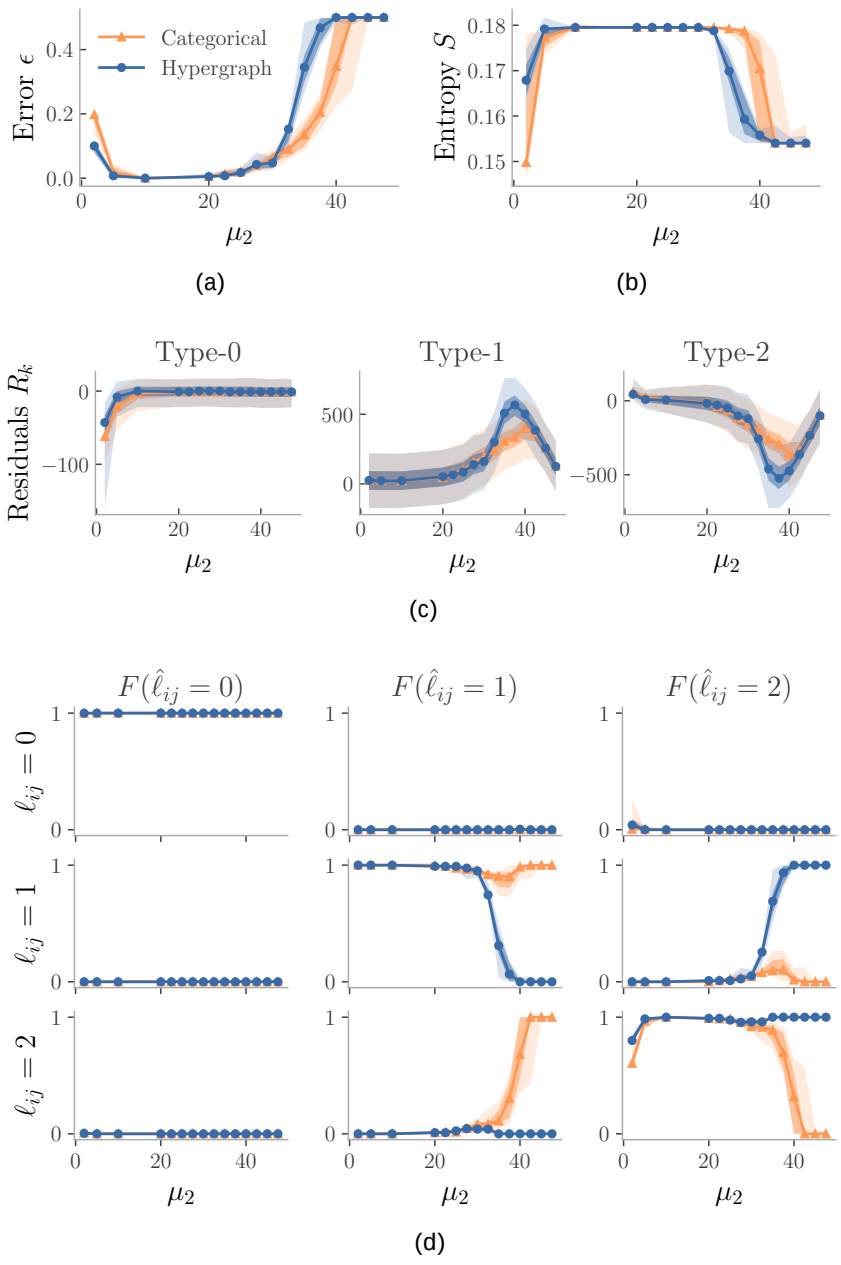


Figure 2.8: Impact of the measurement rate (μ_2) of type-2 interactions on the reconstruction of a worst-case hypergraph. (a) Relative reconstruction error . (b) Entropy S. (c) Sums of residuals R_k. (d) Normalized confusion matrix. The observations are generated with $\mu_0 = 0.01$, $\mu_1 = 50$ and various μ_2 using the hypergraph model (blue) and the categorical-edges graph model (orange). The hypergraph model displays (a, d) more reconstruction errors (b) a smaller entropy (c) greater residuals than the categorical-edges graph model, which indicates a worse reconstruction. See the captions of Figs. 2.5 and 2.6 for details on the numerical experiment.

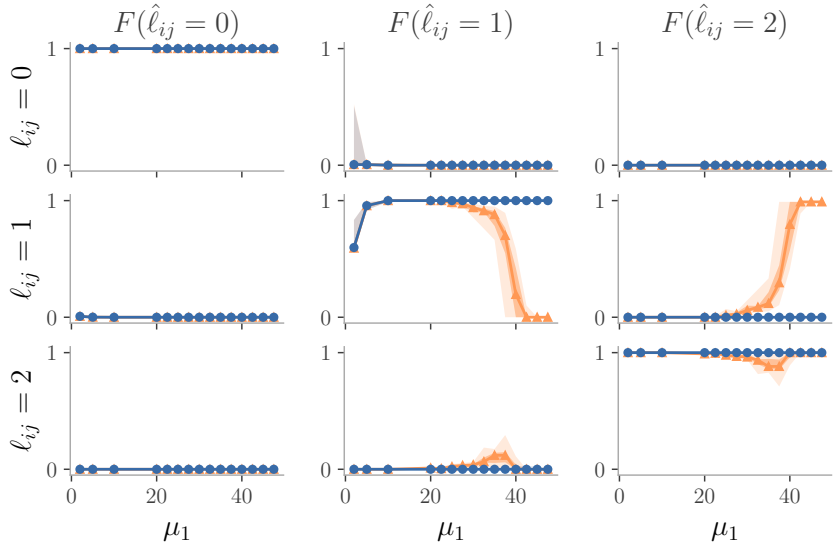


Figure 2.9: Normalized confusion matrix associated to the simulation of Fig. 2.5. The categorical-edges graph model favors the strong edges when μ_1 approaches μ_2, which leads to an inferior reconstruction compared to the hypergraph model that commits little to no error.

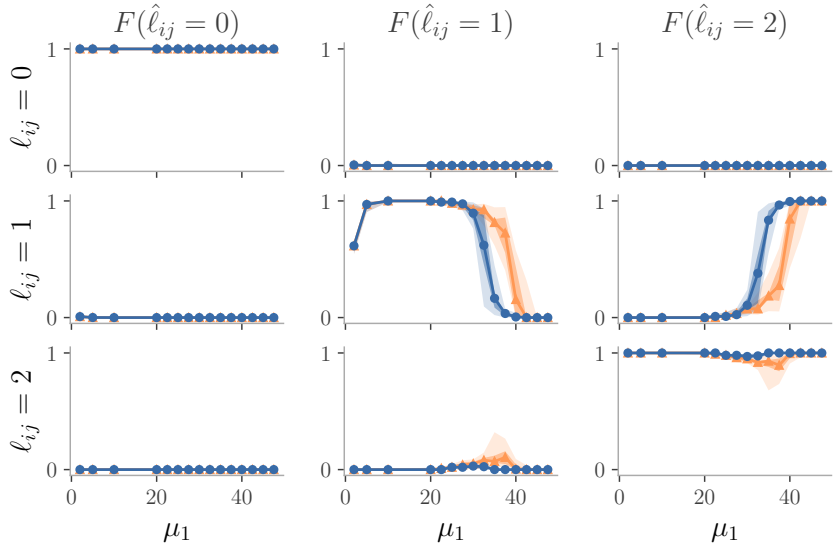


Figure 2.10: Normalized confusion matrix associated to the simulation of Fig. 2.6. While the categorical-edges model still favors strong edges to weak edges, the hypergraph model favors more strongly 3-edges and displays a worse performance for the worst-case hypergraph.

Conclusion

L'accumulation récente de résultats empiriques suggère l'insu□sance des graphes pour la modélisation des systèmes complexes réels : des objets mathématiques pouvant encoder les interactions d'ordre supérieur doivent être utilisés. Or, une grande partie des données de systèmes ne mesurent que les interactions dyadiques, et ces mesures sont souvent indirectes. Malgré la proportion considérable de ce type d'observations, l'importance des interactions d'ordre supérieur et l'étude minutieuse de la reconstruction de graphes dans la littérature, il n'existe aucune méthode à ce jour qui reconstruit la structure d'interactions d'ordre supérieur à partir de ces mesures.

Dans cette optique, le projet de maîtrise avait comme objectif de proposer une approche de reconstruction d'hypergraphes qui tient compte du bruit inhérent aux données. À partir de cette méthode, le but secondaire était de déterminer l'importance des corrélations induites par les interactions d'ordre supérieur au niveau de la précision de la reconstruction.

En se basant sur le contenu théorique du chapitre 1, deux modèles d'inférence bayésienne ont été introduits au chapitre 2 : un premier suppose un graphe avec des liens catégoriques (un lien est « fort » ou « faible ») comme structure d'interactions et un second suppose un hypergraphe latent comprenant des 2-liens et des 3-liens. Cette méthode se démarque en s'appliquant directement sur des mesures dyadiques bruitées. E□ectivement, Roy-Pomerleau [28] ainsi que Young et al. [29] reconstruisent les interactions d'ordre supérieur à partir de données de graphe supposées exactes et Santoro et al. [30] les reconstruisent à partir de séries temporelles, une autre forme commune de données.

Également au chapitre 2, les deux modèles d'inférence ont été comparés. Pour les di□érents hypergraphes reconstruits, il a été constaté que le modèle d'hypergraphe commet au plus le même nombre d'erreurs que le modèle de graphe aux liens catégoriques à l'exception d'un hypergraphe di□cile à inférer par construction. L'analyse a ainsi révélé un lien entre la structure de l'hypergraphe et la qualité de la reconstruction : moins il existe de 2-liens qui forment des triangles dans la projection, plus la reconstruction e□ectuée par le modèle d'hypergraphe est précise. Dans le cas idéal où aucun triangle projeté ne contient de 2-lien, le modèle détecte presque parfaitement les types d'interactions pour les di□érents régimes de paramètres, une nette amélioration par rapport au modèle de graphe. Toutefois, dans le pire cas où tous les 2-

liens font partie d'un triangle, l'erreur du modèle d'hypergraphe surpasse le modèle de graphe. De ce comportement on déduit que la restriction des 3-liens en triangles, une corrélation engendrée par la projection, est ce qui permet d'améliorer la précision de la reconstruction. Effectivement, contrairement au modèle de graphe qui ne considère que l'observation x_{ij} pour déterminer le type d'une paire $_{ij}$, le modèle d'hypergraphe s'informe de tous les triangles (x_{ij}, x_{ik}, x_{jk}) qui comprennent la paire (i, j) pour distinguer un 3-lien d'un 2-lien. Enfin, le modèle d'hypergraphe parvient généralement à de meilleurs résultats en raison de la faible abondance et concentration des interactions d'un réseau réel et des modèles d'hypergraphe aléatoire qui visent à reproduire les réseaux réels. Avec ces caractéristiques, la plupart des 2-liens ne forment pas de triangles.

Néanmoins, la contribution de ce projet est principalement conceptuelle. Malgré son excellente performance, le modèle bayésien dans sa forme actuelle modélise un nombre limité de jeux de données. En effet, seules les observations discrètes sont modélisées avec une loi de Poisson, et si les données n'ont pas la forme d'un mélange de lois de Poisson, la limitation en degrés de liberté de la vraisemblance risque de produire des résultats insatisfaisants. Ainsi, les lois du mélange devraient être adaptées afin de rendre l'approche plus utile pratiquement. Il est aussi à noter que le modèle d'hypergraphe requiert un coût computationnel plus important que le modèle de graphe, ce qui signifie qu'il pourrait être mis de côté dans un contexte ou l'importance des hyperliens est moindre.

Étant un premier pas dans la reconstruction d'interactions d'ordre supérieur, de nombreuses pistes d'exploration méritent d'être présentées. Notamment, le modèle d'hypergraphe se généralise à des hyperliens connectant plus de trois noeuds. Cependant, en raison de l'information limitée contenue dans une valeur scalaire d'une paire, l'inférence de ce modèle serait sans doute difficile. Ce faisant, il serait avantageux d'incorporer des données supplémentaires à la vraisemblance telles que des séries temporelles de noeuds, des mesures d'interactions d'ordre supérieur ou simplement d'autres types d'observations dyadiques. Un autre défi associé à cette généralisation provient du nombre grandissant d'interactions cachées. Effectivement, en utilisant la projection présentée, le nombre d'interactions cachées possibles croît exponentiellement avec la taille des hyperliens permise. Il serait donc préférable dans ce contexte de remplacer l'hypergraphe par un complexe simplicial, un cas particulier d'hypergraphe dans lequel l'existence d'une interaction de taille k implique l'existence de toutes les interactions de taille $k - 1$ connectant les mêmes noeuds (un 3-lien (i, j, k) implique les 2-liens (i, j), (i, k) et (j, k), par exemple).

Hormis cette généralisation à de plus grandes interactions, d'autres avenues d'exploration bonifieraient notre compréhension et l'applicabilité du modèle. Par exemple, généraliser le modèle d'hypergraphe à d'autres formes de vraisemblance lui permettrait de s'appliquer à un plus grand nombre de jeux de données, et déterminer les limites de détectabilité des interactions offrirait une manière de quantifier directement l'importance de la corrélation

pour la détection. Ces sujets sont explorés dans l'annexe C.

La reconstruction de la structure d'interactions est une tâche fondamentalement ardue. Compte tenu de la structure latente inconnue, la qualité de la modélisation est di□cilement quantifiée pour des observations réelles. En dépit de cette di□culté, l'inférence bayésienne procure une distribution plutôt qu'une seule estimation, ce qui favorise une interprétation nuancée et robuste. Grâce à ces méthodes de reconstruction, il est possible d'analyser la structure d'une grande variété de systèmes qui resterait autrement inconnue.

Annexe A

Algorithmes d'échantillonnage

Cette annexe présente des algorithmes d'échantillonnage utilisés dans les méthodes de MCMC du chapitre 2. Le contenu est basé sur les algorithmes des sections 1.6.2 et 1.6.3.

A.1 Loi géométrique tronquée

La fonction de masse de la loi géométrique tronquée est

$$\mathbb{P}(x) \quad (1-p)^x, \quad p \quad (0,1). \tag{A.1}$$

En considérant l'intervalle de troncature $[x_1, x_2]$, la constante de normalisation C est

$$\begin{aligned} C &= \sum_{x=x_1}^{x_2} (1-p)^x = \sum_{x=0}^{x_2-x_1} (1-p)^{x+x_1} = (1-p)^{x_1} \sum_{x=0}^{x_2-x_1} (1-p)^x \\ &= (1-p)^{x_1} \frac{1-(1-p)^{x_2-x_1+1}}{p}, \end{aligned} \tag{A.2}$$

Ainsi, la fonction de masse exacte est

$$\mathbb{P}(x) = \frac{p(1-p)^{x-x_1}}{1-(1-p)^{x_2-x_1+1}} \tag{A.3}$$

et la fonction de répartition est

$$\mathrm{CDF}(x) = \sum_{y=x_1}^{x} \mathbb{P}(y) = \frac{p}{1-(1-p)^{x_2-x_1+1}} \sum_{y=x_1}^{x} (1-p)^{y-x_1} = \frac{1-(1-p)^{x-x_1+1}}{1-(1-p)^{x_2-x_1+1}}. \tag{A.4}$$

Pour échantillonner cette loi, la méthode de la transformée inverse est idéale étant donné qu'une forme analytique existe pour la fonction de répartition inverse.

Pour générer une variable aléatoire discrète X à partir d'une variable aléatoire continue u $U(0,1)$, on cherche la plus petite réalisation x telle que $u \leq \text{CDF}(x)$. Ainsi,

$$\begin{aligned} u &\leq \frac{1-(1-p)^{x-x_1+1}}{1-(1-p)^{x_2-x_1+1}} \\ u[1-(1-p)^{x_2-x_1+1}] &\leq 1-(1-p)^{x-x_1+1} \\ (1-p)^{x-x_1+1} &\leq 1-u[1-(1-p)^{x_2-x_1+1}] \\ (x-x_1+1)\ln(1-p) &\leq \ln\{1-u[1-(1-p)^{x_2-x_1+1}]\}. \end{aligned} \tag{A.5}$$

Puisque $\ln(1-p) < 0$ pour $p < 1$,

$$x \geq \frac{\ln\{1-u[1-(1-p)^{x_2-x_1+1}]\}}{\ln(1-p)} + x_1 - 1. \tag{A.6}$$

Le plus petit entier respectant cette condition est donc

$$x = \left\lfloor \frac{\ln\{1-u[1-(1-p)^{x_2-x_1+1}]\}}{\ln(1-p)} \right\rfloor + x_1 - 1 \tag{A.7}$$

où $\lfloor \cdot \rfloor$ dénote la fonction partie entière.

A.2 Loi Gamma tronquée

La fonction de densité de la loi Gamma de paramètres α et β tronquée sur l'intervalle $[x_1, x_2]$ est

$$f(x) = \frac{\beta^\alpha}{\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)} x^{\alpha-1} e^{-\beta x}, \tag{A.8}$$

où $\gamma(\beta x, \alpha) := \int_0^{\beta x} t^{\alpha-1} e^{-t} \mathrm{d}t$ est la fonction gamma incomplète inférieure.

Contrairement à la loi géométrique tronquée, une expression analytique est inconnue pour la fonction de répartition inverse de la loi Gamma tronquée. Il faut alors utiliser une approximation numérique ou une méthode alternative comme la méthode du rejet. Dans cette section, les deux approches sont présentées.

A.2.1 Méthode de la transformée inverse

La méthode de la transformée inverse est implémentée à l'aide la librairie C++ Boost[1] qui permet d'estimer les fonctions gamma incomplètes ainsi que leurs inverses de manière numérique. Pour obtenir une fonction de répartition inverse compatible avec la fonction gamma incomplète inverse, le changement de variable $x = g(x) = \beta x$ doit être effectué. Cette transformation étant une fonction monotone, il suffit d'appliquer la correction du jacobien à la

[1] https://www.boost.org

densité

$$\begin{aligned} f(x) &= f(g^{-1}(x)) \left| \frac{\mathrm{d}g^{-1}(x)}{\mathrm{d}x} \right| \\ &= \frac{\beta^{\alpha}}{\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)} (x/\beta)^{\alpha-1} e^{-x} \cdot \left| \frac{1}{\beta} \right| \\ &= \frac{1}{\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)} (x)^{\alpha-1} e^{-x}, \end{aligned} \tag{A.9}$$

où $g^{-1}(x) = x/\beta$.

La fonction de répartition pour x est

$$\mathrm{CDF}(x) = \frac{\gamma(\beta x, \alpha) - \gamma(\beta x_1, \alpha)}{\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)} \tag{A.10}$$

et sa fonction de répartition inverse est

$$\begin{aligned} u &= \mathrm{CDF}(x) \\ u[\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)] &= \gamma(x, \alpha) - \gamma(\beta x_1, \alpha) \\ \mathrm{CDF}^{-1}(u) &= \gamma^{-1}\left[u[\gamma(\beta x_2, \alpha) - \gamma(\beta x_1, \alpha)] + \gamma(\beta x_1, \alpha), \alpha\right], \end{aligned} \tag{A.11}$$

où $\gamma^{-1}(\cdot, \alpha)$ dénote l'inverse de la fonction gamma incomplète. Cette fonction inverse existe, car la fonction gamma incomplète est monotone croissante. Avec l'équation (A.11) et la transformation inverse, $\mathrm{CDF}^{-1}(u)/\beta$ $f(x)$ où u $U(0,1)$.

A.2.2 Méthode du rejet

L'échantillonnage d'une loi Gamma tronquée peut également s'effectuer grâce à la méthode du rejet. Cette sous-section présente l'algorithme utilisant la loi de proposition Gamma et la loi de proposition linéaire.

La loi Gamma est généralement une excellente loi de proposition pour la loi tronquée : comme les lois sont identiques à une constante près, cette constante est contenue dans M et les valeurs y proposées dans l'intervalle $[x_1, x_2]$ sont acceptées avec probabilité 1

$$\mathbb{P}(\text{Accepter } y | y) = \mathbb{1}_{[x_1, x_2]}(y). \tag{A.12}$$

La probabilité qu'une valeur proposée soit rejetée est donc la probabilité qu'elle soit hors de l'intervalle de troncature

$$\mathbb{P}(y \quad [x_1, x_2]) = \gamma(x_1, \alpha) + \Gamma(x_2, \alpha), \tag{A.13}$$

où $\Gamma(x, \alpha) := \int_x^{\infty} t^{\alpha-1} e^{-t} \mathrm{d}t$ est la fonction gamma incomplète supérieure.

Cependant, la probabilité de rejet (A.13) peut être grande lorsque l'intervalle $[x_1, x_2]$ est petit. Le cas échéant, la densité linéaire introduite à la section 2.9.1 est mieux adaptée. En effet,

puisque la densité de la loi Gamma est lisse, une droite en est une approximation raisonnable sur un petit intervalle.

La pente c de la loi de proposition est celle de la ligne reliant $f(x_2)$ à $f(x_2)$

$$c = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \tag{A.14}$$

et M est le ratio des maxima

$$M = \frac{\max\limits_{x\ [x_1,x_2]} h(x)}{\max\limits_{x\ [x_1,x_2]} f(x)}, \tag{A.15}$$

où h est la loi linéaire. Le maximum de h se situe à x_1 si $f(x_1) > f(x_2)$ et à x_2 autrement. Le maximum de la loi Gamma tronquée f est au mode $(\alpha - 1)/\beta$ si celui-ci est compris dans le support. Autrement, il se situe au même endroit que h. La probabilité d'accepter une valeur y dans l'algorithme est donnée par l'équation (1.83).

Annexe B

Complexité algorithmique

La complexité algorithmique d'une méthode numérique est un aspect important à considérer. Effectivement, un algorithme peut s'avérer trop coûteux en mémoire ou en temps dans certains contextes, ce qui limite sa praticité.

Cette annexe détaille la complexité algorithmique des différentes opérations effectuées dans les méthodes de MCMC du chapitre 2. L'expression $O(f(n))$ dénotera l'ensemble des fonctions dont le rythme de croissance est borné supérieurement par $f(n)$. Formellement, $O(f(n))$ est l'ensemble des fonctions g pour lesquels il existe un c et un n_0 de sorte que $|g(n)| < cf(n)$ pour tout $n \geq n_0$ [94].

B.1 Échantillonnage de la structure

La structure de données utilisée pour le graphe aux liens catégoriques est la *liste d'adjacence*. Dans cette structure de données, une liste de voisins et d'entiers qui encodent si l'interaction est de type 1 ou 2 est associée à chaque noeud. Ce faisant, ajouter un lien dans un graphe a une complexité constante $O(1)$. De plus, retirer un lien, déterminer l'existence d'un lien et identifier ou modifier le type d'interaction d'une paire (i, j) possèdent une complexité de $O(k_i + k_j)$, où k_i est le nombre de voisins du noeud i.

Dans l'algorithme de MH du modèle de graphe aux liens catégoriques, deux types de pas sont proposés : une incrémentation du type d'interaction d'une paire (i, j) pour laquelle $_{ij} < 2$ et une décrémentation du type d'une paire pour laquelle $_{ij} > 0$. En stockant les paires ayant un type d'interaction non nul et ayant un type d'interaction différent de 2 dans des structures *SamplableSet*[1], les propositions possèdent une complexité constante $O(1)$. Dans le calcul de la probabilité d'acceptation de l'équation (2.35), le type d'interaction de la paire (i, j) doit être déterminé, ce qui implique une complexité $O(\min(k_i, k_j))$ (il suffit de chercher dans la

[1] Afin de simplifier les expressions obtenues, la complexité $\mathcal{O}(\log\log(w_{\max}/w_{\min}))$ de la librairie *SamplableSet* (https://github.com/gstonge/SamplableSet) est considérée constante $\mathcal{O}(1)$.

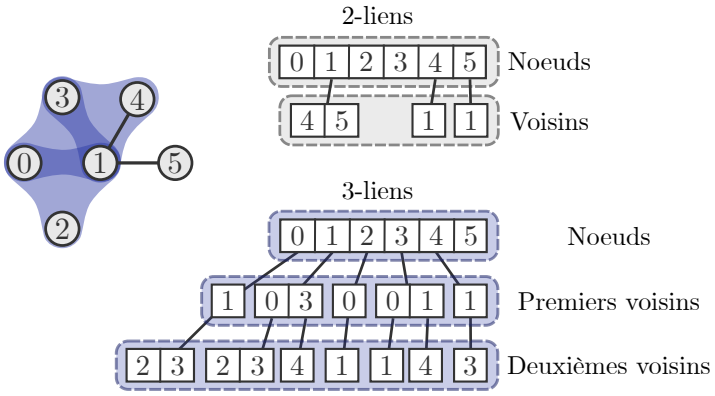


Figure B.1 : Structure de donnée utilisée pour les hypergraphes.

plus courte liste d'adjacence).

Pour contenir les hypergraphes, une structure de données similaire aux listes d'adjacence est utilisée. De manière identique au graphe aux liens catégoriques, les 2-liens de l'hypergraphe sont contenus dans des listes d'adjacence. Les 3-liens sont conservés dans des arbres (structure *map*[2]) de listes (structure *list*[3]) de la manière suivante : pour un 3-lien (i, j, k) où $i < j < k$, chaque noeud possède dans son arbre de *premiers voisins* le prochain plus petit noeud ($i \to j$, $j \to i$ et $k \to i$), et chaque premier voisin contient dans sa liste de *deuxièmes voisins* le noeud restant ($i \to j \to k$, $j \to i \to k$ et $k \to i \to j$). Un schéma de la structure de données est présenté à la figure B.1. Les opérations d'ajout, de retrait et de recherche d'un 3-lien ont une complexité de $O(\log k_i^{(1)} + k_{ij}^{(2)})$, où $k_i^{(1)}$ et $k_{ij}^{(2)}$ sont respectivement le nombre de premiers voisins d'un noeud et de deuxièmes voisins d'une paire de noeuds. Déterminer le type d'interaction d'une paire requiert alors une complexité $O(k + \log k^{(1)} + k^{(2)})$, où k est le nombre de 2-liens connectés au noeud.

Dans l'algorithme d'échantillonnage d'hypergraphe, les 3-liens retirés sont choisis uniformément parmi ceux qui sont existants. Pour ce faire, un noeud i est choisi proportionnellement à son nombre de 3-liens $\sum_j k_{ij}^{(2)}$, puis un de ses 3-liens est choisi uniformément. Identifier le n-ième 3-lien connecté à un noeud requiert une complexité $O(k^{(1)} + k^{(2)})$. Les 3-liens ajoutés sont proposés en temps constant $O(1)$, car la procédure n'implique que trois tirages de distributions discrètes dont les probabilités sont constantes. De plus, puisque les 2-liens existants et les 2-liens inexistants sont contenus dans deux structures *SamplableSet*, la proposition d'un retrait et celle d'un ajout de 2-lien requièrent une complexité constante $O(1)$. Finalement, tous les 2-liens cachés doivent être identifiés pour les pas les impliquant. Ainsi, pour chaque 3-lien (i, j, k), il faut déterminer si les 2-liens (i, j), (j, k) et (i, k) existent. La complexité résultante est $O(k_{\max}|T|)$ où $k_{\max}$ est le nombre maximal de 2-liens connectés à un noeud et $|T|$ est le nombre de 3-liens.

[2] https://en.cppreference.com/w/cpp/container/map
[3] https://en.cppreference.com/w/cpp/container/list

Dans l'algorithme de MH, la probabilité d'acceptation de l'équation (2.38) doit également être calculée. Pour les pas impliquant un 2-lien ou un 3-lien, le type d'interaction est requis et donc le calcul possède une complexité $O(k + \log k^{(1)} + k^{(2)})$. Cependant, comme le type d'interaction est 2 malgré la présence ou l'absence d'un 2-lien caché, le nouveau type d'interaction n'a pas besoin d'être déterminé et le calcul de la probabilité d'acceptation s'effectue en temps constant $O(1)$.

B.2 Échantillonnage des paramètres

L'échantillonnage des paramètres comporte deux étapes principales : calculer les paramètres des lois conditionnelles $P(\theta \mid \theta_{-\theta}, S, X)$ et les échantillonner. Or, la complexité de l'échantillonnage est négligée, car le calcul des paramètres a un plus grand impact. Dans ce dernier, l'évaluation des quantités $X^{(k)}$ et $L^{(k)}$ pour $k = 0, 1, 2$ (voir équations (2.26) et (2.27)) est coûteuse. Ce calcul requiert le type d'interaction de chaque pair d'indice, ce qui correspond à une complexité

$$O\left(\sum_{i<j} k_i\right) = O(nm) \tag{B.1}$$

et

$$\begin{aligned} O\left(\sum_{i=1}^{n}\sum_{j=1}^{n} (k_i + \log k_i^{(1)} + k_{ij}^{(2)})\right) &= O\left(nm + \sum_{i=1}^{n}\sum_{j=1}^{n} (\log k_i^{(1)} + k_{ij}^{(2)})\right) \\ &= O(nm + n \log k_{\max}^{(1)} + |T|) \end{aligned} \tag{B.2}$$

pour le modèle de graphe aux liens catégoriques et le modèle d'hypergraphe respectivement, où m est le nombre de 2-liens et où $k_{\max}^{(1)}$ est le nombre maximal de premiers voisins dans l'hypergraphe.

Annexe C

Contenu supplémentaire au projet de recherche

Cette annexe contient des idées qui pourraient être explorées afin d'améliorer notre compréhension et l'applicabilité des modèles développés. Le contenu est présenté à des fins d'archivage.

C.1 Proportions moyennes des types d'interaction

Dans le modèle d'hypergraphe du chapitre 2, les paramètres q et p contrôlant les probabilités d'existence des 2-liens et 3-liens ne s'interprètent pas directement comme les proportions des types d'interactions. En effet, une interaction peut être de type 2 malgré l'existence d'un 2-lien et un 3-lien génère plusieurs interactions de type 2.

Néanmoins, l'espérance des proportions des types d'interaction peut être obtenue. Une paire de noeuds interagit selon le type 2 s'il existe au moins un 3-lien qui l'inclut. Conséquemment, le type d'interaction est différent de 2 si les $n-2$ 3-liens incluant i et j sont absents. Par le complément de la probabilité et par le fait que le type d'interaction est 1 s'il existe un lien (probabilité q) et aucun 3-lien,

$$\mathbb{P}(\ _{ij} = 2|\varphi_H) = \mathbb{E}[\mathbb{1}[\ _{ij} = 2]] = 1 - (1-p)^{n-2} \tag{C.1a}$$

$$\mathbb{P}(\ _{ij} = 1|\varphi_H) = q(1-p)^{n-2} \tag{C.1b}$$

$$\mathbb{P}(\ _{ij} = 0|\varphi_H) = (1-q)(1-p)^{n-2}. \tag{C.1c}$$

Par souci de complétude, les proportions pour le modèle de graphe aux liens catégoriques

(*categorical edges graph*) sont

$$\mathbb{P}(\ _{ij} = 2|\varphi_G) = q_2 \tag{C.2a}$$

$$\mathbb{P}(\ _{ij} = 1|\varphi_G) = q_1(1 - q_2) \tag{C.2b}$$

$$\mathbb{P}(\ _{ij} = 0|\varphi_G) = (1 - q_1)(1 - q_2), \tag{C.2c}$$

où q_1 et q_2 sont respectivement la probabilité qu'un lien de type 1 et de type 2 existe.

C.2 Limite de détectabilité

Au chapitre 2, il a été constaté que la proportion d'interactions mal classifiées ( ) est liée au chevauchement des lois du mélange des observations. Dans cette section, on s'intéresse à la probabilité que le modèle commette les di□érents types d'erreurs de classification dans le cas idéal où les vrais paramètres μ et φ_G sont connus.

Dans ce contexte, le type d'interaction la plus probable d'une paire (i, j) est

$$\begin{aligned}
\hat{}_{ij} &= \operatorname*{argmax}_{k=0,1,2} \mathbb{P}(\ _{ij} = k|X, \mu, \varphi) \\
&= \operatorname*{argmax}_{k=0,1,2} \mathbb{P}(\ _{ij} = k, X, \mu, \varphi) \\
&= \operatorname*{argmax}_{k=0,1,2} \mathbb{P}(X|\mu,\ _{ij} = k)\mathbb{P}(\ _{ij} = k|\varphi),
\end{aligned} \tag{C.3}$$

en utilisant le théorème de Bayes. Dans le modèle d'hypergraphe, la présence d'une interaction de type 2 implique la présence de deux d'autres interactions dyadiques du même type. Toutefois, cette corrélation est inexistante pour la structure en graphe aux liens catégoriques, de sorte que seule l'observation x_{ij} impacte $\hat{}_{ij}$

$$\hat{}_{ij} = \operatorname*{argmax}_{k=0,1,2} \mathbb{P}(x_{ij}|\ _{ij} = k, \mu)\mathbb{P}(\ _{ij} = k|\varphi_G). \tag{C.4}$$

Pour ce modèle, la probabilité d'identifier le type d'interaction $\hat{}_{ij} = a$ d'une paire (i, j) dont le type d'interaction est $_{ij} = b$ est donc

$$\begin{aligned}
\mathbb{P}\left(\hat{}_{ij} = a|\ _{ij} = b, \mu, \varphi_G\right) &= \sum_{x_{ij}=0}^{\infty} \mathbb{P}\left(\hat{}_{ij} = a, x_{ij}\middle|\ _{ij} = b, \mu, \varphi_G\right) \\
&= \sum_{x_{ij}=0}^{\infty} \mathbb{P}\left(\hat{}_{ij} = a\middle|x_{ij}, \mu, \varphi_G\right) \mathbb{P}(x_{ij}|\ _{ij} = b, \mu) \\
&= \sum_{x_{ij}=0}^{\infty} \mathbb{1}_{\{\hat{}_{ij}=a\}}(x_{ij})\mathbb{P}(x_{ij}|\ _{ij} = b, \mu),
\end{aligned} \tag{C.5}$$

où $\hat{}_{ij}$ est indépendant de $_{ij}$ et où $\{\hat{}_{ij} = a\}$ est l'ensemble des x_{ij} pour lesquels $\hat{}_{ij} = a$. L'équation (C.5) est « l'aire sous la courbe » de la fonction de masse de $x_{ij}|\ _{ij} = b, \mu$ dans la région où $\hat{}_{ij} = a$ (voir la figure C.1).

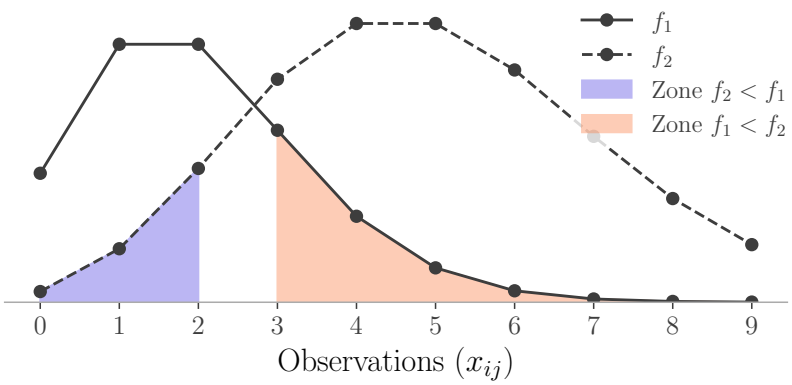


Figure C.1 : Chevauchement entre deux lois de Poisson.

Afin de trouver $\{\hat{}_{ij} = a\}$, on note que pour deux lois de Poisson f_1 de paramètres λ_1 et λ_2 respectivement, l'ensemble sur lequel $w_1 f_1 < w_2 f_2$, où $w_1, w_2 \quad (0, 1)$, est

$$\begin{aligned} w_1 \frac{(\lambda_1)^x}{x!} e^{-\lambda_1} &< w_2 \frac{(\lambda_2)^x}{x!} e^{-\lambda_2} \\ e^{\lambda_2 - \lambda_1} &< \left(\frac{\lambda_2}{\lambda_1}\right)^x \frac{w_2}{w_1} \\ \lambda_2 - \lambda_1 &< x \ln \frac{\lambda_2}{\lambda_1} + \ln \frac{w_2}{w_1}. \\ x &> \frac{1}{\ln \lambda_2 - \ln \lambda_1} (\lambda_2 - \lambda_1 + \ln w_2 - \ln w_1), \end{aligned} \tag{C.6}$$

où $\lambda_1 < \lambda_2$ est supposé sans perte de généralité.

Puisque $\mu_0 < \mu_1 < \mu_2$ dans le modèle de graphe aux liens catégoriques, on déduit de l'équation (C.6) que $\{\hat{}_{ij} = 0\} = [0, x_1)$, $\{\hat{}_{ij} = 1\} = [x_1, x_2)$ et $\{\hat{}_{ij} = 2\} = [x_2, \infty)$. Les quantités x_1 et x_2 s'obtiennent par l'évaluation du côté droit de l'équation (C.6) en utilisant les paramètres $\lambda_1 = \mu_0$ et $\lambda_2 = \mu_1$, et $\lambda_1 = \mu_1$ et $\lambda_2 = \mu_2$ respectivement.

En notant $\{\hat{}_{ij} = a\} = [x_{a,\min}, x_{a,\max})$, l'équation(C.5) s'écrit à l'aide de la fonction de répartition de la vraisemblance

$$\begin{aligned} \mathbb{P}(\hat{}_{ij} = a \mid {}_{ij} = b, \mu) &= \mathbb{P}(x_{ij} \quad [x_{a,\min}, x_{a,\max}] \mid {}_{ij} = b, \mu) \\ &= \mathrm{CDF}_{x_{ij} \mid {}_{ij} = b, \mu}(x_{a,\max}) - \mathrm{CDF}_{x_{ij} \mid {}_{ij} = b, \mu}(x_{a,\min}) \end{aligned} \tag{C.7}$$

où

$$\mathrm{CDF}_{x_{ij} \mid {}_{ij} = b, \mu}(x_{ij}) = \frac{\Gamma(x_{ij} + 1, \mu_b)}{x_{ij}!}. \tag{C.8}$$

La fonction de répartition de la loi de Poisson est définie à l'aide de la fonction gamma incomplète supérieure en raison de sa relation de récurrence. En e□ectuant une intégrale par

partie avec $u = t^{x-1}$ et $v = e^{-t}$,

$$\begin{aligned}\Gamma(x+1, r) &= \int_r^\infty t^x e^{-t}\mathrm{d}t \\ &= -\left[t^x e^{-t}\right]_r^\infty + x\int_r^\infty u^{x-1}e^{-u}\mathrm{d}u \\ &= r^x e^{-r} + x\Gamma(x, r). \end{aligned} \tag{C.9}$$

Comme $\Gamma(1, r) = \int_r^\infty e^{-t}\mathrm{d}t = e^{-r}$, on obtient par récurrence

$$\Gamma(x+1, r) = x!\sum_{y=0}^{x}\frac{1}{y!}r^y e^{-r}, \tag{C.10}$$

soit la fonction de répartition d'une loi de Poisson de paramètre r multipliée par $x!$.

L'équation (C.7) calcule la probabilité de commettre chaque type d'erreur de classification pour le modèle de graphe aux liens catégoriques. Ainsi, cette limitation intrinsèque serait intéressante à comparer à celle du modèle d'hypergraphe. Comme le modèle de graphe aux liens catégoriques n'utilise que la donnée x_{ij} pour prédire le type d'interaction d'une paire (i, j) contrairement au modèle d'hypergraphe qui tient compte des triangles (x_{ij}, x_{ik}, x_{jk}), on pourrait s'attendre à ce que la probabilité de commettre une erreur soit généralement plus petite pour le modèle d'hypergraphe.

C.3 Vraisemblance binomiale négative

La vraisemblance utilisée pour modéliser les mesures dyadiques au chapitre 2 suppose des lois de Poisson. Cette loi n'admettant qu'un paramètre λ, il est impossible de contrôler séparément sa variance et son espérance, ce qui a comme e et $\mathbb{V}[X] = \mathbb{E}[X] = \lambda$ pour X Poisson(λ).

On propose ainsi dans cette section une vraisemblance utilisant une loi binomiale négative. La fonction de masse de cette distribution est

$$\mathbb{P}(x) = \binom{x+r-1}{x}(1-p)^r p^x, \tag{C.11}$$

où p $[0, 1]$ et r $\mathbb{R}_+$. Cette loi se complémente à la loi de Poisson, car $\mathbb{E}[x] < \mathbb{V}[x]$ pour une variable aléatoire x binomiale négative.

La vraisemblance utilisant cette loi est alors

$$\mathbb{P}(X|r, p, S) = \prod_{i<j}\binom{x_{ij}+r_{ij}-1}{x_{ij}}(1-p_{ij})^{r_{ij}}p_{ij}^{x_{ij}}, \tag{C.12}$$

où r et p sont des vecteurs contenant les paramètres des di érentes classes. Nous supposons pour p les lois conjuguées indépendantes bêta d'hyperparamètres α et β et pour r des lois

uniformes continues $U(0, r_{\max})$ indépendantes

$$\mathbb{P}(p) = \prod_k \frac{1}{B(\alpha_k, \beta_k)} p_k^{\alpha_k - 1}(1 - p_k)^{\beta_k - 1}, \tag{C.13}$$

$$\mathbb{P}(r) = \prod_k \frac{1}{r_{\max}}. \tag{C.14}$$

Des échantillons de la loi *a posteriori* sont générés à l'aide d'un échantillonneur de Gibbs et des lois conditionnelles suivantes

$$\begin{aligned} \mathbb{P}\left(p_k | X, S, p_{-k}\right) &\propto p_k^{\alpha_k - 1}(1 - p_k)^{\beta_k - 1} \prod_{i<j} (1 - p_{ij})^{r_{ij}} p_{ij}^{x_{ij}} \qquad k \\ &\propto p_k^{\alpha_k + X^{(k)} - 1}(1 - p_k)^{\beta_k + L^{(k)} - 1} \qquad k \end{aligned} \tag{C.15}$$

où $L^{(k)} = L^{(k)} = \sum_{i<j} \delta_{k,\,ij}$ est repris de l'équation (2.27), et

$$\begin{aligned} \mathbb{P}(r_k | X, S, r_{-k}) &\propto \prod_{i<j} \binom{x_{ij} + r_{ij} - 1}{x_{ij}} (1 - p_{ij})^{r_{ij}} \qquad k \\ &\propto (1 - p_k)^{r_k L^{(k)}} \prod_{i<j} \binom{x_{ij} + r_{ij} - 1}{x_{ij}} \qquad k. \end{aligned} \tag{C.16}$$

Les lois $p_k | X, S, p_{-k}$ sont donc des lois bêta tandis que les lois $r_k | X, S, r_{-k}$ ont une forme inconnue.

On tente de simplifier l'équation (C.16) pour diminuer la complexité algorithmique de son évaluation. Pour ce faire, les r_k sont supposés entiers[1]. Dans l'équation (C.16), le coefficient binomial se réécrit

$$\binom{x_{ij} + r_k - 1}{x_{ij}} = \frac{(x_{ij} + r_k - 1)!}{x_{ij}!(r_k - 1)!} = \frac{(x_{ij} + r_k - 1) \cdots r_k}{x_{ij}!}. \tag{C.17}$$

Cette factorielle montante peut également s'écrire sous forme d'une série de puissance en r_k à l'aide des nombres de Stirling de première espèce non signés[2] [95]

$$r_k(r_k + 1) \cdots (r_k + x_{ij} - 1) = \sum_{s=0}^{x_{ij}} \left[{x_{ij} \atop s} \right] r_k^s. \tag{C.18}$$

Ainsi,

$$\mathbb{P}(r_k | X, S, \theta_{-r_k}) \propto (1 - p_k)^{r_k L^{(k)}} \prod_{\substack{i<j \\ ij = k}} \sum_{s=0}^{x_{ij}} \left[{x_{ij} \atop s} \right] r_k^s. \tag{C.19}$$

En regroupant les x_{ij} de même valeur,

$$\mathbb{P}(r_k | X, S, \theta_{-r_k}) \propto (1 - p_k)^{r_k L^{(k)}} \prod_{x \in \{x_{ij}\}} \sum_{s=0}^{x} \left[{x \atop s} \right] r_k^s \tag{C.20}$$

[1] La loi *a priori* est donc une loi uniforme discrète.
[2] Merci à Olivier Ribordy pour cette trouvaille.

où $\#(x)$ est le nombre d'éléments du triangle inférieur de X qui valent x et $\{x_{ij}\}$ est l'ensemble des éléments de X. Avec ces manipulations, le produit sur les paires (i, j) s'est transformé en un produit sur les différentes valeurs des éléments x_{ij}, soit un produit comportant généralement moins de termes. En utilisant la relation de récurrence

$$\left\{ {n+1 \atop k} \right\} = n \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\} \tag{C.21}$$

aux conditions initiales

$$\left\{ {0 \atop 0} \right\} = 1 \quad \text{et} \quad \left\{ {0 \atop n} \right\} = \left\{ {n \atop 0} \right\} = 0, \tag{C.22}$$

les nombres de Stirling peuvent être calculés efficacement. L'équation (C.20) pourrait possiblement être simplifiée à l'aide de fonctions génératrices.

Bibliographie

[1] Zaslavskiy, M., Bach, F. et Vert, J.-P. *Global alignment of protein–protein interaction networks by graph matching methods*. Bioinformatics **25**, i259–1267 (2009). doi :10.1093/bioinformatics/btp196.

[2] Ings, T. C., Montoya, J. M., Bascompte, J., Blüthgen, N. et al.. *Review : Ecological networks – beyond food webs*. J. Anim. Ecol. **78**, 253–269 (2009). doi : 10.1111/j.1365-2656.2008.01460.x.

[3] Toivonen, R., Onnela, J.-P., Saramäki, J., Hyvönen, J. et Kaski, K. *A model for social networks*. Physica A **371**, 851–860 (2006). doi :10.1016/j.physa.2006.03.050.

[4] Pastor-Satorras, R., Castellano, C., Van Mieghem, P. et Vespignani, A. *Epidemic processes in complex networks*. Rev. Mod. Phys. **87**, 925–979 (2015). doi :10.1103/RevModPhys.87.925.

[5] Bassett, D. S. et Sporns, O. *Network neuroscience*. Nat. Neurosci. **20**, 353–364 (2017). doi :10.1038/nn.4502.

[6] Pringle, R. M. et Hutchinson, M. C. *Resolving Food-Web Structure*. Annu. Rev. Ecol. Evol. Syst **51**, 55–80 (2020). doi :10.1146/annurev-ecolsys-110218-024908.

[7] Kolaczyk, E. D. *Statistical Analysis of Network Data : Methods and Models*. Springer Series in Statistics. Springer-Verlag (2009). ISBN 978-0-387-88145-4. doi :10.1007/978-0-387-88146-1.

[8] Wang, P., Xu, B., Wu, Y. et Zhou, X. *Link prediction in social networks : The state-of-the-art*. Sci. China Inf. Sci. **58**, 1–38 (2015). doi :10.1007/s11432-014-5237-y.

[9] Peixoto, T. P. *Reconstructing Networks with Unknown and Heterogeneous Errors*. Phys. Rev. X **8**, 041011 (2018). doi :10.1103/PhysRevX.8.041011.

[10] Zhou, H., Du, W., Xu, S. et Xin, Q. *An Empirical Study of Network Topology Inference*. Dans *Computer and Information Science 2011*, pp. 213–225 (2011). doi : 10.1007/978-3-642-21378-6_17.

[11] Ni, J., Xie, H., Tatikonda, S. et Yang, Y. R. *E cient and Dynamic Routing Topology Inference From End-to-End Measurements*. IEEE/ACM Trans. Netw. **18**, 123–135 (2010). doi :10.1109/TNET.2009.2022538.

[12] Brugere, I., Gallagher, B. et Berger-Wolf, T. Y. *Network Structure Inference, A Survey : Motivations, Methods, and Applications*. ACM Comput. Surv. **51**, 24 :1–24 :39 (2018). doi :10.1145/3154524.

[13] Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. et Geurts, P. *Inferring Regulatory Networks from Expression Data Using Tree-Based Methods*. PLOS ONE **5**, e12776 (2010). doi : 10.1371/journal.pone.0012776.

[14] Specht, A. T. et Li, J. *LEAP : Constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering*. Bioinformatics **33**, 764–766 (2017). doi :10.1093/bioinformatics/btw729.

[15] Matsumoto, H., Kiryu, H., Furusawa, C., Ko, M. S. H. et al.. *SCODE : An e cient regulatory network inference algorithm from single-cell RNA-Seq during di erentiation*. Bioinformatics **33**, 2314–2321 (2017). doi :10.1093/bioinformatics/btx194.

[16] Jansen, R., Yu, H., Greenbaum, D., Kluger, Y. et al.. *A Bayesian networks approach for predicting protein-protein interactions from genomic data*. Science **302**, 449–453 (2003). doi :10.1126/science.1087361.

[17] Butts, C. T. *Network inference, error, and informant (in)accuracy : A Bayesian approach*. Soc. Networks **25**, 103–140 (2003). doi :10.1016/S0378-8733(02)00038-2.

[18] Korhonen, O., Zanin, M. et Papo, D. *Principles and open questions in functional brain network reconstruction*. Human Brain Mapping. **42**, 3680–3711 (2021). doi :10.1002/hbm.25462.

[19] Runge, J. *Causal network reconstruction from time series : From theoretical assumptions to practical estimation*. Chaos **28**, 075310 (2018). doi :10.1063/1.5025050.

[20] Newman, M. E. J. *Network structure from rich but noisy data*. Nat. Phys. **14**, 542–545 (2018). doi :10.1038/s41567-018-0076-1.

[21] Kramer, M. A., Eden, U. T., Cash, S. S. et Kolaczyk, E. D. *Network inference with confidence from multivariate time series*. Phys. Rev. E **79**, 061916 (2009). doi :10.1103/PhysRevE.79.061916.

[22] Young, J.-G., Cantwell, G. T. et Newman, M. E. J. *Bayesian inference of network structure from unreliable data*. J. Complex Netw. **8**, cnaa046 (2021). doi :10.1093/comnet/cnaa046.

[23] Battiston, F., Cencetti, G., Iacopini, I., Latora, V. et al.. *Networks beyond pairwise interactions : Structure and dynamics*. Phys. Rep. **874**, 1–92 (2020). doi :10.1016/j.physrep.2020.05.004.

[24] Yu, S., Yang, H., Nakahara, H., Santos, G. S. et al.. *Higher-Order Interactions Characterized in Cortical Activity*. J. Neurosci. **31**, 17514–17526 (2011). doi :10.1523/JNEUROSCI.3127-11.2011.

[25] Battiston, F., Amico, E., Barrat, A., Bianconi, G. et al.. *The physics of higher-order interactions in complex systems*. Nat. Phys. **17**, 1093–1098 (2021). doi :10.1038/s41567-021-01371-4.

[26] Bairey, E., Kelsic, E. D. et Kishony, R. *High-order species interactions shape ecosystem diversity*. Nat. Commun. **7**, 12285 (2016). doi :10.1038/ncomms12285.

[27] Burgio, G., Matamalas, J. T., Gómez, S. et Arenas, A. *Evolution of Cooperation in the Presence of Higher-Order Interactions : From Networks to Hypergraphs*. Entropy **22**, 744 (2020). doi :10.3390/e22070744.

[28] Roy-Pomerleau, X. *Inférence d'interactions d'ordre Supérieur et de Complexes Simpliciaux à Partir de Données de Présence/Absence*. Mémoire de maîtrise, Université Laval (2020).

[29] Young, J.-G., Petri, G. et Peixoto, T. P. *Hypergraph reconstruction from network data*. Commun. Phys. **4**, 1–11 (2021). doi :10.1038/s42005-021-00637-w.

[30] Santoro, A., Battiston, F., Petri, G. et Amico, E. *Unveiling the higher-order organization of multivariate time series* (2022). doi :10.48550/arXiv.2203.10702.

[31] Musciotto, F., Battiston, F. et Mantegna, R. N. *Detecting informative higher-order interactions in statistically validated hypergraphs*. Commun. Phys. **4**, 1–9 (2021). doi :10.1038/s42005-021-00710-4.

[32] Yen, T.-C. et Larremore, D. B. *Community detection in bipartite networks with stochastic block models*. Phys. Rev. E **102**, 032309 (2020). doi :10.1103/PhysRevE.102.032309.

[33] Suwan, S., Lee, D. S., Tang, R., Sussman, D. L. et al.. *Empirical Bayes estimation for the stochastic blockmodel*. Electron. J. Stat. **10**, 761–782 (2016). doi :10.1214/16-EJS1115.

[34] Snijders, T. A. et Nowicki, K. *Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure*. J. Classif. **14**, 75–100 (1997). doi :10.1007/s003579900004.

[35] Peixoto, T. P. *Network Reconstruction and Community Detection from Dynamics*. Phys. Rev. Lett. **123**, 128301 (2019). doi :10.1103/PhysRevLett.123.128301.

[36] van der Pas, S. L. et van der Vaart, A. W. *Bayesian Community Detection*. Bayesian Anal. **13**, 767–796 (2018). doi :10.1214/17-BA1078.

[37] Nowicki, K. et Snijders, T. A. B. *Estimation and Prediction for Stochastic Blockstructures*. J. Am. Stat. Assoc. **96**, 1077–1087 (2001). doi :10.1198/016214501753208735.

[38] Hofman, J. M. et Wiggins, C. H. *Bayesian Approach to Network Modularity*. Phys. Rev. Lett. **100**, 258701 (2008). doi :10.1103/PhysRevLett.100.258701.

[39] Peixoto, T. P. *Bayesian Stochastic Blockmodeling*. Dans P. Doreian, V. Batagelj et A. Ferligoj (éditeurs), *Advances in Network Clustering and Blockmodeling*, pp. 289–332. John Wiley & Sons, Ltd (2019). doi :10.1002/9781119483298.ch11.

[40] Peixoto, T. P. *Nonparametric Bayesian inference of the microcanonical stochastic block model*. Phys. Rev. E **95**, 012317 (2017). doi :10.1103/PhysRevE.95.012317.

[41] Birlutiu, A., d'Alché-Buc, F. et Heskes, T. *A Bayesian Framework for Combining Protein and Network Topology Information for Predicting Protein-Protein Interactions*. IEEE/ACM Trans. Comput. Biol. Bioinform. **12**, 538–550 (2015 May-Jun). doi : 10.1109/TCBB.2014.2359441.

[42] Asthana, S., King, O. D., Gibbons, F. D. et Roth, F. P. *Predicting Protein Complex Membership Using Probabilistic Network Reliability*. Genome Res. **14**, 1170–1175 (2004). doi :10.1101/gr.2203804.

[43] Farine, D. R. et Strandburg-Peshkin, A. *Estimating uncertainty and reliability of social network data using Bayesian inference*. R. Soc. Open Sci. **2**, 150367 (2015). doi :10.1098/rsos.150367.

[44] Young, J.-G., Valdovinos, F. S. et Newman, M. E. J. *Reconstruction of plant–pollinator networks from observational data*. Nat. Commun. **12**, 3911 (2021). doi :10.1038/s41467-021-24149-x.

[45] Ho□, P. D., Raftery, A. E. et Handcock, M. S. *Latent Space Approaches to Social Network Analysis*. J. Am. Stat. Assoc. **97**, 1090–1098 (2002). doi :10.1198/016214502388618906.

[46] Yin, J., Ho, Q. et Xing, E. P. *A Scalable Approach to Probabilistic Latent Space Inference of Large-Scale Networks*. Dans *Advances in Neural Information Processing Systems*, volume 26 (2013).

[47] Mitchell, M. *Complexity : A Guided Tour*. OUP USA (2011). ISBN 978-0-19-979810-0.

[48] Anderson, P. W. *More Is Di erent*. Science **177**, 393–396 (1972). doi :10.1126/science.177.4047.393.

[49] Crane, H. *Probabilistic Foundations of Statistical Network Analysis*. Chapman and Hall/CRC (2018). ISBN 978-1-138-63015-4.

[50] Klenke, A. *Probability Theory*. Springer London (2014). ISBN 978-1-4471-5361-0.

[51] Kolmogorov, A. N. *Foundations of the Theory of Probability*. Martino Fine Books (2013). ISBN 978-1-61427-514-5.

[52] Billingsley, P. *Probability and Measure*. Wiley (1995). ISBN 978-0-471-00710-4.

[53] Li, M., Liu, R.-R., Lü, L., Hu, M.-B. et al.. *Percolation on complex networks : Theory and application*. Phys. Rep. **907**, 1–68 (2021). doi :10.1016/j.physrep.2020.12.003.

[54] Röttjers, L., Vandeputte, D., Raes, J. et Faust, K. *Null-model-based network comparison reveals core associations*. ISME Commun. **1**, 1–8 (2021). doi :10.1038/s43705-021-00036-w.

[55] Drobyshevskiy, M. et Turdakov, D. *Random Graph Modeling : A Survey of the Concepts*. ACM Comput. Surv. **52**, 131 :1–131 :36 (2019). doi :10.1145/3369782.

[56] Goldenberg, A., Zheng, A. X., Fienberg, S. E. et Airoldi, E. M. *A Survey of Statistical Network Models*. Found. Trends Mach. Learn. **2**, 129–233 (2010). doi :10.1561/2200000005.

[57] Gilbert, E. N. *Random Graphs*. Ann. Math. Stat. **30**, 1141–1144 (1959). doi :10.1214/aoms/1177706098.

[58] Paul, S., Milenkovic, O. et Chen, Y. *Higher-Order Spectral Clustering under Superimposed Stochastic Block Model* (2018). doi :10.48550/arXiv.1812.06515.

[59] Miller, J. C. *Percolation and epidemics in random clustered networks*. Phys. Rev. E **80**, 020901 (2009). doi :10.1103/PhysRevE.80.020901.

[60] Stasi, D., Sadeghi, K., Rinaldo, A., Petrović, S. et Fienberg, S. E. *β models for random hypergraphs with a given degree sequence* (2014). doi :10.48550/arXiv.1407.1004.

[61] Popper, K. *The Logic of Scientific Discovery*. Routledge, 2e édition (2002). ISBN 978-0-415-27844-7.

[62] Jaynes, E. T. *Information Theory and Statistical Mechanics*. Phys. Rev. **106**, 620–630 (1957). doi :10.1103/PhysRev.106.620.

[63] Fink, D. *A Compendium of Conjugate Priors*. Technical, Montana State University (1997).

[64] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B. et al.. *Bayesian Data Analysis*. CRC Press, 3e édition (2013). ISBN 978-1-4398-4095-5.

[65] Gentle, J. E. *Random Number Generation and Monte Carlo Methods*. Springer-Verlag New York Inc., 2e édition (2004). ISBN 978-0-387-00178-4.

[66] Newman, M. E. J. *Computational Physics*. CreateSpace Independent Publishing Platform (2012). ISBN 978-1-4801-4551-1.

[67] Brooks, S., Gelman, A., Jones, G. et Meng, X.-L. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC (2011). ISBN 978-1-4200-7941-8.

[68] Levin, D. A. et Peres, Y. *Markov Chains and Mixing Times (Second Edition)*. Providence : American Mathematical Society (2017). ISBN 978-1-4704-2962-1.

[69] Robert, C. P. et Casella, G. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer (2004). ISBN 978-1-4419-1939-7 978-1-4757-4145-2. doi :10.1007/978-1-4757-4145-2.

[70] Hastings, W. K. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*. Biometrika **57**, 97–109 (1970). doi :10.2307/2334940.

[71] Tjelmeland, H. et Hegstad, B. K. *Mode Jumping Proposals in MCMC*. Scand. J. Stat. **28**, 205–223 (2001). doi :10.1111/1467-9469.00232.

[72] Pompe, E., Holmes, C. et Łatuszyński, K. *A framework for adaptive MCMC targeting multimodal distributions*. Ann. Stat. **48**, 2930–2952 (2020). doi :10.1214/19-AOS1916.

[73] Gelman, A. et Rubin, D. B. *Inference from Iterative Simulation Using Multiple Sequences*. Stat. Sci. **7**, 457–472 (1992). doi :10.1214/ss/1177011136.

[74] He, B., De Sa, C., Mitliagkas, I. et Ré, C. *Scan Order in Gibbs Sampling : Models in Which it Matters and Bounds on How Much*. Dans *Advances in Neural Information Processing Systems*, volume 29, p. 6589 (2016).

[75] Basilio, A. M., Medan, D., Torretta, J. P. et Bartoloni, N. J. *A year-long plant-pollinator network*. Austral Ecol. **31**, 975–983 (2006). doi :10.1111/j.1442-9993.2006.01666.x.

[76] Lei, C. et Ruan, J. *A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity*. Bioinformatics **29**, 355–364 (2013). doi : 10.1093/bioinformatics/bts688.

[77] Cai, L., Wei, X., Wang, J., Yu, H. et al.. *Reconstruction of functional brain network in Alzheimer's disease via cross-frequency phase synchronization*. Neurocomputing **314**, 490–500 (2018). doi :10.1016/j.neucom.2018.07.019.

[78] Hlaváčková-Schindler, K., Paluš, M., Vejmelka, M. et Bhattacharya, J. *Causality detection based on information-theoretic approaches in time series analysis*. Phys. Rep. **441**, 1–46 (2007). doi :10.1016/j.physrep.2006.12.004.

[79] Qiao, L., Zhang, H., Kim, M., Teng, S. et al.. *Estimating functional brain networks by incorporating a modularity prior*. NeuroImage **141**, 399–407 (2016). doi :10.1016/j.neuroimage.2016.07.058.

[80] Bick, C., Gross, E., Harrington, H. A. et Schaub, M. T. *What are higher-order networks ?* (2022). doi :10.48550/arXiv.2104.11329.

[81] Mayfield, M. M. et Stou□er, D. B. *Higher-order interactions capture unexplained complexity in diverse communities*. Nat. Ecol. Evol. **1**, 1–7 (2017). doi :10.1038/s41559-016-0062.

[82] Grilli, J., Barabás, G., Michalska-Smith, M. J. et Allesina, S. *Higher-order interactions stabilize dynamics in competitive network models*. Nature **548**, 210–213 (2017). doi :10.1038/nature23273.

[83] Milojević, S. *Principles of scientific research team formation and evolution*. Proc. Natl. Acad. Sci. U.S.A. **111**, 3984–3989 (2014). doi :10.1073/pnas.1309723111.

[84] Stam, C. J. *Functional connectivity patterns of human magnetoencephalographic recordings : A 'small-world' network ?* Neurosci. Lett. **355**, 25–28 (2004). doi :10.1016/j.neulet.2003.10.063.

[85] Zachary, W. W. *An Information Flow Model for Conflict and Fission in Small Groups*. J. Anthropol. Res. **33**, 452–473 (1977).

[86] Decker, S. H., Kohfeld, C., Rosenfeld, R. et Sprague, J. D. *The St. Louis Homicide Project : Local Responses to a National Problem*. University of Missouri-St. Louis (1991).

[87] Rocha, L. E. C., Liljeros, F. et Holme, P. *Simulated Epidemics in an Empirical Spatio-temporal Network of 50,185 Sexual Contacts*. PLOS Comput. Biol. **7**, e1001109 (2011). doi :10.1371/journal.pcbi.1001109.

[88] Kato, M., Kakutani, T., Inoue, T. et Itino, T. *Insect-flower Relationship in the Primary Beech Forest of Ashu, Kyoto : An Overview of the Flowering Phenology and the Seasonal Pattern of Insect Visits*. Contr. Biol. Lab. Kyoto Univ. **27**, 309–376 (1990).

[89] Kunegis, J. *KONECT : The Koblenz network collection*. Dans *Proceedings of the 22nd International Conference on World Wide Web*, pp. 1343–1350 (2013). doi :10.1145/2487788.2488173.

[90] Gelman, A., Meng, X.-l. et Stern, H. *Posterior Predictive Assessment of Model Fitness Via Realized Discrepancies*. Stat. Sin. **6**, 733–760 (1996).

[91] Betancourt, M. *Identifying Bayesian Mixture Models* (2017).

[92] Ahrens, J. H. et Dieter, U. *Computer methods for sampling from gamma, beta, poisson and bionomial distributions*. Computing **12**, 223–246 (1974). doi :10.1007/BF02293108.

[93] Gallagher, R. J., Young, J.-G. et Welles, B. F. *A clarified typology of core-periphery structure in networks*. Sci. Adv. **7**, eabc9800 (2021). doi :10.1126/sciadv.abc9800.

[94] Knuth, D. E. *Big Omicron and big Omega and big Theta*. ACM SIGACT News **8**, 18–24 (1976). doi :10.1145/1008328.1008329.

[95] Stanley, R. P. *Enumerative Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2e édition (2011). ISBN 978-1-107-01542-5. doi : 10.1017/CBO9781139058520.