Analyse des réseaux d'ordre supérieur consruits à partir de séquences historio-géographiques

Julie Queiros

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

THÈSE DE DOCTORAT DE

NANTES UNIVERSITÉ

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *« Informatique »*

Par
Julie QUEIROS

« Analyse de réseaux d'ordre supérieur construits à partir de séquences historio-géographiques »

Thèse présentée et soutenue à Nantes, le 17/10/2024
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N) UMR 6004, Nantes Université

Rapporteurs avant soutenance :

Cécile BOTHOREL	Professeur, IMT Atlantique Brest
Jean-Loup GUILLAUME	Professeur, Université de la Rochelle

Composition du Jury :

Président :	Guy MÉLANÇON	Professeur, Université de Bordeaux
Examinateurs :	Cécile BOTHOREL	Professeur, IMT Atlantique Brest
	Jean-Loup GUILLAUME	Professeur, Université de la Rochelle
	Claire LAGESSE	Maître de conférences, Université de Franche-Comté
Dir. de thèse :	Marc GELGON	Professeur, Polytech' Nantes
Co-enc. de thèse :	François QUEYROI	Chargé de Recherche, CNRS

SOMMAIRE

TABLE DES FIGURES

Liste des tableaux

Liste des Algorithmes

Chapitre 1

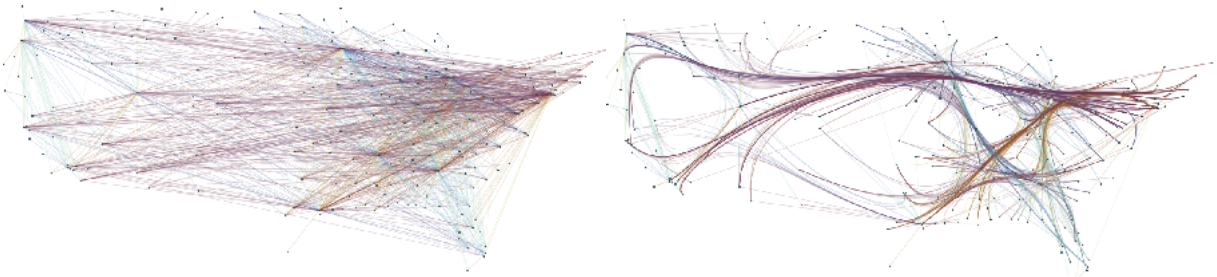
Introduction

1.1 Analyse de réseaux

Les *systèmes complexes* sont des abstractions de phénomènes réels où des entités interagissent entre elles, directement ou indirectement. Ces relations ont une influence sur la structure et la dynamique du système et définissent des *réseaux* (parfois aussi qualifiés de *complexes*). De ce fait, il est naturel d'aborder ces problèmes avec les outils de la théorie des graphes. Les graphes modélisant ces réseaux sont caractérisés par leurs propriétés topologiques non triviales, telles que la présence de nœuds hautement connectés, d'une structure communautaire ou l'existence de corrélations entre les nœuds, *etc.* Notons d'ores et déjà que bien que la distinction entre le système modélisé (réseau) et l'objet mathématique (graphe) est importante, nous utiliserons ici les deux termes de façon interchangeable.

La compréhension de la structure et de la dynamique des réseaux est cruciale pour de nombreuses applications, comme la propagation des maladies ou la diffusion de l'information. L'étude des réseaux complexes est un domaine interdisciplinaire qui s'appuie sur des idées et des méthodes issues de nombreux autres domaines tels que la physique, les mathématiques, l'informatique et les sciences sociales. L'*analyse de réseaux* ou *fouille de graphes* (*graph mining*) correspond, dans ce cadre, à la mobilisation d'un ensemble d'outils pour réaliser cette étude. Une caractéristique primordiale de l'analyse de réseaux est la prise en compte des relations indirectes, lorsqu'une entité est indirectement connectée à une autre par le biais d'un ou plusieurs entités tierces. L'extraction d'information du réseau ne se limite ainsi pas à la prise en compte des relations directes.

Dans la majorité des cas, les modèles de réseaux de la littérature nous permettent uniquement de modéliser des relations entre deux entités, ou *dyadiques*. Il est toutefois possible, pour un système complexe donné, de construire et d'étudier des réseaux différents. Par exemple, en prenant un système de transport aérien (voir figure 1.1), on peut s'intéresser à la structure du réseau défini par les liaisons directes (c'est-à-dire tous les vols possibles entre deux aéroports) mais aussi aux flux de déplacement de voyageurs utilisant cette structure. Dans ce travail, nous nous intéressons à la représentation des flux



(a) Visualisation nœuds-liens d'un réseau aérien

(b) Visualisation améliorée par le regroupement d'arêtes en faisceaux (*graph bundling*)

FIGURE 1.1 – Visualisation d'un réseau représentant les itinéraires d'avions aux États-Unis tiré de [81]. Un nœud correspond à un aéroport et une arête représente le flux d'avions entre les deux aéroports. L'analyse de réseau nous permet de faire sens des dynamiques d'un système, ici une technique de dessin de graphe (figure 1.1b) permet de regrouper les arêtes pour visualiser plus clairement les itinéraires importants.

(pour ce type de système et d'autres). Les relations indirectes sont également pertinentes dans ce cadre. En effet, si on observe un message entre une entité `A` et `B` puis entre `B` et `C`, alors on peut inférer le transfert d'une information entre `A` et `C` même en l'absence de relation directe. Nous verrons que, même en se limitant à un même ensemble de flux comme les *données séquentielles* détaillé dans la section suivante, il y aura plusieurs façons de construire des réseaux à partir de ces données qui pourront influer les analyses faites sur le système sous-jacent. C'est en partie cette multiplicité des représentations qui nous intéresse dans cette thèse.

1.2 Données séquentielles

Dans le cadre de ce travail, nous nous concentrons sur des trajectoires (l'historique) d'un agent dans un espace (géographique). Dans notre cas, on s'intéresse à la suite d'«états» visités par cet agent. Prenons l'exemple de données maritimes de la figure 1.2, elles correspondent aux trajectoires de navires (les agents) dans un système de transport maritime. L'approche par les réseaux suppose l'agrégation des trajectoires individuelles des agents en flux ; nous mettons ainsi de côté les données de l'agent/navire (le nom, le type, le tonnage, *etc.*) et nous concentrons uniquement sur la succession de ports qui constituent la trajectoire du navire. D'autres variables telles que le temps de trajet ou le temps passé dans un port sont également laissées de côté pour ne conserver que des *séquences discrètes* sur un ensemble d'*états* correspondant ici aux ports.

Les séquences discrètes sont un type de données courant en informatique, notamment dans les domaines de la fouille de motifs séquentiels [2] ou de l'encodage de séquences

Navire	Départ	Date Dep.	Arrivée	Date Arr.
Unicorn	Tokyo	13-03-2009	Singapour	15-03-2009
Unicorn	Singapour	17-03-2009	Le Havre	02-04-2009
Argos	Tokyo	20-03-2009	Singapour	28-03-2009
Argos	Singapour	31-03-2009	Le Havre	27-04-2009
Pequod	Tokyo	01-03-2009	Singapour	22-03-2009
Pequod	Singapour	31-03-2009	Los Angeles	18-04-2009
Noah's Arc	Shanghai	11-03-2009	Singapour	28-03-2009
Noah's Arc	Singapour	09-05-2009	Le Havre	17-05-2009
Nautilus	Shanghai	17-03-2009	Singapour	08-04-2009
Nautilus	Singapour	09-04-2009	Los Angeles	02-05-2009
...	...	...	...	...

(a) Données brutes

```
Tokyo;Singapour;Le Havre
Tokyo;Singapour;Le Havre
Tokyo;Singapour;Los Angeles
Shanghai;Singapour;Le Havre
Shanghai;Singapour;Los Angeles
Shanghai;Singapour;Los Angeles
...
```

(b) Séquences

FIGURE 1.2 – Les données brutes (figure 1.2a) sont transformées en séquences discrètes (figure 1.2b) en ne conservant que la suite de transition d'un port à un autre pour chaque navire. Par exemple, le navire « Unicorn » part de Tokyo pour s'arrêter à Singapour et, à une date ultérieure, repart de Singapour pour aller au Havre.

de symboles [77] renvoyant à des problèmes comme celui de la compression de texte [6]. Nous voulons, dans notre cas, construire des réseaux afin d'offrir la possibilité d'analyser et de fouiller les relations entre états pour décrire les dynamiques et comportements du système. Des applications possibles pour les données de navigation sont l'analyse historique de l'évolution du commerce maritime [15] ou encore la détection et prévention du *biofouling* [71]. Le *biofouling*, ou « encrassement biologique », est la couche d'organismes qui se forme sur la coque d'un navire. Il entraîne des risques d'invasion d'espèces exotiques dans des nouveaux milieux. Pour prédire et modéliser ces changements, il est important de tenir compte des relations indirectes entre ports et pas seulement de tenir compte des paires de ports ayant échangé beaucoup de navires.

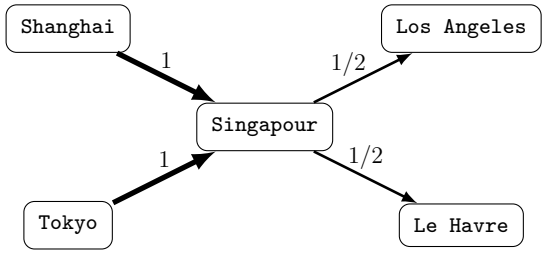
1.3 Propriété de Markov

Pour ces différentes applications, il est possible de modéliser les flux correspondant aux données séquentielles par des graphes. Une approche classique consiste à ne tenir compte que des transitions directes entre les états du système ; le graphe est construit en agrégeant les occurrences entre paires d'états dans le jeu de données (voir figure 1.3). On obtient un graphe orienté et stochastique où le poids des arcs correspond à la probabilité de transition entre les deux états, c'est-à-dire le nombre de fois où on compte le trajet entre les deux ports sur le nombre total de trajets qui partent du premier.

Comme noté précédemment, les relations indirectes sont centrales en analyse de réseaux. Or, on peut constater que ce modèle de réseau entraîne une mauvaise représentation de celles-ci. Dans les données séquentielles (fig. 1.3a), les navires venant de `Tokyo` avant de faire escale à `Singapour` ont davantage de chance d'aller au port du `Havre` (deux chances sur trois). En suivant uniquement le graphe de la figure 1.3b, la probabilité d'un navire quittant `Singapour` en venant de `Tokyo` d'aller au `Havre` ou `Los Angeles` est la

```
Tokyo;Singapour;Le Havre
Tokyo;Singapour;Le Havre
Tokyo;Singapour;Los Angeles
Shanghai;Singapour;Le Havre
Shanghai;Singapour;Los Angeles
Shanghai;Singapour;Los Angeles
...
```

(a) Séquences



(b) Graphe stochastique

FIGURE 1.3 – Exemple de construction d'un réseau (graphe stochastique) ne tenant compte que des transitions directes.

même (une chance sur deux). En représentant les données séquentielles avec ce modèle de réseau, on perd une partie de l'information sur les relations existantes dans ce système. En effet, on suppose implicitement que celles-ci respectent la « propriété de Markov », c'est-à-dire qu'elle sont issues d'un processus de Markov traditionnel où l'état futur d'un agent dépend uniquement de son état actuel, et non de ses états précédents. Le caractère « markovien » des trajectoires réelles a été remis en cause dans plusieurs travaux, tels que ceux de Chierichetti *et al.* [19] sur les trajectoires d'utilisateurs sur le Web.
La perte d'information liée à l'utilisation de réseaux classiques peut avoir un impact sur les résultats obtenus par les outils de fouille appliqués aux réseaux, notamment les méthodes basées sur les marches aléatoires qui sont très répandues. En partant de ce constat, est-il possible de modéliser la dynamique d'un système sous forme de graphe tout en conservant ce que nous nommons les *dépendances séquentielles*, *i.e* les situations où la propriété de Markov ne serait pas respectée (ex. la suite de ports `Shangai;Singapour` ou `Tokyo;Singapour`) ?

1.4 Réseaux d'ordre supérieur

Pour représenter ces dépendances séquentielles, les chercheurs ont développé des modèles appelés *réseaux d'ordre supérieur*, où l'« ordre » fait ici référence au nombre d'états pris en compte pour calculer les transitions, le réseau « classique » de la figure 1.4a étant ainsi dit « d'ordre 1 ». Ce domaine de recherche récent s'inscrit dans la lignée des travaux en analyse de réseaux cherchant à dépasser les relations *dyadiques* [26]. Les premiers travaux dans ce domaine remontent à 2014 avec Rosvall *et al.* [70] suivis par Scholtes *et al.* [76, 75] ou encore Xu *et al.* [84, 71] qui ont chacun proposé des modèles de réseaux d'ordre supérieur et des méthodes d'analyse différents. Nous nous basons en grande partie sur ces travaux dans cette thèse. Bien que la recherche sur ces modèles soit relativement nouvelle dans le contexte de l'analyse de réseaux, il faut toutefois noter que la conception de modèles markoviens d'ordre supérieur, dans le cadre de la prédiction des trajectoires notamment, est loin d'être récente [11, 78].

(a) Réseau d'ordre 1 (b) Réseau d'ordre 2

FIGURE 1.4 – Plusieurs modèles de réseaux possibles

Ces réseaux d'ordre supérieur permettent la prise en compte des dépendances séquentielles en utilisant des « nœuds-mémoires ». La figure 1.4b est un exemple de réseau d'ordre 2. Ces nœuds sont à lire comme suit : `Tokyo;Singapour` correspond à un navire étant actuellement à `Singapour` mais venant de `Tokyo`. Les probabilités de transitions pour `Tokyo;Singapour` sont donc calculées en ne regardant que les destinations des navires ayant suivi cette trajectoire. Nous ne rentrerons pas ici dans les détails de construction ; les définitions seront données dans le chapitre suivant. On peut d'ores et déjà noter que, dans le cas général, on pourrait choisir un ordre plus grand, voire choisir d'inclure certains nœuds-mémoire et pas d'autres. Pour un même jeu de données séquentielles, différents réseaux sont donc possibles.

Les réseaux d'ordre supérieur permettent de préserver deux types de relations indirectes entre états :

1. Deux états sont présents dans une même séquence (*i.e.* un navire a visité un port avant un autre ; des marchandises ont pu transiter entre les deux) ;
2. Deux états n'apparaissent pas dans une séquence différente mais une relation peut être obtenue en passant d'une séquence à l'autre (*i.e.* un navire achemine des marchandises vers un port intermédiaire, un autre termine la livraison).

Les réseaux d'ordre 1 ne considèrent que des relations du second type car on ne compte que les transitions directes : que ce soit un même navire faisant `Tokyo;Singapour;Le Havre` ou deux navires, le premier faisant `Tokyo;Singapour` et le second `Singapour;Le Havre`, la même information est ajoutée au réseau. Il est possible de combiner les deux types de relations avec un ordre supérieur. En ajoutant les nœuds-mémoires, on sort du cadre des relations dyadiques, en prenant en compte les relations entre plus que deux états. Par ailleurs, dans le cadre que nous proposons, les réseaux d'ordre supérieur conservent l'information contenue dans les réseaux d'ordre 1 (*i.e.* le graphe 1.4b conserve le nœud `Singapour` regroupant les navires faisant escale à Singapour peu importe les ports précédemment visités).

1.5 Problématiques traitées dans cette thèse

Comme mentionné plus haut, la construction de réseau n'est pas une fin en soi mais fait partie d'un processus de recherche plus général. En effet, une fois une modélisation en réseau établie, l'expert pourra être amené à fouiller le réseau en utilisant des méthodes visuelles et interactives [36] ou en calculant des métriques [24, 9]. Nous nous concentrons sur le calcul de métriques pour évaluer la *centralité* ou faire du *clustering*. Sans perte de généralité, nous considérons que l'analyse de réseaux à partir de données séquentielles suit les étapes suivantes qui sont, nous allons le voir, généralement posées comme indépendantes dans la littérature :

1. Recueil des données brutes (*e.g.* figure 1.2a)
2. Transformations en données séquentielles (*e.g.* figure 1.2b)
3. Construction d'un réseau (éventuellement d'ordre supérieur) (*e.g.* figure 1.4)
4. Applications d'algorithmes pour le calcul de métriques ou le clustering dans l'espace des nœuds
5. Projections des résultats sur l'espace des états.

Les deux problématiques traitées dans cette thèse concernent principalement les étapes 3, 4 et 5. L'étape 3 correspond à la transformation des séquences en modèles de réseaux d'ordre supérieur. Les enjeux liés à la définition de ces nouveaux modèles de réseaux sont bien résumés par Lambiotte *et al.* [52] :

> Un important défi épistémologique est de trouver de nouvelles manières d'inférer des modèles optimaux de systèmes complexes à partir de données [séquentielles]. En se basant sur le rasoir d'Ockham, de tels modèles devraient ainsi être les plus parcimonieux possibles. En effet, nous voulons limiter nos hypothèses pour permettre d'identifier des règles généralisables au-delà du système étudié. Mais un bon modèle doit être assez complexe pour pouvoir expliquer les trajectoires observées dans des systèmes réels, c'est sur ce point que la science des réseaux standard est limitée.
> (...)
> Trouver ces modèles optimaux (...) correspond alors à un problème de *machine learning* où les modèles de réseaux standards sont simplement une des nombreuses possibilités pour expliquer les trajectoires observées.
> (Lambiotte *et al.* [52])

Ce défi est relativement singulier du point de vue de l'analyse de réseaux car la création du réseau avec ses différentes caractéristiques est rarement vue comme un problème de modélisation statistique [1]. Une **première problématique** traitée dans ce manuscrit est donc la recherche d'un « modèle optimal » ou, du moins, d'une méthode permettant de parcourir des solutions intéressantes offrant un bon compromis entre la « généralité »

1. Ce n'est pas vrai pour des thématiques telles que la *sparcification* de réseaux avec les méthodes d'extraction de *network backbone* notamment [33].

et la « précision ».

Ce défi est également singulier du point de vue du *machine learning* car il s'agit bien de construire un modèle n'ayant pas pour finalité la prédiction mais qui puisse être fouillé pour obtenir des informations sur les états. Malgré les nœuds-mémoires ajoutés, les réseaux d'ordre supérieur restent des graphes de même nature que les réseaux d'ordre 1 *i.e.* en omettant les labels, les graphes de la figure 1.3 sont des graphes dirigés et pondérés « classiques ». Des algorithmes de fouille pourraient ainsi y être directement appliqués. Nous sommes toutefois intéressés par des informations sur les états et non les nœuds. Dans les réseaux « classiques », une bijection existe entre les nœuds et les états alors que, dans le cas de réseau d'ordre supérieur, chaque état peut correspondre à plusieurs nœuds du réseau. Dans la littérature, cette différence est peu discutée et l'étape 5 est considérée comme relativement évidente peu importe la métrique considérée [84]. **Une seconde problématique** traitée dans cette thèse est le questionnement de l'indépendance entre les étapes 4 et 5 de la chaîne d'analyse, en particulier dans le cas des analyses de centralité ou du *clustering* de graphes. Nous verrons qu'il n'est pas opportun de directement utiliser des algorithmes de fouille de graphes mais que des adaptations sont nécessaires.

1.6 Ce qui n'est pas traité dans cette thèse

Avant de détailler l'organisation de ce document et les travaux réalisés, il est important de positionner notre sujet par rapport à d'autres thématiques différentes mais qui emploient des termes proches : dans notre cas, « ordre supérieur » et l'« analyse de séquences ». Notons que des passerelles peuvent exister entre ces domaines et notre sujet mais elles ne seront pas explorées ici.

Tout d'abord, la notion d'« ordre supérieur » (*higher-order*) est employée pour désigner des concepts très différents. Eliassi-Rad *et al.* [26] définissent les « réseaux d'ordre supérieur » comme tous les réseaux conçus pour capturer davantage que les relations *dyadiques* (entre deux entités). Les relations de co-autorat [5] sont un exemple notable où la transformation de ces relations en graphe simple représente une perte d'information. Par exemple, un article de trois auteur · rices ou trois articles entre chaque paire d'auteur · rices ajouteront dans les deux cas une 3-clique au graphe. Un *hypergraphe* peut être utilisé pour encoder ces relations sans perte d'information. Les dépendances séquentielles sont un autre exemple de relations dépassant les relations dyadiques. Afin d'éviter les confusions lorsque nous évoquons ces autres concepts, nous utilisons le terme « réseaux d'ordre markovien supérieur » pour parler des réseaux construits à partir de séquences qui nous intéressent ici.

L'utilisation de données séquentielles dans les réseaux d'ordre supérieur doit également être distinguée de l'« analyse de séquences » (*sequence analysis*). Dans cette dernière, les individus sont définis par une séquence d'états. Une application possible consiste à utiliser des mesures de distances entre ces séquences [79] pour en déduire des classes d'individus. Ces techniques sont notamment utilisées dans le cadre de l'étude des évolutions de composition socio-professionnelle de quartiers [67] ou encore des « parcours de vie » [68]. L'analyse de séquences incorpore des dimensions temporelles, à savoir la durée passée par l'individu dans un état donné ou l'instant du passage dans cet état (avec une ligne temporelle souvent discrétisée). Dans un ouvrage sur le présent et le futur de l'analyse de séquences, Cornwell [23] présente la construction et l'analyse de réseaux temporels formés à partir des séquences d'état. Il suggère également qu'il serait intéressant d'inclure des « transitions d'ordre supérieur » dans ces réseaux. Ainsi, malgré la différence des dimensions prises en compte dans les séquences, nous pensons qu'un rapprochement entre l'analyse de réseaux d'ordre séquentiel supérieur et l'analyse de séquences est une perspective prometteuse.

1.7 Travaux réalisés et organisation du document

Nous détaillons ici la trame de ce document en faisant le lien avec les articles et productions logicielles effectués durant la thèse. Nous soulignons en particulier les différences apportées dans ce manuscrit par rapport aux articles déjà publiés. Une grande partie des outils et algorithmes présentés dans cette thèse sont implémentés dans un *package* Python nommé `HONyx` [65] basé sur `NetworkX` [34].

Le chapitre 2 s'attache à définir précisément les différents concepts mathématiques qui seront utilisés dans ce manuscrit. Nous invitons le lecteur à s'y référer notamment à travers une synthèse des notations (tableau 2.6). Nous faisons également une présentation des jeux de données utilisés tout au long de ce document ainsi que d'autres cas d'études traités dans la littérature.

Le chapitre 3 est centré sur la problématique de construction des réseaux d'ordre supérieur. Le nouveau modèle proposé MC-Von a fait l'objet d'une publication à la conférence *French Regional Conference on Complex Systems (FRCCS 2024 - Montpellier)* [66]. Nous développons ici la discussion sur les autres modèles de réseaux et les résultats expérimentaux. Le module Python développé, `HONyx`, permet le calcul des différents modèles discutés dans ce chapitre.

Dans le chapitre 4, nous nous intéressons à la deuxième problématique de fouille de réseaux d'ordre supérieur. On s'intéresse à l'évaluation de la *centralité* des états en généra-

lisant la mesure de PageRank aux réseaux d'ordre supérieur. Ce travail a fait l'objet d'une publication dans la conférence *International Conference on Complex Networks and their Applications (Complex Networks 2023 - Madrid)* [21]. Nous développons ici les différences méthodologiques avec la littérature existante et proposons une comparaison des résultats selon les différents modèles de réseaux présentés dans le chapitre 3. Le module Python `HONyx` permet le calcul de la centralité des états à partir d'un HON construit selon l'un des modèles.

Le chapitre 5 traite également de la deuxième problématique en se concentrant cette fois sur le *clustering* chevauchant des états à partir d'algorithmes de partitionnement appliqués aux réseaux d'ordre supérieur. Ces résultats furent publiés dans la revue *Network Science (Cambridge Press)* [64]. Notre contribution sur ce sujet ne consiste pas à fournir un algorithme « définitif » pour le clustering de réseau d'ordre supérieur mais plutôt à s'interroger sur l'influence du choix des modèles sur les méthodes existantes. Ce chapitre inclut une discussion plus détaillée sur les pistes futures sur ce sujet.

La conclusion de cette thèse (chapitre 6) contient une réflexion plus globale sur les hypothèses formulées dans ce chapitre. Nous nous interrogeons notamment sur la séparation posée entre la construction du modèle et la fouille de celui-ci (étapes 3-4-5 ci-dessus). Cette séparation pourrait en effet être remise en cause dans le cadre de tâches d'apprentissage supervisé sur les réseaux (*i.e.* avec les *Graph Neural Networks* ou GNN).

Chapitre 2

Définitions et Notations

Dans ce chapitre, nous allons nous attacher à fournir certaines définitions préliminaires qui permettront une meilleure compréhension du sujet. En particulier, nous donnons ici une définition formelle des réseaux d'ordre supérieur, des concepts qui y sont liés ainsi que certaines de leurs propriétés. Un récapitulatif des notations utilisées est disponible dans le tableau 2.6 en page 37. Ce tableau ne couvre pas l'ensemble des notations utilisées dans les chapitres suivants mais celles qui sont utilisées dans plus d'un chapitre. Certaines notations, en particulier pour les modèles d'ordre supérieur, seront introduites dans le chapitre 3.

2.1 Séquences

Les *séquences*, *trajectoires* ou *traces* forment la base de l'information dont nous disposons. Les notations ou concepts ci-dessous peuvent ainsi se retrouver dans la littérature concernant la prédiction de séquences ou la compression de texte [6].

On considère que l'on connaît l'ensemble des *états* possibles $\mathcal{A}$ dans le système étudié. Dans le cadre de textes, ces états correspondent à ce qu'on nomme un *alphabet*, c'est-à-dire, par exemple, l'ensemble des caractères de texte possibles.

Définition 2.1 (Séquences)**.** *Pour un ensemble d'état donné $\mathcal{A}$, s est appelée une* séquence *d'éléments de $\mathcal{A}$ et correspond à une suite finie de $\mathcal{A}$ i.e. $s = \sigma_1\sigma_2\ldots\sigma_m$.*
Un jeu de données *est constitué d'un multi-ensemble de séquences $\mathcal{S}$ sur les états de $\mathcal{A}$. Ainsi, une même séquence peut être observée plusieurs fois.*

EXEMPLE

Les séquences suivantes nous serviront de fil rouge pour le reste du chapitre.
$\mathcal{A} := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$
$\mathcal{S} := \{\mathtt{abc}, \mathtt{abc}, \mathtt{bcde}, \mathtt{ef}, \mathtt{bde}, \mathtt{f}, \mathtt{eed}\}$

Définition 2.2 (Concaténation et sous-séquences)**.** *Pour deux séquences $s_1 = \sigma_1\ldots\sigma_2$ et $s_2 = \sigma_3\ldots\sigma_4$, la séquence $s_1s_2 = \sigma_1\ldots\sigma_2\sigma_3\ldots\sigma_4$ est la* concaténation *de s_1 et s_2. On dit que s' est une* sous-séquence *de s si correspond à une suite d'éléments consécutifs dans s*

Définition 2.3 (Suffixes, préfixes et extensions)**.** *Pour une séquence donnée $s = \sigma_1\sigma_2\ldots\sigma_m$, la séquence s' est appelée* suffixe *de s si les $|s'|$ derniers états de s forment la sous-séquence s'. De plus, on dira que s' est un* préfixe *de s si les $|s'|$ premiers états de s forment la sous-séquence s'. On note $\texttt{suffixes}(s)$ (respectivement $\texttt{prefixes}(s)$) l'ensemble des suffixes (resp. préfixes) de s. Par ailleurs si s' est suffixe de s, on dit que s est une* extension *(par la gauche) de s'.*

Précisons que nous n'utiliserons pas de notions de séquence vide ici.

EXEMPLE

Pour $s = \texttt{bcde}$ une séquence de $\mathcal{S}$, `cd` est une sous-séquence de s qui n'est ni préfixe ni suffixe. `ce` n'est pas une sous-séquence de s.
$\texttt{suffixes}(s) = \{\texttt{bcde}, \texttt{cde}, \texttt{de}, \texttt{e}\}$
$\texttt{prefixes}(s) = \{\texttt{bcde}, \texttt{bcd}, \texttt{bc}, \texttt{b}\}$
`bcde` est une extension de `cde`

Définition 2.4 (Ordre)**.** *L'*ordre *de la séquence s est sa longueur et est noté $|s|$.*

Définition 2.5 (Support)**.** *Nous utilisons la notation $c(s)$ pour désigner le nombre d'occurrences de s en tant que sous-séquence dans l'ensemble de données $\mathcal{S}$. Nous définissons également le vecteur $C_s = (c(s\sigma))_{\sigma\in\mathcal{A}}$ qui correspond aux occurrences d'états* suivant *la séquence s.*

EXEMPLE

Pour les séquences $\mathcal{S} := \{\texttt{abc}, \texttt{abc}, \texttt{bcde}, \texttt{ef}, \texttt{bde}, \texttt{f}, \texttt{eed}\}$, on a
$c(\texttt{e}) = 5$, $c(\texttt{bc}) = 3$, $c(\texttt{ca}) = 0$ et
$C_{\texttt{e}} = (0, 0, 0, 1, 1, 1)$
$C_{\texttt{ab}} = (0, 0, 2, 0, 0, 0)$

2.2 Probabilités et modèles séquentiels

Dans l'estimation de séquences discrètes, nous voulons connaître $\mathbb{P}(\sigma|\sigma_1 \dots \sigma_k)$ *i.e.* la probabilité d'observer l'état σ après la séquence $\sigma_1 \dots \sigma_k$.

Définition 2.6 (Probabilité de transition)**.** *Pour un jeu de données $\mathcal{S}$, la* probabilité de transition *de s vers σ correspond à l'estimation du maximum de vraisemblance de $\mathbb{P}(\sigma|s)$ donné par*

$$p(\sigma|s) = \frac{c(s\sigma)}{\sum_{\sigma'\in A} c(s\sigma')} \tag{2.1}$$

On note $P_s = (p(\sigma|s))_{\sigma\in\mathcal{A}}$ la distribution *des états possibles après la séquence s.*

EXEMPLE

Pour les séquences $\mathcal{S} := \{\texttt{abc}, \texttt{abc}, \texttt{bcde}, \texttt{ef}, \texttt{bde}, \texttt{f}, \texttt{eed}\}$, on a
$p(\texttt{f}|\texttt{e}) = 1/3$, $p(\texttt{e}|\texttt{bc}) = 0/3$ et
$P_{\texttt{e}} = (0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$
$P_{\texttt{ab}} = (0, 0, 1, 0, 0, 0)$

Nous ne ferons pas la différence, dans les notations, entre l'estimation et la vraie valeur (inconnue en pratique) que l'on veut estimer. On parlera parfois de *« dynamique de flux »* pour décrire les règles ayant, dans un système donné, engendré les traces $\mathcal{S}$. Dans ce cadre, les *« dépendances séquentielles »* peuvent être décrites comme des situations où la transition vers un état dépend non seulement du dernier état mais d'autres états précédents *i.e.* où la propriété de Markov n'est pas valable.

On considère que $\mathcal{S}$ est la seule source d'information disponible. On omet également de mentionner cet ensemble dans l'estimation (équation 2.1). Le dénominateur de l'équation 2.1 n'est pas $c(s)$ car on ne compte pas les occurrences de s en fin de séquence afin que P_s soit effectivement un vecteur de probabilités.

En pratique, le calcul de l'équation 2.1 pour toute séquence d'états n'est pas possible. Par ailleurs, certaines sous-séquences ont pu ne jamais être observées. Un enjeu dans les algorithmes de prédictions utilisés notamment dans la compression de textes [6] est ainsi de fournir une approximation rapide lors du calcul des probabilités pour un nombre de séquences restreint. Pour représenter des modèles ayant ainsi une information restreinte, nous allons définir des *modèles séquentiels*.

Définition 2.7 (Modèle séquentiel)**.** *Un* modèle séquentiel $\mathcal{M}$ *est un ensemble de séquences d'état (aussi appelées* contextes*) tel que, pour toute séquence s, il existe au moins un suffixe s' de s dans $\mathcal{M}$. Pour une séquence s donnée et un modèle séquentiel $\mathcal{M}$, la probabilité de transition vers $\sigma \in A$ depuis s selon $\mathcal{M}$ est :*

$$p^{\mathcal{M}}(\sigma|s) = p(\sigma|s^*) \tag{2.2}$$

où s^* est le plus grand contexte suffixe de s dans $\mathcal{M}$. On note $P_s^{\mathcal{M}} = \left(p^{\mathcal{M}}(\sigma|s)\right)_{\sigma\in\mathcal{A}}$ la distribution des états possibles après la séquence s.

EXEMPLE

Pour les séquences $\mathcal{S} := \{\texttt{abc}, \texttt{abc}, \texttt{bcde}, \texttt{ef}, \texttt{bde}, \texttt{f}, \texttt{eed}\}$ et le modèle séquentiel $\mathcal{M} := \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{ee}\}$, on aura $p^{\mathcal{M}}(\texttt{f}|\texttt{dfe}) = p(\texttt{f}|\texttt{e}) = \frac{1}{3}$

Notons que, dans la définition 2.7, il n'y a pas de contraintes quant à la longueur des contextes dans $\mathcal{M}$. Ce modèle se rapproche donc des modèles de Markov d'ordre supérieur [44] discutés en introduction (section 1.3). Cette définition nous permettra de définir plus facilement les réseaux d'ordre supérieur dans la section 2.4.

Définition 2.8 (Entropie)**.** *Soit P une distribution de probabilité discrète dans l'ensemble Ω. L'*entropie *de la distribution P, notée $H(P)$, est*

$$H(P) = -\sum_{x \in \Omega} P(\sigma) \log_2 P(\sigma) \tag{2.3}$$

Définition 2.9 (Divergence de Kullback-Leibler)**.** *Soit P et Q, deux distributions de probabilité discrètes sur un même ensemble Ω tel que, pour tout $x \in \Omega$, $Q(x) > 0$ si $P(x) > 0$, la divergence de Kullback-Leibler, notée $D_{KL}(P||Q)$, est donnée par*

$$D_{KL}(P||Q) = \sum_{x \in \Omega} P(x) \log_2 \left(\frac{P(x)}{Q(x)} \right) \tag{2.4}$$

On exprimera toujours les quantités correspondant à H et D_{KL} en bits (log en base 2). La divergence $D_{KL}(P||Q)$ a une valeur de 0 bits lorsque $P = Q$. La divergence de Kullback-Leibler est un calcul de dissimilarité entre deux distributions. Elle servira à comparer les probabilités de transitions de modèle séquentiel pour savoir si une extension (définition 2.3) d'un contexte donné mène à une distribution suffisamment différente.

2.3 Graphes

Nous donnons dans cette partie quelques définitions relatives aux *graphes*. Nous cherchons seulement à définir les concepts qui nous seront utiles par la suite. Ainsi la définition 2.10 correspond au type de graphe avec lequel nous travaillons la plupart du temps.

Définition 2.10 (Graphe simple orienté)**.** *Un* graphe simple orienté *est noté $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ avec $\mathcal{V}$ l'ensemble des* sommets *ou* nœuds *et $\mathcal{E}$ l'ensemble des* arcs*. Ce dernier est un ensemble de paires ordonnées d'éléments de $\mathcal{V}$. Le* poids *d'un arc est donné par la fonction $w : \mathcal{E} \to \mathbb{R}$. Il n'y a ainsi ni symétrie ni réciprocité dans les arcs entre les sommets. En d'autres termes, si on a deux sommets x et y reliés par un arc (x, y), l'inverse n'est pas automatiquement vrai. On autorise toutefois l'existence d'arc de la forme (x, x) appelée* boucle*. Par ailleurs, c'est un graphe* simple *car un arc (x, y) n'est présent qu'une fois dans $\mathcal{E}$.*

Définition 2.11 (Graphe complet ou Clique)**.** *Un* graphe complet *ou* clique *est un graphe simple orienté tel que chaque paire $(x, y) \in \mathcal{E}$.*

Définition 2.12 (Sous-graphe)**.** *Un sous-graphe est un graphe contenu dans un autre graphe. $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', w)$ est un sous-graphe de $G = (\mathcal{V}, \mathcal{E}, w)$, si $\mathcal{E}'$ l'ensemble de ses arcs et $\mathcal{V}'$ l'ensemble de ses sommets est un sous-ensemble de $\mathcal{E}$ et $\mathcal{V}$ respectivement. Un sous-graphe est dit induit par un sous-ensemble $\mathcal{V}' \subseteq \mathcal{V}$ lorsque le sous-graphe obtenu, $\mathcal{G}'$, conserve tous les arcs présents dans $\mathcal{G}$ entre les sommets $\mathcal{V}'$.*

Définition 2.13 (Chemin)**.** *Un chemin dans un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ est une suite finie de sommets $(v_1, v_2, \ldots, v_m)$ telle que $\forall i \in [2, m]$, $(v_{i-1}, v_i) \in \mathcal{E}$.*

Définition 2.14 (Connexité)**.** *Un graphe est dit* fortement connexe *si pour tout couple de sommets $\{u, v\} \in \mathcal{V} \times \mathcal{V}$ il existe un chemin entre u et v.*

Définition 2.15 (Densité)**.** *La* densité $D(\mathcal{G})$ *de $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ est le rapport entre le nombre d'arcs dans $\mathcal{G}$ et le nombre maximal d'arcs possible :*

$$D(\mathcal{G}) = \frac{|\mathcal{E}|}{|\mathcal{V}|^2} \tag{2.5}$$

Une densité de 0 correspond à un graphe sans aucun arc et une densité de 1, à un graphe complet. Dans le cas d'une faible (respectivement forte) densité, on parlera de graphes creux *(resp.* denses*).*

Définition 2.16 (Degré)**.** *Le* degré sortant *de $v \in \mathcal{V}$, noté $deg^+(v)$, est le nombre d'arcs sortants du sommet v. Le* degré entrant *de $v \in \mathcal{V}$, noté $deg^-(v)$, est le nombre d'arcs vers le sommet v. Le* voisinage sortant *de $v \in \mathcal{V}$, noté $\mathcal{N}^+(v)$, correspond à l'ensemble $\{u \in V, (v, u) \in \mathcal{E}\}$. Le* voisinage entrant *de $v \in \mathcal{V}$, noté $\mathcal{N}^-(v)$, correspond à l'ensemble $\{u \in V, (u, v) \in \mathcal{E}\}$*

On parlera plus généralement du *degré* d'un sommet pour désigner le nombre d'arcs entrants ou sortants du sommet.

Définition 2.17 (Marche aléatoire)**.** *La* marche aléatoire *est un processus stochastique qui décrit un chemin constitué d'une suite de pas aléatoires réalisée sur le graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$.*
Ce processus est dit markovien *car le nœud suivant n'est déterminé que par le nœud courant du* marcheur *(agent imaginaire se déplaçant sur le graphe).*
Soit u le sommet courant du marcheur, la probabilité que le marcheur visite le voisin v est en effet proportionnelle à $w(u, v)$ si $(u, v) \in \mathcal{E}$ et 0 sinon.

Nous parlerons de *marche* pour décrire un circuit résultant d'une marche aléatoire. Toutefois, nous utiliserons parfois la métaphore du *marcheur* pour décrire des dynamiques de flux qui ne peuvent être formulées comme des processus markoviens.

2.4 Réseaux d'ordre supérieur

Nous définissons ici les réseaux d'ordre supérieur (HON) étudiés dans ce document (définition 2.18) et discutons certaines propriétés intéressantes de ces objets. Le formalisme détaillé ici, largement tiré de travaux existants [84, 71], n'incorpore pas forcément toutes les constructions trouvées dans la littérature ou certaines alternatives discutées

dans ce document. Il permet toutefois de ramener la construction des HON détaillée dans le chapitre suivant à simplement déterminer quelles dépendances séquentielles doivent être prises en compte dans un modèle séquentiel.

Définition 2.18 (Réseau d'ordre supérieur)**.** *Soit un ensemble d'états $\mathcal{A}$ et un modèle séquentiel $\mathcal{M}$ tel que*

$$\mathcal{A} \subseteq \mathcal{M} \tag{2.6}$$

$$\forall s \in \mathcal{M}, \texttt{prefixes}(s) \subseteq \mathcal{M} \tag{2.7}$$

Le réseau d'ordre supérieur (HON) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ représentant $\mathcal{M}$ est un graphe orienté pondéré où chaque sommet de $\mathcal{V}$, appelé nœud-mémoire correspond à une séquence de $\mathcal{M}$. Par simplicité, on considère le sommet et le contexte comme un même objet. On parlera ainsi de l'ordre (longueur) d'un nœud-mémoire.
Soit $\sigma \in \mathcal{A}$ et un contexte s de $\mathcal{M}$ pour lesquels $p^{\mathcal{M}}(\sigma|s) > 0$, $\mathcal{E}$ inclura un arc $(s, s^*\sigma)$ de poids $w(s, s^*\sigma) = p^{\mathcal{M}}(\sigma|s)$ où

$$s^* = \arg\max_{s'\sigma \in \mathcal{M}} \{|s'|, s' \in \texttt{suffixes}(s)\} \tag{2.8}$$

La construction des HON est totalement déterminée par le modèle séquentiel $\mathcal{M}$. Le cas le plus simple est de considérer le modèle $\mathcal{M} = \mathcal{A}$ *i.e.* on considère le réseau sans-mémoire où la probabilité de l'état suivant dépend seulement de l'état précédent. Notons que la condition 2.6 est une conséquence directe de la définition des modèles séquentiels. La façon dont le modèle séquentiel est choisi va permettre de différencier les modèles de HON discutés dans le chapitre 3.

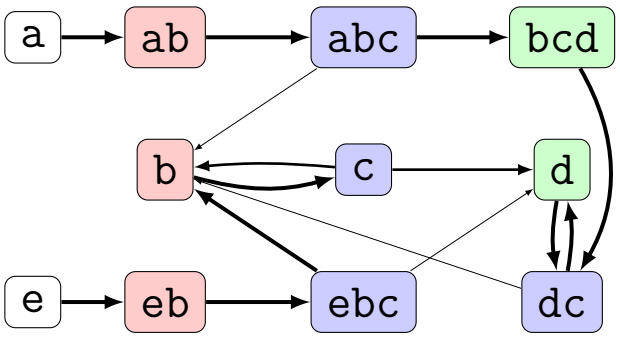


FIGURE 2.1 – Exemple de réseau HON

EXEMPLE

Le HON illustré en figure 2.1 inclut les nœuds-mémoires `abc` et `bcd`. Ils représentent les états `c` et `d` respectivement. On a $\texttt{abcd} \notin \mathcal{V}$ et $p^M(\texttt{d}|\texttt{abc}) > 0$, le graphe contient donc l'arc $(\texttt{abc}, \texttt{bcd})$. Notons qu'il n'y a pas d'arc $(\texttt{abc}, \texttt{cd})$ ou $(\texttt{abc}, \texttt{d})$ dans $\mathcal{G}$. C'est en effet la relation $(\texttt{abc}, \texttt{bcd})$ qui préserve le plus d'information quant à l'origine d'un marcheur aléatoire.

La façon dont l'ensemble des arcs $\mathcal{E}$ est défini est une formulation plus courte de l'idée développée par Xu *et al.* [84]. Leur construction n'était pas formellement définie et implique un algorithme basé sur le *reroutage* d'arcs conduisant au même résultat.

Les relations entre nœuds-mémoires forment le principal intérêt des HON. En effet, comme nous le montrons dans le théorème 2.1, une marche aléatoire (bien qu'étant un processus sans mémoire) dans un HON construit à partir de $\mathcal{M}$ va correspondre à une simulation de $\mathcal{M}$ (donc un processus avec mémoire).

Théorème 2.1. ***Marche aléatoire comme simulation d'un modèle de Markov d'ordre supérieur*** *Soit $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ et $s = \sigma_1\sigma_2 \dots \sigma_m \in \mathcal{V}$ une représentation de σ_m, il existe un chemin $\sigma_1 \to \sigma_1\sigma_2 \to \dots \to s$ suivi par un marcheur aléatoire commençant à σ_1 avec une probabilité $\prod_{i=2}^{m} p^{\mathcal{M}}(\sigma_i|\sigma_1 \dots \sigma_{i-1}) > 0$*

Démonstration. En raison de 2.7, chaque séquence $(\sigma_1, \sigma_1\sigma_2, \dots, s)$ est un noeud étiqueté de $\mathcal{V}$ en tant que préfixe de s. Soit s'_i le préfixe d'ordre $i < m$ de s. Puisque $s'_i\sigma_{i+1}$ est aussi un préfixe de s, nous avons $c(s'_i\sigma_{i+1}) > 0$ donc il existe un arc $e = (s'_i \to s'_i\sigma_{i+1}) \in \mathcal{E}$ avec $w(e) = p(\sigma_{i+1}|s'_i) > 0$ par définition puisque s'_i est le plus grand suffixe de s'_i (équation 2.8). □

Le circuit $(v_1, v_2, \dots, v_m)$ résultant d'une marche aléatoire sur le réseau d'ordre supérieur $\mathcal{G}$ correspond à une séquence d'état $(\sigma_1, \sigma_2, \dots, \sigma_m)$ où chaque σ_i est l'état représenté par v_i. On discutera dans le chapitre suivant de la capacité d'un réseau de bien simuler une dynamique de flux (observée à travers un jeu de données $\mathcal{S}$).

Définition 2.19 (Ordre du réseau)**.** *L'*ordre *du réseau d'ordre supérieur $\mathcal{G}$ est l'ordre maximum d'un nœud-mémoire dans $\mathcal{G}$. On note $\mathcal{V}_k$ l'ensemble des nœuds-mémoires d'ordre k dans $\mathcal{G}$.*

Définition 2.20 (Représentations et nœuds-mémoires)**.** *Pour un état $\sigma \in A$, $\mathcal{V}(\sigma) \subset \mathcal{V}$ est l'ensemble des* représentations *de σ dans $\mathcal{G}$ i.e. les séquences dont le dernier état est σ. On note $N_{\mathcal{V}}(\sigma) = |\mathcal{V}(k)|$ le nombre de représentations de σ dans $\mathcal{G}$.*

EXEMPLE

Le HON illustré en figure 2.1 est d'ordre 3.
Les représentants de l'état `c` (en bleu) sont $\mathcal{V}(\texttt{c}) = \{\texttt{abc}, \texttt{c}, \texttt{ebc}, \texttt{dc}\}$, on a donc $N_{\mathcal{V}}(\texttt{c}) = 4$.
Les nœuds-mémoires d'ordre 2 sont $\mathcal{V}_2 = \{\texttt{ab}, \texttt{eb}, \texttt{dc}\}$.

Comme nous le verrons dans le chapitre suivant, le nombre de nœuds dans un réseau d'ordre supérieur peut être élevé par rapport au nombre d'états $|\mathcal{A}|$ qui correspond au nombre de nœuds dans le réseau sans-mémoire.
La construction proposée dans la définition 2.18 implique également une augmentation du nombre d'arcs dans le réseau. Celle-ci est toutefois plus « modérée ». En effet, le nombre maximum d'arcs dans un graphe orienté augmente quadratiquement avec le nombre de nœuds. Dans le cas des HON, chaque nœud de $\mathcal{G}$ ne peut être connecté qu'à au plus un représentant de chaque état de $\mathcal{A}$ (*i.e.* $\max_{v\in\mathcal{V}} deg^+(v) \leq |\mathcal{A}|$). La densité du graphe est donc limitée en conséquence (voir théorème 2.2).

Théorème 2.2. ***Densité des* Hon** *Soit $\mathcal{M}$ un modèle séquentiel et $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ un réseau d'ordre supérieur construit à partir de $\mathcal{M}$. On a*

$$D(\mathcal{G}) \leq \frac{|\mathcal{A}|}{|\mathcal{V}|} \tag{2.9}$$

Démonstration. Premièrement, si $deg^+(\sigma)$ est le degré sortant de σ dans le réseau sans-mémoire alors $|\mathcal{E}| \leq \sum_{\sigma\in\mathcal{A}} N_{\mathcal{V}}(\sigma) deg^+(\sigma)$. En effet, pour $s \in \mathcal{M}$ représentant d'un état $\sigma \in \mathcal{A}$ et tout autre $\sigma' \in \mathcal{A}$, on a $(p^{\mathcal{M}}(\sigma'|s) > 0) \rightarrow (p^{\mathcal{M}}(\sigma'|\sigma) > 0)$ car $c(s\sigma') \leq c(\sigma\sigma')$.
On a ensuite $\sum_{\sigma\in\mathcal{A}} N_{\mathcal{V}}(\sigma) deg^+(\sigma) \leq |\mathcal{A}| \sum_{\sigma\in\mathcal{A}} N_{\mathcal{V}}(\sigma) = |\mathcal{A}||\mathcal{V}|$. □

2.5 Jeux de données

Nous allons ici nous attacher à présenter les jeux de données séquentielles qui ont été utilisés dans cette thèse. Nous tâcherons d'expliciter quel mécanisme suggère l'existence (ou l'absence) de dépendances séquentielles. Notons toutefois que les expérimentations détaillées tout au long de ce document ne constituent pas des cas d'étude complets sur les systèmes concernés.

Les caractéristiques des quatre jeux de données utilisés sont données dans le tableau 2.1. Ils constituent tous des séquences de « déplacements ». Deux d'entre eux correspondent à des déplacements dans un espace géographique (*AIR* et *PORTS* détaillés en sections 2.5.1 et 2.5.2) et deux à des déplacements dans un espace virtuel (*MSNBC* et

WIKI détaillés en sections 2.5.3 et 2.5.4).

Les séquences de déplacements constituent une source majeure d'applications pour l'analyse des réseaux d'ordre supérieur. Les jeux de données utilisés sont toutefois différents en termes de nature, de nombre d'états ou de nombre de séquences. En outre, ils ont tous été utilisés dans des travaux précédents comme applications de l'exploration de réseaux d'ordre supérieur. Pour tous les jeux de données utilisés nous avons décidé de supprimer les répétitions d'états lorsqu'elles se présentaient. Ce choix sera discuté dans le cadre des jeux de données *PORTS* et *MSNBC*.

La section 2.5.5 présente également d'autres jeux de données ou systèmes pouvant être étudiés avec les outils présentés dans ce document. Dans la section 2.5.6, nous discutons d'applications où les séquences ne sont pas disponibles directement mais sont obtenues indirectement après transformations des données de base.

TABLE 2.1 – Tableau récapitulatif des jeux de données utilisés

Nom	Notation	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{S}\rvert$	Ordre Min/Max	Ref.
Transport Aérien	*AIR*	175	286 810	2/14	[75, 70]
Transport Maritime	*PORTS*	909	4 243	2/183	[84, 72]
MSNBC	*MSNBC*	17	388 434	2/1810	[78, 75]
Wikispeedia	*WIKI*	100	9 573	2/22	[78, 75]

2.5.1 Transport Aérien

Le jeu de données *AIR* [70, 75] sont des itinéraires de vols aux États-Unis qui ont eu lieu au cours du premier trimestre 2011. Ces données sont extraites de la base de données *RITA TransStat* 2014 [1]. Une séquence correspond aux aéroports utilisés par un voyageur au cours d'un voyage (incluant l'aéroport de départ et d'arrivée). Les états sont ici les codes IATA des aéroports américains (on en compte 175).

Les itinéraires peuvent correspondre à des aller-retours avec, éventuellement, des correspondances (voir les exemples de séquences dans le tableau 2.2). Notons que les séquences ne correspondent pas toujours à des aller-retours. La présence d'aller-retours est un exemple clair de potentielle dépendance séquentielle dans ce système. Une question reste toutefois de savoir si des dépendances d'ordre 2 permettent de prédire suffisamment bien les trajets ou si des ordres plus élevés sont nécessaires. En effet, certains aller-retours peuvent nécessiter plus d'étapes si des correspondances sont nécessaires.

1. `https://www.transtats.bts.gov/`

TABLE 2.2 – Exemples de séquences *AIR*

Séquences *AIR*
DFW ORD DFW
ABQ DFW LIT DFW ABQ
AUS DFW BOS BWI
AUS DFW BOS BWI DFW AUS

2.5.2 Transport Maritime

Le jeu de données *PORTS* [84] est formé des séquences de ports visités par des navires extraits de la *Lloyd's Maritime Intelligence Unit* qui recueille les enregistrements de capitainerie (date d'arrivée et départ de navires faisant escale dans le port) [15]. L'échantillon correspond à des observations qui ont eu lieu entre le 1er avril et le 31 juillet 2009. Une séquence correspond à l'historique de navigation d'un navire transportant des marchandises (containers, gaz, *etc.*, voir les exemples de séquences dans le tableau 2.3). Les états correspondent donc à des ports capables d'accueillir de tels navires (909 dans la base). Notons que la route maritime empruntée par les navires n'est pas prise en compte. Xu *et al.* [84] emploie un jeu de données similaire mais sur une période différente. Leur échantillon n'est pas publiquement accessible.

À la différence des passagers aériens, les séquences de ports forment un ensemble plus complexe. En effet, si des séquences d'aller-retours entre deux ports sont possibles, beaucoup de navires suivent des routes pouvant impliquer davantage de ports. Les aléas de navigation ou des calendriers altèrent ses routes. De plus, un navire peut être arrêté sur une longue période (par exemple pour des réparations) avant d'être affecté à une autre zone. Le jeu de données de base contient également des répétitions d'états. Cela correspond aux cas où un navire a quitté un port pour y retourner peu après (suite à un mouillage temporaire ou une avarie) ou simplement aux cas où une escale n'a pas été correctement enregistrée. Nous avons donc choisi de supprimer les répétitions d'états pour ce jeu de données.
Toutefois, comme pour le jeu de données *AIR*, l'existence de dépendances séquentielles peut s'expliquer par la régularité des routes et les contraintes physiques existantes. Par exemple, un *super-tanker* transportant du pétrole brut va nécessairement effectuer des trajets entre les ports à proximité de champs pétrolifères et des ports à proximité de raffineries.

2.5.3 MSNBC

Le jeu *MSNBC* [75, 78] est constitué de données de navigation issues du site de la chaîne de télévision américaine MSNBC NEWS. Chaque séquence correspond à une ses-

TABLE 2.3 – Exemples de séquences *PORTS*

Séquences *PORTS*
`Singapour Bandar_Abbas Jebel_Ali Dammam Alang`
`Singapour Sabang Lumut Singapour Lahad Datu Sandakan Singapour`
`Zhanjiang Jakarta Surabaya Lianyungang Qingdao Singapour Dammam`
`Umm_Qasr Jebel_Ali Umm_Qasr Jebel_Ali Umm_Qasr Jebel_Ali Alang`

sion de navigation d'un utilisateur sur le site (voir les exemples de séquences dans le tableau 2.4). La séquence ne correspond toutefois pas aux pages visitées par l'utilisateur mais aux catégories auxquelles ces pages appartiennent, il y a ainsi 17 grandes catégories et donc 17 états possibles.

À la différence des jeux de données de trajectoires géographiques détaillés plus haut, il est plus compliqué d'expliquer pourquoi un tel système comporterait des dépendances séquentielles. Toutefois, il est probable que le regroupement en catégories puisse donner lieu à des dépendances (ce mécanisme est décrit dans la section 2.5.6). À l'instar de *PORTS*, nous avons également supprimé les répétitions de séquences.

TABLE 2.4 – Exemples de séquences *MSNBC*

Séquences *MSNBC*
`on-air msn-news local health tech health opinion health local`
`weather frontpage misc weather`
`news local tech`
`on-air msn-news`

2.5.4 Wikispeedia

Le jeu de données *WIKI* [75, 78] est issu du jeu en ligne « Wikispeedia ». Dans ce jeu, deux articles du site *Wikipedia* (édition anglophone) sont donnés aux joueurs au hasard. Son but est de partir du premier pour arriver au second en utilisant le moins d'hyperliens possible.

Les joueurs ne connaissent pas la structure complète du réseau ; ils doivent donc se baser uniquement sur les informations locales qu'ils voient sur chaque page et leurs attentes quant aux articles susceptibles d'être interconnectés. À l'instar de Scholtes [75], nous ne retenons ici que les 100 pages les plus visitées au cours du jeu par l'ensemble des participants. Cette limitation est chez Scholtes liée à la volonté d'équilibrer la taille de l'échantillon avec le nombre d'articles total. Dans notre cas, il s'agit de pouvoir comparer

nos résultats avec un jeu de données déjà étudié dans la littérature.

Certaines séquences (voir des exemples de séquences dans le tableau 2.5) peuvent être courtes (*i.e.* ordre 2 : la solution serait trouvée en un clic) et correspond ainsi aux trajectoires prises par les joueurs quand ils passent par des sujets très généraux. Notons que les retours à la page précédente sont pris en compte. On peut ainsi suspecter l'existence de dépendances séquentielles de type aller-retours lorsqu'un joueur passe d'un article spécifique à un article plus général sans trouver le lien qu'il espérait. Toutefois, le filtre sur les pages populaires peut potentiellement amoindrir cet effet.

TABLE 2.5 – Exemples de Séquences *WIKI*

Séquences *WIKI*
`Europe North_America United_States President_of_the_United_States`
`Biology Science Physics`
`France World_War_II Nuclear_weapon`
`England Great_Britain England`

2.5.5 Autres jeux de données séquentielles

La littérature contient d'autres jeux de données de séquences géographiques. On peut en particulier mentionner les déplacements « continus », c'est-à-dire qui correspondent à des suites de relevés GPS indiquant la position d'un véhicule.
Ce type de données se retrouve dans la prédiction des trajectoires de navires [17]. Les études dans ce domaine se basent sur les données de positionnement AIS : il s'agit d'un « Système d'identification automatique » obligatoire pour les gros navires permettant d'identifier un navire ainsi que sa position. Un autre domaine est la prédiction des déplacements des taxis dans un environnement urbain. Les trajectoires de taxis de la ville de Porto entre 2013 et 2014 ont ainsi donné lieu à un défi de la conférence ECML/PKDD [59]. Ce jeu de données est utilisé pour l'analyse de réseau d'ordre supérieur par Saebi *et al.* [71]. Pour se ramener à une séquence d'évènements discrets, les auteurs remplacent les coordonnées par le quartier dans lequel le véhicule est présent. Les « quartiers » sont définis par la station de police la plus proche. L'influence de cette transformation sur la présence de dépendances séquentielles n'est pas claire mais constitue une piste de recherche intéressante.

Un autre domaine où l'aspect séquentiel est pris en compte est celui de la « recommandation séquentielle » [39]. Le but est ici, à partir des articles vus et/ou achetés par un client sur un site de *e*-commerce, de proposer des recommandations tenant compte de ces historiques. La différence avec la recommandation « classique » est que la séquence de vue

ou d'achat est ici centrale. De nombreux jeux de données utilisés dans ce domaine correspondent à des avis laissés par des utilisateurs sur des produits, souvent culturels [45]. Il existe également une grande proximité entre la recommandation séquentielle et le domaine de la « fouille de motifs séquentiels » (*Sequential Pattern Mining*) [28]. Dans ce domaine, les séquences correspondent à un sous-ensemble d'objets acquis lors d'une transactions au lieu d'un état unique.

2.5.6 Inférences de séquences

Les deux types d'applications détaillées ici correspondent à des cas d'études où les séquences sont générées à partir d'autres jeux de données. Dans ce cadre, les séquences ne peuvent pas directement être rapprochées aux déplacements d'agents dans un espace. Ces approches ouvrent des perspectives intéressantes mais les choix effectués dans ces transformations ne permettent pas, selon nous, de constituer des *benchmarks* réutilisables pour évaluer les différents outils discutés dans cette thèse.

Il est possible de générer des séquences à partir d'un réseau connu en effectuant et enregistrant le résultat de marches aléatoires sur un graphe. Par définition, les séquences ainsi produites ne devraient pas contenir de dépendances séquentielles. Il est toutefois possible d'obtenir des dépendances séquentielles en projetant ces séquences de nœuds en séquences de catégories auxquelles ces nœuds appartiennent.
Une application étudiée par Rosvall *et al.* [70] est le réseau de citations entre articles scientifiques. Une marche aléatoire dans ce réseau consiste à partir d'un article puis, à chaque étape, de choisir une citation aléatoire de l'article courant. Toutefois, chaque article est associé à un journal particulier. En remplaçant dans cette marche les articles par les journaux, on peut s'attendre à voir émerger des dépendances séquentielles. En effet, des journaux tels que *Nature* ou *Science* regroupent des articles de domaines très divers. Il est ainsi probable qu'une marche ayant débuté avec une revue spécialisée dans, par exemple, la biologie cellulaire et passant dans la revue *Nature* retourne davantage vers des revues de biologie cellulaire. Notons que ce principe peut s'appliquer dans d'autre domaines où les entités sont agrégées en catégories plus larges (*i.e.* les jeux de données *MSNBC* ou des taxis de Porto discutés plus haut).
Une limite à l'utilisation de ces cas d'études est la transformation requise. Rosvall *et al.* [70] se concentre en effet sur la prise en compte des dépendances d'ordre 2. Le réseau d'ordre supérieur n'est donc pas généré à partir de séquences mais en listant tous les *trigrames* (séquences de 3 papiers ou 2 citations) existants. Une telle approche nécessite de poser comme *a priori* que les dépendances sont limitées à cet ordre.

L'analyse de réseaux dynamiques ou temporels [38] est proche de l'analyse de réseaux

d'ordre supérieur car l'ordre des interactions entre entités est importante dans les deux cas. Dans certains réseaux temporels, les relations (u, v, t) sont *dyadiques* (impliquant deux nœuds u et v) mais inclut une dimension temporelle (le moment t où cette relation est observée). On pourrait, dans ces cas, s'intéresser à l'existence de dépendances séquentielles. En effet, si u interagit avec v au temps t_1, il y a-t-il plus de chances que u interagisse avec v au temps t_2 si t_1 est proche de t_2 ? De tels effets ne peuvent pas être capturés par la méthode consistant à agréger les relations selon différentes fenêtres temporelles afin de réduire l'analyse du réseau temporel à l'analyse d'une suite de graphes statiques [76].

Scholtes *et al.* [76, 75] s'intéressent à ces applications à travers les jeux de données *e-mails* Enron [58] ou de contacts entre personnels d'un hôpital [40] (en utilisant des capteurs permettant de dire si deux personnes sont face-à-face à un instant t). Les auteurs proposent de générer les séquences à partir du réseau temporel en énumérant tous les « chemins temporellement cohérents » (*time-respecting paths*). Deux liens temporels (u, v, t_1) et (u, w, t_2) produiront ainsi la séquence (ou formeront une sous-séquence d'une plus longue séquence) u, v, w si la durée $(t_2 - t_1)$ est inférieure à un paramètre δ fixé *a priori*.
Un problème de cette définition apparaît dans le cas de relations temporelles non-dirigées comme dans les données de contacts entre personnes. Dans ce cas, (v, w, t_2) est équivalent à (w, v, t_2) : "v est face-à-face avec w au temps t_2" est en effet une relation symétrique. Or, l'algorithme proposé repose sur un ordre arbitraire correspondant à l'ordre dans lequel les extrémités des relations sont encodées en mémoire. Dans le premier cas, on observera ainsi la sous-séquence u, v, w et dans l'autre non. Cet arbitraire est absent des relations dirigées (pour les jeux de données d'*e-mails* notamment). On peut toutefois noter que dans ce cas que les messages avec plusieurs destinataires sont traités comme des messages séparés simultanés. Ce type d'application pourrait toutefois être intéressant dans le développement d'outils prenant en compte à la fois la dimension séquentielle et la dimension de sous-ensembles. Cet aspect sera discuté dans la conclusion de ce manuscrit.

TABLE 2.6 – Tableau récapitulatif des notations

	En rapport avec les séquences	
$\mathcal{A}$	Ensemble d'état (parfois appelé *Alphabet*)	Sec. 2.1
σ	Un état (élément générique de $\mathcal{A}$)	
$\mathcal{S}$	multi-ensemble de séquences (jeu de données)	
$s = \sigma_1\sigma_2\sigma_3\ldots$	Une séquence d'état	Def. 2.1
$s' = \sigma_3\sigma_4\sigma_5$	Une sous-séquence de s (toujours continue)	Def. 2.2
$\lvert s\rvert$	Ordre (longueur) de s	Def. 2.4
$c(s\sigma) : \mathcal{A} \to \mathbb{N}^+$	Occurrences de la séquence $s\sigma$ dans $\mathcal{S}$	Def. 2.5
$C_s = (c(s\sigma))_{\sigma\in\mathcal{A}}$	Occurrences de chaque σ suivant s dans $\mathcal{S}$	
	En rapport avec les probabilités	
$p(\sigma\lvert s) : \mathcal{A} \to [0,1]$	Probabilité de transition à partir de s vers σ	Def. 2.6
$P_s = (p(s\sigma))_{\sigma\in\mathcal{A}}$	Distribution suivant le contexte s	
$H(P)$	Entropie de la distribution discrète P	Def. 2.8
$D_{KL}(P\lvert\lvert Q)$	Divergence de Kullback-Leibler entre les distributions P et Q	Def. 2.9
	En rapport avec les réseaux	
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$	HON (Graphe orienté pondéré)	Def. 2.18
$\mathcal{V}$	Ensemble de noeuds (représentations d'état)	
$\mathcal{E}$	Ensemble des arcs	
$w : \mathcal{E} \to [0,1]$	Poids des arcs (probabilités de transitions)	
$\mathcal{V}(\sigma)$	Ensemble des représentations de l'état σ	Def. 2.20
$N_{\mathcal{V}} = (\lvert\mathcal{V}(\sigma)\rvert)_{\sigma\in\mathcal{A}}$	Nombre de représentation par état	
$\mathcal{V}_k$	Ensemble de représentation d'ordre k	Def. 2.19
π_v	PageRank du noeud v	Def. 4.1
	Modèles de HON	
FON_k	Réseau fixe d'ordre k	Def. 3.1
SN_k	«*State Network*» d'ordre k	Sec. 3.2.1
VON	Réseau ordre variable	Sec. 3.2.2
D_{KL}-VON(λ)	VON de [71] utilisant le facteur $\lambda \in \mathbb{R}^+$	Def. 3.2
D_{KL}-$\text{VON}_k(\lambda)$	*idem* avec un ordre maximal k	
MC-VON(α)	VON utilisant un test statistique avec seuil $\alpha \in (0,1)$	Def. 3.3
Acc	Score de Précision	Eq. 3.18

Chapitre 3

Modèles de Réseaux d'ordre supérieurs

3.1 Introduction

Dans le chapitre 1, nous avons présenté un exemple simple montrant l'intérêt des réseaux d'ordre supérieur. Nous avons ensuite proposé dans le chapitre 2 une définition formelle de ce concept (définition 2.18). Cependant, cette définition ne permet pas de savoir quel modèle séquentiel choisir. C'est le but de ce chapitre.

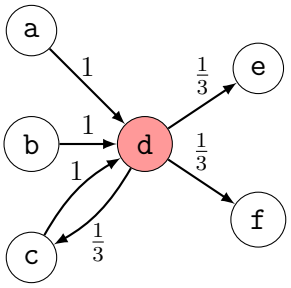
Comme suggéré dans l'introduction (section 1.4), nous souhaitons obtenir des modèles de HON parcimonieux. Dans ce cadre, deux caractéristiques sont importantes ; la taille des réseaux et la capacité des réseaux à bien modéliser les séquences observées. Pour ce qui est de la taille, le nombre de nœuds-mémoire est la variable principale à ajuster (voir théorème 2.2). Nous appellerons la deuxième caractéristique la *précision* du modèle. Nous pouvons également définir une métrique permettant de quantifier cette précision en se basant sur la probabilité de correctement prédire le prochain état à partir d'un contexte donné. Un enjeu important sera pour nous d'équilibrer ces deux valeurs, afin d'obtenir le meilleur compromis possible entre la taille et la précision. Un écueil possible serait en effet d'ajouter trop de nœuds-mémoire pour obtenir une meilleure précision, et d'aboutir à un réseau trop grand pour un gain de précision faible, voire d'entraîner un risque de sur-apprentissage. Au contraire, ne pas inclure assez de contextes peut entraîner un « sous-apprentissage » et risque d'omettre une partie importante de l'information.

La figure 3.1 présente différentes manières de construire des réseaux à partir de séquences dont les occurrences sont données dans la figure 3.1a. Comme déjà illustré dans le chapitre 1, le réseau « classique » de la figure 3.1b et qui sera défini comme FON_1 par la suite, ne permet pas de tenir compte de dépendances séquentielles telles que les aller-retours entre les états `d` et `c`. Dans cet exemple, les dépendances séquentielles concernent surtout l'état `d` car c'est le seul qui peut se produire après un état et avant un autre. Ainsi, une autre solution consiste à ajouter au réseau toutes les sous-séquences de longueurs 2 précédant un autre état. Cela aboutit au réseau 3.1c, qui sera appelé par la suite FON_2. L'intérêt de cette représentation est qu'une marche aléatoire débutant en `a` et allant à `d` va indirectement utiliser les probabilités de transitions correspondant aux arcs sortants de `ad` pour déterminer le prochain état visité. Cette marche sera ici plus fidèle aux données en entrée qu'une marche aléatoire effectuée sur le réseau 3.1c.

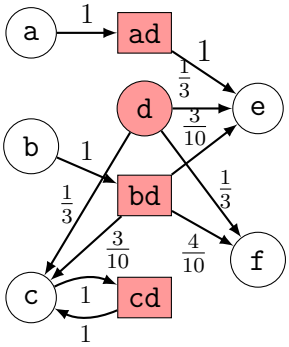
Il est toutefois possible de construire un réseau plus parcimonieux : les probabilités de transition suivant `d` et `bd` sont très similaires. On pourrait même supposer que le faible écart observé est dû à un biais d'échantillonnage. Dans ce cas, il est possible de construire un réseau en utilisant un modèle séquentiel ne contenant pas `bd`. Cela aboutit au réseau illustré en figure 3.1d. Nous obtenons un réseau de taille plus faible et espérons ici ne pas

Séquence	#
de	2
ade	2
bde	3
df	2
bdf	4
bdc	3
cdc	3

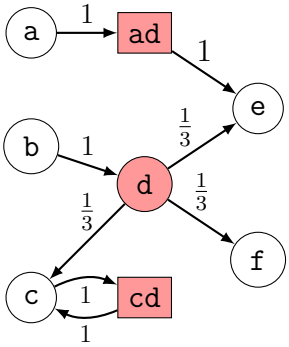
(a) Occurrences de séquences dans $\mathcal{S}$



(b) Réseau d'ordre 1



(c) Réseau d'ordre 2



(d) Réseau d'ordre variable

FIGURE 3.1 – Exemple de construction de réseaux à partir des séquences 3.1a pour un ensemble d'états possibles $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. Par exemple, la séquence `bdf` est observée quatre fois.

« trop » perdre en terme de précision *i.e.* les états visités par un marcheur aléatoire dans le réseau 3.1d devraient autant (voir davantage) correspondre aux séquences $\mathcal{S}$ que le réseau 3.1c. Le réseau 3.1d sera qualifié d'ordre variable (ou VON) par la suite car, dans ce cas, on examinera au cas par cas l'apport d'information issue de l'ajout d'un nœud-mémoire au modèle.

Nous détaillons les différentes stratégies de sélection dans la section 3.2. Nous nous concentrons particulièrement sur les modèles de réseau d'ordre variable. Nous définissons un algorithme générique de construction de réseaux d'ordre variable inspiré des travaux de Saebi et al. [71] et montrons que la définition de ces modèles se ramène à la façon de définir la « pertinence » d'un contexte donné *i.e.* si la distribution des états suivants après ce contexte est « significativement » différente par rapport à un contexte pertinent précédent. Dans la section 3.3, nous proposons un modèle de réseau d'ordre variable, noté MC-VON, qui définit la « pertinence » par un test statistique. Nous discutons, dans la section 3.4, du compromis taille/précision obtenu en comparant les différents modèles sur des jeux de données réels. Nous étudions aussi les caractéristiques de contextes retenus dans les réseaux d'ordre supérieur afin de comparer les stratégies de sélection.

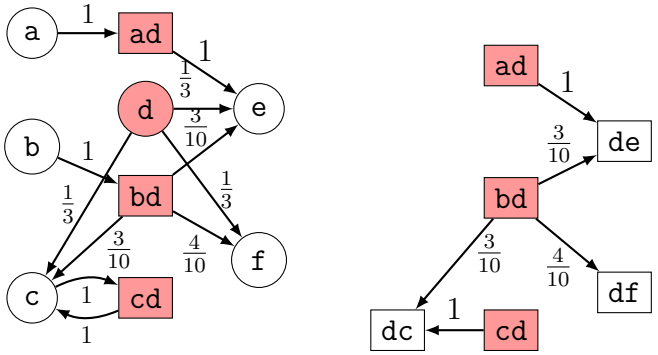
3.2 État de l'art

Dans cette section, nous allons nous concentrer sur les deux différentes approches majeures possibles : les réseaux d'ordre fixe présentés dans la section 3.2.1 et les réseaux d'ordre variable présentés dans la section 3.2.2. Nous proposons un cadre générique pour définir les réseaux d'ordre variable qui sera utilisé pour définir notre modèle MC-VON.

3.2.1 Les réseaux d'ordre fixe

Les modèles d'ordre fixe sont les premiers modèles HON à être étudiés dans la littérature. Ils sont qualifiés « d'ordre fixe », car le paramètre de l'ordre doit être fixé *a priori* et est valable pour tout le système. Nous allons présenter la façon dont ils sont construits mais aussi les problèmes inhérents à ces modèles.

Les « réseaux d'états » (*State Networks*) [70], notés SN_k, sont les premiers modèles d'ordre fixe étudiés dans la littérature. Ils ne rentrent pas dans le cadre de la définition 2.18. En effet, ils peuvent être définis comme des sous-graphes du graphe de De Bruijn sur l'alphabet $\mathcal{A}$. Un graphe de De Bruijn est un graphe orienté qui représente tous les mots de longueur k sur un alphabet donné (ici $\mathcal{A}$). Un nœud $\sigma_1\sigma_2, \dots \sigma_l$ est connecté aux nœuds $\sigma_2 \dots \sigma_l\sigma'$ pour tout $\sigma' \in \mathcal{A}$. Contrairement aux graphes de De Bruijn complet, seules les sous-séquences et les transitions présentes dans le jeu de données en entrée sont représentées. La figure 3.2b correspond au réseau SN_2 construit à partir du jeu de données de l'exemple 3.1a. L'avantage de ce modèle est qu'il intègre toutes les dépendances susceptibles d'exister à l'ordre k. Néanmoins, les SN_k ne contiennent que des nœuds-mémoires d'ordre k et non l'information présente dans les ordres inférieurs. Pour utiliser un tel modèle, par exemple pour effectuer une marche aléatoire sur les états, il faudrait donc travailler avec toutes les « couches » $(\text{SN}_1, \text{SN}_2, \dots \text{SN}_k)$.



(a) FON_2 (b) SN_2

FIGURE 3.2 – Différence entre les modèles FON_k et SN_k construit à partir des données de la figure 3.1a.

Pour cette raison, on préférera travailler avec des modèles qui combinent les différentes « couches », notés FON_k (voir définition 3.1). Ce modèle a été proposé par Scholtes *et al.* [75] en étant qualifié de modèle « mixte » ou « d'ordre multiple ». Dans le cadre de ce travail, nous appellerons ce modèle de réseau « réseau d'ordre fixe k » car bien qu'il intègre différents ordres, l'ordre maximal est fixé *a priori*.

Définition 3.1 (FON_k)**.** *Le réseau d'ordre fixe FON_k est le réseau d'ordre supérieur (définition 2.18) obtenu en prenant pour modèle séquentiel l'ensemble des sous-séquences s telles que $c(s) > 0$ et $|s| \leq k$.*

À l'instar du modèle SN_k, il faudra, pour construire ce réseau, énumérer les sous-séquences d'ordre inférieures ou égales à k présentes dans le jeu de données $\mathcal{S}$. Le modèle FON_k inclut donc l'information contenue dans SN_1 jusqu'à SN_k (voir l'exemple de la figure 3.2a), la différence entre les deux modèles est donc surtout un souci pratique.

Un problème évident est, dans ce cadre, le choix de la valeur du paramètre k. Un expert peut utiliser des hypothèses sur la dynamique du flux dans le système considéré. Mais, en l'absence de telles hypothèses, il est intéressant de déterminer l'ordre « optimal » en fonction d'un ensemble d'observations $\mathcal{S}$. Scholtes [75] propose une méthode pour déterminer cet ordre « optimal » reposant sur un test statistique de rapport de vraisemblance.

La vraisemblance $L(\text{FON}_k|\mathcal{S})$ du modèle FON_k sur le jeu de données $\mathcal{S}$ est la probabilité d'observer l'ensemble des séquences $s \in \mathcal{S}$ d'après le modèle *i.e.*

$$L(\text{FON}_k|\mathcal{S}) = \prod_{s\in\mathcal{S}} p^{\text{FON}_k}(s) \tag{3.1}$$

où $p^{\text{FON}_k}(s)$ correspond à la probabilité d'observer une séquence $s = \sigma_0 \dots \sigma_l$. Puisque FON_k est un modèle séquentiel, cette probabilité peut être exprimée de la façon suivante

$$p^{\text{FON}_k}(\sigma_0 \dots \sigma_l) = \prod_{i=1}^{l} p^{\text{FON}_k}(\sigma_i|\sigma_0 \dots \sigma_{i-1}) \tag{3.2}$$

Pour rappel, lorsque $i > (k+1)$, on n'utilisera que les k derniers états en mémoire *i.e.* $p^{\text{FON}_k}(\sigma_i|\sigma_0 \dots \sigma_{i-1}) = p^{\text{FON}_k}(\sigma_i|\sigma_{i-k} \dots \sigma_{i-1})$ (voir définition 2.7). L'éq. 3.2 correspond à la probabilité qu'un marcheur aléatoire génère le chemin d'état $\sigma_0 \dots \sigma_l$ dans FON_k.

La vraisemblance $L(\text{FON}_k|\mathcal{S})$ augmente mécaniquement avec l'ordre k. En effet, puisqu'elle est évaluée sur le même jeu de données que celui permettant de définir les probabilités de transition, FON_{k+1} sera toujours plus proche des données que FON_k. Toutefois, prendre un k très grand peut mener à un réseau qui n'est pas assez général et donc à un risque de sur-apprentissage. Le nombre de nœuds et de transitions supplémentaires à

définir augmentant avec l'ordre, la question est de savoir à quel moment le gain de vraisemblance est suffisamment « intéressant » par rapport à l'augmentation de la complexité du réseau. En partant d'un k donné, le « gain » de vraisemblance obtenu en augmentant l'ordre de 1 peut être évalué avec la statistique :

$$-2\log\left(\frac{L(\text{FON}_k|\mathcal{S})}{L(\text{FON}_{k+1}|\mathcal{S})}\right) \tag{3.3}$$

D'après le théorème de Wilks [82], dans le cas où il n'y a pas de gain significatif entre le modèle k et le modèle $k+1$, la distribution de cette statistique de test suit une loi du χ^2 dont le paramètre correspond au nombre de paramètres supplémentaires *i.e.* le nombre d'arcs supplémentaires à définir dans FON_{k+1} par rapport à FON_k. En théorie, le nombre de transitions dans FON_k est $\mathcal{O}(|\mathcal{A}|^{k+1})$. Toutefois, Scholtes propose de se limiter uniquement aux transitions possibles dans le système considéré. Par exemple, dans le cas de *AIR*, on ne compte pas l'ensemble des sous-séquences d'aéroports possibles mais celles pour lesquelles il existe vraiment des liaisons aériennes entre les aéroports consécutifs. Si $\mathcal{G}'$ correspond au graphe orienté où un arc (σ_1, σ_2) indique une transition possible entre les deux états, le nombre de probabilités de transition à définir dans FON_k est donné par le nombre de chemins de longueur $k+1$ dans $\mathcal{G}'$.

Puisque la distribution de la statistique (équation. 3.3) est connue, il est possible de calculer la p-valeur du test (*i.e.* la probabilité d'observer un gain de précision au moins aussi grand) et de la comparer à un seuil de significativité γ (fixé par Scholtes à 10^{-3}). En partant de $k=1$, la méthode consiste donc à tester chaque accroissement de l'ordre. L'ordre optimal k^* est soit 1 soit le dernier ordre pour lequel la p-valeur est inférieure à γ (*i.e.* pour lequel l'hypothèse que la précision n'augmente pas est rejetée). On notera FON_{opt} le réseau obtenu.

Un premier problème avec les modèles de réseaux d'ordre fixe est l'augmentation exponentielle théorique de la taille des réseaux avec le paramètre k. En effet, le nombre de sous-séquences d'ordre k est $\mathcal{O}(|\mathcal{A}|^k)$. Le réseau FON_k peut s'avérer difficile à construire et à analyser même pour des valeurs de k relativement faibles. Cependant, il est important de noter que ce maximum théorique est peu probable en pratique de la même façon que les réseaux « classiques » construits à partir de données réelles sont généralement *creux*. En effet, FON_k n'inclut que les sous-séquences effectivement observées dans les données. De plus, comme noté plus haut avec l'exemple de *AIR*, le nombre de séquences d'état effectivement réalisables dans un système donné peut être contraint et ne pas correspondre à l'ensemble des combinaisons d'états possibles.

Un second problème des réseaux d'ordre fixe est que, même en admettant que l'ordre k

permet de capturer toutes les dépendances séquentielles, l'ensemble des nœuds-mémoires de FON$_k$ correspond à un ensemble probablement plus grand. Dans l'exemple de la figure 3.1, la plupart des extensions observées de `d` apportent de l'information mais pas l'ensemble (*i.e.* `bd`). Au delà de cet exemple il est possible que les dépendances séquentielles représentent un sous-ensemble beaucoup plus petit par rapport aux sous-séquences existantes à l'ordre k.

3.2.2 Les réseaux d'ordre variable

Les deux problèmes inhérents aux modèles FON décrits plus haut sont liés à l'hypothèse qu'il existe un ordre k valable pour n'importe quelle sous-séquence. Les modèles dits d'ordre variable, notés VON, ne reposent pas sur cette hypothèse, et permettent d'obtenir des modèles plus parcimonieux (*i.e.* nécessitant moins de nœuds-mémoire) sans pour autant sacrifier la fidélité aux données d'entrée. Dans ces modèles, l'idée principale est de conserver uniquement les nœuds-mémoires qui sont considérés comme statistiquement « pertinents ».

EXEMPLE

Dans l'exemple de la figure 3.1, on a $P_{\texttt{d}} = [0, 0, \frac{1}{3}, 0, \frac{1}{3}, \frac{1}{3}]$ et $P_{\texttt{ad}} = [0, 0, 0, 0, 0, 1, 0]$. Autrement dit, le seul état possible après `d` en venant de `a` est l'état `e`. L'état `ad` sera considéré comme un contexte pertinent par rapport à `d` car il ajoute de l'information. Au contraire, on a $P_{\texttt{bd}} = [0, 0, \frac{3}{10}, 0, \frac{3}{10}, \frac{4}{10}]$, ce qui est très proche de $P_{\texttt{d}}$; la séquence `bd` ne semble donc pas ajouter beaucoup d'information. Ne conserver que les contextes pertinents conduit à obtenir le réseau d'ordre variable illustré en figure 3.1d

Nous proposons ici de réduire la différence entre chacun des modèles VON au calcul de la pertinence d'un contexte. Nous proposons donc une généralisation de l'algorithme de Saebi *et al.* [71] qui permet de construire n'importe quel modèle VON. Nous présentons ensuite la façon dont chacun des modèles VON de la littérature calcule la pertinence des contextes.

3.2.2.1 Algorithme générique de construction des Von

L'algorithme 1 est un cadre général utilisé dans les travaux de Saebi *et al.* [71] pour extraire les contextes pertinents d'un ensemble de données. Les contextes pertinents sont trouvés de manière récursive en tant qu'*extensions* des contextes pertinents d'ordres inférieurs (définition 2.3). Pour un ensemble de données $\mathcal{S}$ et un ensemble d'états $\mathcal{A}$, l'ensemble

final de contextes est défini comme $\mathcal{R} \coloneqq \bigcup_{\sigma \in \mathcal{A}} \texttt{Von}(\sigma, \sigma)$. Les fonctions `estPertinent` et `existeExtPertinente` dépendent du modèle utilisé.

Données : $\mathcal{S}$: ensemble de séquences sur un ensemble d'états $\mathcal{A}$
Entrées : s_c, s_v : le contexte *courant* et le dernier contexte *pertinent*
Résultat : R : ensemble de contextes pertinents

```
1 si existeExtPertinente(s_c, s_v) alors
2     pour σ ∈ A faire
3         si estPertinent(σs_c, s_v) alors
4             R ← R ∪ Von(σs_c, σs_c);
5         sinon
6             R ← R ∪ Von(σs_c, s_v);
7 retourner R ∪ préfixes(s_v)
```

Algorithme 1 : Algorithme générique `Von`

Le test `estPertinent`(s, s_v) (ligne 3 de l'algorithme 1) est réussi lorsque σs_c (extension de s_v) est jugé pertinent par rapport au dernier suffixe pertinent identifié s_v. Comme première étape, les contextes d'ordre 1 sont toujours considérés pertinents. L'algorithme effectue un parcours en profondeur des extensions *i.e.* pour un état $\sigma \in \mathcal{A}$, toutes les extensions de σ seront testées avant de passer à un autre état. En pratique, ce test revient à déterminer si les distributions $P_{\sigma s_c}$ et P_{s_v} doivent être considérées comme « suffisamment » différentes.

La fonction `existeExtPertinente` (ligne 1 de l'algorithme 1) est utilisée pour identifier les situations où aucune extension de s_v ne pourra être identifiée comme pertinente et, par conséquent, où la récursion doit être arrêtée. En tant que telle, cette fonction ne devrait pas avoir besoin de compter les sous-séquences $\sigma_1 s_c \sigma_2$ *i.e.* calculer le vecteur $c(\sigma_1 s_c)$. En pratique, cette fonction va donc beaucoup dépendre de la définition de `estPertinent` ; l'idée est d'essayer de déterminer des bornes supérieures à la différence entre les distributions $P_{\sigma s_c}$ et P_{s_v} à partir de l'information dont on dispose sur le contexte courant s_c. Si elle est bien conçue, elle devrait permettre l'utilisation de l'algorithme 1 sur de grands ensembles de données d'après Saebi *et al.* [71]. De plus, si on a des informations sur les dépendances maximales du jeu de données, il est également possible d'arrêter l'algorithme en imposant un ordre maximal.

Enfin, la fonction `préfixes` (ligne 7 de l'algorithme 1) renvoie l'ensemble des préfixes de s_v (y compris lui-même), comme précisé dans la définition 2.18. En effet, un marcheur aléatoire sur un HON ne peut atteindre le nœud-mémoire $s_1 s_2 \dots s_k$ que s'il existe un chemin $s_1 \to s_1 s_2 \to \cdots \to s_1 s_2 \dots s_k$. Par conséquent, tous les préfixes de s_v sont ajoutés au réseau, même si certains ne sont pas pertinents.

EXEMPLE

Prenons comme exemple l'appel à `Von(a, a)` ; nous allons tester si des extensions pertinentes de $s_v = s_c =$ `a` peuvent exister avec `existeExtPertinente(a, a)`. Si le test est `Faux` alors l'algorithme s'arrête sinon il continue en testant les extensions de `a`.
La première extension testée est `ba` et sa pertinence est calculée avec `estPertinent(ba, a)`. Si le test est `Vrai`, `ba` devient le dernier contexte pertinent. Sinon, on conserve $s_v =$ `a`.
Si `ba` n'est pas jugé pertinent, on pourra continuer à chercher d'autres extensions d'ordre plus élevé. Si le test `existeExtPertinente(ba, a)` est `Vrai` alors on testera des extensions d'ordre 3 par rapport à `a` *e.g.* `estPertinent(cba, a)`. Si, cette fois, la séquence `cba` est évaluée comme pertinente alors ses préfixes {`cba`, `cb`, `c`} seront inclus au réseau (notons que l'inclusion de `c` est redondante).

Comme suggéré plus haut, le coût algorithmique du calcul de `Von` vient du calcul du nombre d'occurrences des sous-séquences $c(.)$ (en admettant que le calcul de `estPertinent` et `existeExtPertinente` s'effectue en $\mathcal{O}(1)$ si les valeurs de $c(.)$ sont connues). L'approche directe consiste à parcourir l'ensemble des séquences $\mathcal{S}$ mais cette approche n'est pas efficace en pratique. Une façon d'optimiser ce calcul est de garder en mémoire les positions où le contexte courant s_c est observé. Il est alors possible de calculer $c(\sigma_1 s_c \sigma_2)$ en un temps $\mathcal{O}(c(s_c))$ en supposant que l'ensemble $\mathcal{S}$ tient entièrement en mémoire.

L'algorithme 1 a été proposé par Saebi *et al.* pour le calcul du modèle D_{KL}-Von présenté dans la section suivante. Xu *et al.* [84], une équipe proche de Saebi *et al.*, avait, à l'origine, proposé un algorithme n'impliquant pas la fonction `existeExtPertinente` et effectuant une recherche exhaustive de toutes les sous-séquences observées jusqu'à un ordre k donné en paramètre. Cette approche peut rendre le calcul de contextes pertinents assez long comparé à l'algorithme 1. Krieg *et al.* [50] (une autre équipe proche) ont toutefois proposé une amélioration de ce premier algorithme permettant de limiter les appels à la fonction `estPertinent`. Cette méthode utilise toujours le paramètre de l'ordre maximal.

3.2.2.2 Modèle D_{KL}-Von

Pour le modèle D_{KL}-Von, la pertinence est définie en utilisant la divergence de *Kullback-Leibler* (définition 2.9) et en utilisant une fonction de seuil définie par Xu *et al.* [84].

Définition 3.2. *Le réseau d'ordre variable D_{KL}-VON(λ) est obtenu en traitant la sous-séquence σs_c comme pertinente par rapport à s_v ssi :*

$$D_{KL}\left(P_{\sigma s_c}||P_{s_v}\right) > \frac{\lambda|\sigma s_c|}{\log_2(1+c(\sigma s_c))} \tag{3.4}$$

pour $\lambda > 0$.

La partie droite de l'équation 3.4, appelée « fonction de seuil », est croissante avec l'ordre mais décroissante avec l'ordre de grandeur du nombre d'occurrences de σs_c (l'ajout de 1 permet d'éviter un dénominateur nul dans les cas où $c(\sigma s_c) = 1$). Ainsi, une séquence sera plus facilement jugée pertinente si elle est souvent observée et qu'elle n'est pas trop longue. La fonction `estPertinent`$(\sigma s_c, s_v)$ renvoie donc `Vrai` si l'équation 3.4 est vérifiée.

Saebi *et al.* [71] suggèrent l'inclusion d'un facteur à la fonction de seuil noté λ dans la partie droite de l'équation 3.4. L'augmenter ou le diminuer permet ainsi de plus facilement (respectivement difficilement) juger une extension comme pertinente. Il va permettre d'ajuster la taille du réseau en fonction des besoins des applications. Notons toutefois qu'il est fixé à 1 par les auteurs pour leurs expérimentations et que son effet est peu discuté, les auteurs mettant en avant un modèle « sans paramètres ». Nous utiliserons cette variable afin de pouvoir comparer des réseaux de taille similaire dans la section 3.4.

EXEMPLE

Reprenons l'exemple de la figure 3.1 avec $P_{\texttt{d}} = \left[0, 0, \frac{1}{3}, 0, \frac{1}{3}, \frac{1}{3}\right]$. Nous voulons évaluer la pertinence de l'extension `bd` avec $P_{bd} = \left[0, 0, \frac{3}{10}, 0, \frac{3}{10}, \frac{4}{10}\right]$. Puisque $|\texttt{bd}| = 2$ et $c(\texttt{bd}) = 9$, nous avons donc

$$D_{KL}\left(P_{\texttt{bd}}||P_{\texttt{d}}\right) = 0,014$$
$$\frac{|\texttt{bd}|}{\log_2(1+c(\texttt{bd}))} = 0,6$$

En utilisant $\lambda = 1$, l'extension `bd` n'est pas pertinente dans le cadre du modèle D_{KL}-VON.

Afin de définir la fonction `existeExtPertinente`(s_c, s_v), les auteurs utilisent une borne supérieure à la divergence D_{KL} (voir équation 3.5).

$$\max_{\sigma\in\mathcal{A}}(D_{KL}(P_{\sigma s_c}||P_{s_v})) \leq -\log_2(\min_{\sigma\in\mathcal{A}}(P_{s_v}(\sigma))) \tag{3.5}$$

Intuitivement, la divergence est maximisée lorsque l'état le moins probable après la séquence s_v devient le seul état possible après la séquence σs_c. La fonction `existeExtPertinente` renvoie donc `Faux` si :

Acronymes littérature	Ref.	Dans ce document
First-Order Network (FON)	[84, 71]	FON_1
$M1/M2$	[70]	SN_1/SN_2
Multi-order Network $\bar{M}_K$	[75]	FON_k
Optimal-order $\bar{M}_{K_{opt}}$	[75]	FON_{opt}
HON (dans le cas général)	[84, 71]	VON
HON (Saebi *et al.*)	[71]	D_{KL}-VON

TABLE 3.1 – Équivalence des notations entre la littérature et ce travail.

$$-\log_2(\min_{\sigma\in\mathcal{A}}(P_{s_v}(\sigma))) \leq \frac{\lambda|s_c|}{\log_2(1+c(s_c))} \tag{3.6}$$

3.2.2.3 Définitions alternatives de la pertinence dans la littérature

L'algorithme 1 est générique et peut utiliser d'autres définitions de pertinence. Par exemple, en considérant une extension pertinente si son ordre est inférieur ou égal à un paramètre k alors le résultat correspondra à l'ensemble des sous-séquences du modèle FON_k.

Le modèle D_{KL}-VON est toutefois le seul à avoir été envisagé dans le cadre de l'analyse des réseaux d'ordre variable. C'est donc seulement à ce modèle et aux modèles d'ordre fixe que nous comparerons notre proposition MC-VON définie dans la section suivante. On peut toutefois noter que d'autres critères de pertinence, issus d'autres domaines, sont envisageables.

Borges *et al.* [11, 12], s'intéressent à la prédiction de traces d'utilisateurs sur le web. Ils utilisent pour cela un modèle markovien d'ordre supérieur et évaluent la capacité du modèle à correctement prédire le choix de lien suivant. Leur approche diffère de la construction de réseaux d'ordre supérieur car elle inclut un début et un fin de session. Dans ce cadre, le passage d'une page `a` à `b` ne sera pas équivalent s'il apparaît en début de parcours ou en fin de parcours. On pourrait toutefois adapter le critère qu'ils utilisent pour juger si une sous-séquence est pertinente. Ce critère équivaut dans notre cas à juger une sous-séquence σs_c comme *pertinente* par rapport à s_v s'il existe un σ' tel que :

$$|p(\sigma'|\sigma s_c) - p(\sigma'|s_v)| > \gamma \tag{3.7}$$

où $\gamma \in (0, 1)$ (défini à 0.1 par les auteurs).

3.3 Modèle MC-Von

L'un des principaux avantages de D_{KL}-VON par rapport à FON$_k$ est que la longueur des contextes est définie localement afin de s'adapter au mieux aux données. Le côté droit de l'équation 3.4 rend les contextes plus longs et peu observés plus difficiles à reconnaître comme pertinents.

Cependant, ce test correspond à un test *ad hoc* ou une « règle du pouce ». En effet, il est difficile d'évaluer la relation entre D_{KL}, exprimée en bits, et cette fonction de seuil. On peut affirmer que la définition de D_{KL}-VON cache en fait un choix arbitraire d'« échelle » fait par les auteurs (avec $\lambda = 1$). Par conséquent, l'« absence de paramètres » de D_{KL}-VON est, à notre avis, au moins discutable.

Par ailleurs, bien que l'absence de paramètre ait des vertus, la construction de réseaux d'ordre supérieur est un compromis entre fidélité aux données et taille du modèle. Nous pensons donc qu'il est légitime de disposer d'un paramètre permettant de naviguer dans l'espace des solutions. Toutefois, afin de donner une interprétation à ce paramètre, nous proposons de construire un modèle VON basé sur un test statistique que nous allons définir dans la section 3.3.1. L'idée principale n'est pas de se demander si la quantité D_{KL} est assez grande mais si elle semble *improbable*. Notre modèle n'est toutefois pas directement calculable en pratique et nous proposons une approximation dans la section 3.3.2.

3.3.1 Définition de MC-Von

Pour définir le modèle MC-VON, nous redéfinissons les fonctions `estPertinent`$(\sigma s_c, s_v)$ et `existeExtPertinente`(s_c, s_v) de l'algorithme 1. De la même façon que pour D_{KL}-VON, nous ne fixons pas d'ordre maximum ou de nombre d'occurrences minimum pour les contextes mais nous allons redéfinir la notion de contexte pertinent. Nous nous plaçons pour cela dans le paradigme du test d'hypothèse. Nous utilisons toujours la quantité $D_{KL}\left(P_{\sigma s_c} || P_{s_v}\right)$ comme proposé par Saebi *et al.* [71] pour évaluer la différence entre les distributions mais nous allons nous intéresser à sa distribution dans le cas de tirages aléatoires.

En effet, si σs_c n'est pas un contexte pertinent, alors les états suivant le contexte $C_{\sigma s_c}$ devraient se comporter comme un tirage de $c(\sigma s_c)$ éléments parmi C_{s_v} sans remplacement. Autrement dit, il n'y a, dans ce cas, pas d'état (ou de sous-ensemble d'états) choisi préférentiellement dans C_{s_v}. On peut, dans ce cadre, voir $C_{\sigma s_c}$ comme une variable aléatoire issue d'une distribution hypergéométrique multivariée notée $\mathcal{MH}(C_{s_v}, c(\sigma s_c))$. Par conséquent, nous déciderons que σs_c est un contexte pertinent lorsque $C_{\sigma s_c}$ ne se comporte pas comme un tirage aléatoire, c'est-à-dire lorsque nous pouvons rejeter l'hypothèse nulle H_0

en faveur de l'alternative H_1 avec

$$H_0 : C_{\sigma s_c} \sim \mathcal{MH}(C_{s_v}, c(\sigma s_c)) \quad \text{vs.} \quad H_1 : C_{\sigma s_c} \nsim \mathcal{MH}(C_{s_v}, c(\sigma s_c)). \tag{3.8}$$

Afin de prendre une décision quant à H_0, nous utilisons D_{KL} comme statistique de test. On s'intéresse alors à sa distribution sous l'hypothèse H_0.

Définition 3.3. *Le réseau d'ordre variable MC-*Von(α) *avec le paramètre de* seuil $\alpha \in (0, 1)$ *est obtenu en traitant la sous-séquence σs_c comme pertinente par rapport à s_v ssi*

$$D_{KL}\left(P_{\sigma s_c} || P_{s_v}\right) \geq q_{1-\alpha}\left(c(\sigma s_c), s_v\right) \tag{3.9}$$

où $q_{1-\alpha}\left(c(\sigma s_c), s_v\right)$ est le $(1-\alpha)$-ème quantile de la distribution de $D_{KL}\left(P_D || P_{s_v}\right)$ où D est un tirage aléatoire de $\mathcal{MH}(C_{s_v}, c(\sigma s_c))$. L'équation 3.9 est équivalente à tester $p \leq \alpha$ où

$$p \coloneqq \mathbb{P}\left(D_{KL}\left(P_D || P_{s_v}\right) \geq D_{KL}\left(P_{\sigma s_c} || P_{s_v}\right)\right) \tag{3.10}$$

est la p-valeur du test.

EXEMPLE

Supposons que nous ayons une sous-séquence d'ordre 1, s_v, avec

$$C_{s_v} = (1, 2, 5, 0, \ldots)$$

et que nous voulions évaluer la pertinence de l'extension σs_v (d'ordre 2) avec

$$C_{\sigma s_v} = (1, 0, 0, 0, \ldots)$$

Nous avons $D_{KL}\left(P_{\sigma s_v} || P_{s_v}\right) = -\log_2(1/8) = 3$. Comme il s'agit de la D_{KL} la plus élevée possible avec $c(\sigma s_v) = 1$, la probabilité qu'un tirage de $\mathcal{MH}(C_{s_v}, 1)$ ait une divergence supérieure ou égale est la probabilité de tirer $C_{\sigma s_c}$ *i.e.* $p = \frac{1}{8}$. En prenant un seuil standard de $\alpha = 10^{-3}$, nous accepterions H_0 et déclarerions que cette extension n'est pas pertinente.
En revanche, pour D_{KL}-Von(1), la fonction seuil de l'équation 3.4 est égale à $\frac{2}{\log_2(2)} = 2$. L'extension σs_v serait ici considérée comme pertinente.

Le seuil $\alpha \in (0, 1)$ du test nous permet de choisir à quel point nous voulons que le tirage $C_{\sigma s_c}$ soit « surprenant » afin de considérer σs_c comme un contexte pertinent. Il s'agit également d'une limite supérieure pour la probabilité qu'un contexte soit considéré

comme pertinent alors qu'il ne l'est pas. Ainsi, une valeur de α proche de 0 conduit à souvent accepter H_0 et donc à juger que σs_c n'apporte pas beaucoup d'informations. Au contraire, augmenter la valeur de α va permettre de retenir davantage de nœuds-mémoire dans le réseau. À l'instar des travaux de Scholtes [75], nous nous baserons pour les expérimentations sur une valeur de $\alpha = 10^{-3}$.

Pour la fonction `existeExtPertinente`(s_c, s_v) présentée dans la section 3.2.2.1, nous utilisons une borne inférieure sur la p-valeur. En effet, il y a au plus

$$z = \begin{pmatrix} c(s_v) \\ \min\left(\frac{c(s_v)}{2}, c(s_c)\right) \end{pmatrix} \tag{3.11}$$

tirages de $\mathcal{MH}(C_{s_v}, c(\sigma s_c))$ pour tout $\sigma \in \mathcal{A}$. Par conséquent, si $z^{-1} > \alpha$, aucune extension possible de s_c ne peut être considérée comme pertinente. Cette situation se produira dans les cas où le nombre d'occurrences de la séquence s_v et/ou s_c sont trop faibles.

EXEMPLE

Supposons que $\alpha = 0.001$, $c(s_v) = 10$ et $c(s_c) = 3$, on sait que $\forall \sigma \in \mathcal{A}$, on aura $c(\sigma s_c) \leq 3$. Il y a donc au plus $\begin{pmatrix} 10 \\ 3 \end{pmatrix} = 120$ tirages possibles de $\mathcal{MH}(C_{s_v}, c(\sigma s_c))$. Même si $\sigma^* s_c$ est une extension pour laquelle $D_{KL}\left(C_{\sigma^* s_c} || P_{s_v}\right)$ est minimum, la probabilité de ce tirage sera au mieux de $\frac{1}{120} > \alpha$.

3.3.2 Calcul de MC-Von en pratique

Les valeurs de p ou $q_{1-\alpha}\left(c(\sigma s_c), s_v\right)$ peuvent être difficiles à calculer, en particulier si $c(\sigma s_c)$ n'est ni petit ni proche de $c(s_v)$. En effet, il y a $\begin{pmatrix} c(s_v) \\ c(\sigma s_v) \end{pmatrix}$ tirages possibles. Bien que beaucoup soient équivalents à une permutation près du point de vue de la mesure D_{KL}, il n'est pas possible de calculer la distribution exacte de D_{KL} en temps polynomial. Par ailleurs, des approximations de la distribution de D_{KL}, notamment par des lois Gamma [16], peinent à capturer la forme de la queue de la distribution, or c'est précisément la région qui nous intéresse ici (*i.e.* la p-valeur pour un seuil α petit).

C'est pourquoi nous proposons de nous baser sur un algorithme de Monte-Carlo [51] pour estimer la p-valeur et décider si l'éq. 3.9 est valable ou non. Cette méthode consiste à effectuer M tirages aléatoires de variables indépendantes $\{D_i, 1 \leq i \leq M\}$ de la loi $\mathcal{MH}(C_{s_v}, c(\sigma s_c))$. L'estimation de p est alors réalisée en comptant le nombre de tirages

aboutissant à une D_{KL} supérieure à la valeur observée *i.e.* on prend $\widehat{p} = S_M/M$ avec

$$S_M = \sum_{i=1}^{M} \mathbb{I}\left\{D_{KL}\left(P_{D_i}||P_{s_v}\right) \geq D_{KL}\left(P_{\sigma s_c}||P_{s_v}\right)\right\} \tag{3.12}$$

Notons que cette quantité suit une distribution binomiale de taille M et de probabilité p, *i.e.*

$$\mathbb{P}(S_M = k) = \binom{M}{k} p^k (1-p)^{M-k} =: b(n, p, k) \tag{3.13}$$

Le choix de M affectera la précision de la décision quant à H_0, en particulier si p est proche de α. Au contraire, si la conclusion est plus évidente, c'est-à-dire $p \ll \alpha$ ou $p \gg \alpha$, nous aurions pu choisir une valeur plus petite pour M afin d'obtenir une précision raisonnable. Des méthodes qui adaptent le nombre de tirages à la distance entre p et α ont été proposées par exemple par Gandy *et al.* [30] et par Ding *et al.* [25]. Le but de ces travaux est plus précisément de contrôler le risque de ré-échantillonnage défini par

$$RR_p(\widehat{p}) = \begin{cases} \mathbb{P}_p\left(\widehat{p} > \alpha\right) & \text{if } p \leq \alpha \\ \mathbb{P}_p\left(\widehat{p} \leq \alpha\right) & \text{if } p > \alpha. \end{cases} \tag{3.14}$$

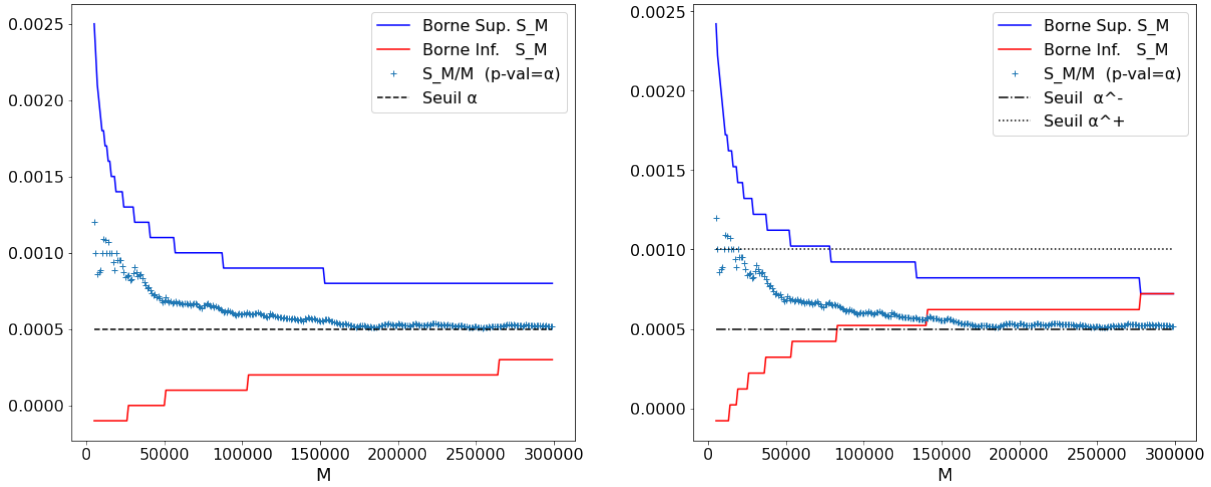
Ce risque de ré-échantillonnage mesure la probabilité de prendre une mauvaise décision concernant la pertinence de σs_c (3.9). Pour un $\epsilon > 0$ donné, Gandy [30] propose une procédure qui garantit que $RR_p \leq \epsilon$. Celle-ci consiste à arrêter le tirage lorsque la valeur de S_M semble trop grande ou trop petite par rapport à ce qu'on pourrait s'attendre dans le cas d'une variable aléatoire binomiale après M tirages *i.e.* lorsque

$$(M+1)b(M, \alpha, S_M) < \epsilon \tag{3.15}$$

L'équation 3.15 définit donc deux bornes (inférieures et supérieures) pour la valeur de S_M dépendant de M (voir Fig. 3.3a). Néanmoins, il n'y a pas de limite au nombre de tirages nécessaires et la procédure peut ne pas se terminer si p est proche de α, c'est-à-dire quand $D_{KL}\left(P_{\sigma s_c}||P_{s_v}\right) = q_{1-\alpha}\left(c(\sigma s_c), s_v\right)$. Un nombre maximum d'itérations doit être choisi et le risque de ré-échantillonnage n'est donc pas réellement contrôlé.

Nous proposons donc une approche légèrement différente adaptée à notre problème. Notons que le RR_p est symétrique *i.e.* le fait de surestimer ou de sous-estimer p sont équivalents en termes de risque. Dans notre cas, on peut considérer que surestimer la valeur de p est plus dommageable car on préférera ajouter un nœud-mémoire peu important que manquer une dépendance séquentielle pertinente.

Nous définissons ainsi un risque « asymétrique » : nous divisons α en une valeur inférieure α^- et une valeur supérieure α^+ afin que le nombre d'itérations soit toujours fini.



(a) Classique (b) Asymétrique

FIGURE 3.3 – Exemple de test selon le nombre de tirages M (abscisses) dans le cas où la p-valeur (inconnue normalement) correspond au seuil de confiance $\alpha = 0.0005$. L'estimation S_M/M (croix bleue) fluctue autour de celle-ci. Le test s'arrête si la borne supérieure (bleu) ou inférieure (rouge) de l'intervalle de confiance de S_M/M de longueur $(1-\epsilon)$ est atteinte. Dans l'approche « classique » [25] (3.3a), il est peu probable de s'arrêter sans un nombre maximum d'itérations. Avec notre méthode (3.3b), la borne inférieure est calculée avec un seuil $\alpha^+ = 0.001$. On peut garantir, avec ces paramètres, un maximum d'environ 277K tirages (valeur de M à partir de laquelle les deux bornes se rejoignent à la valeur $\alpha^* \approx 0.0007$).

Le coût de ce nombre fini d'itérations vient du fait d'accepter des séquences légèrement moins pertinentes (*i.e.* quand $\alpha^- < p \leq \alpha^+$) plutôt qu'en manquant des séquences qui sont pertinentes (*i.e.* quand $p \leq \alpha^-$) et en définissant

$$\widetilde{RR}_p(\hat{p}) = \begin{cases} \mathbb{P}_p\left(\hat{p} > \alpha^*\right) & \text{if } p \leq \alpha^- \\ 0 & \text{if } p \in]\alpha^-, \alpha^+] \\ \mathbb{P}_p\left(\hat{p} \leq \alpha^*\right) & \text{if } p > \alpha^+ \end{cases} \tag{3.16}$$

où $\alpha^* \in]\alpha^-, \alpha^+[$ est une valeur critique pour $\hat{p}$ telle que nous rejetterons H_0 ssi $\hat{p} < \alpha^*$ (la valeur choisie pour α^* est discutée ci-dessous). Notons que, dans le cas où $p \in]\alpha^-, \alpha^+]$, la valeur de $\widetilde{RR}_p(\hat{p}) = 0$ est choisie arbitrairement pour être inférieure à ϵ. Le risque n'est pas contrôlé dans ce cas.

L'algorithme 2 est celui utilisé pour estimer la p-valeur et donc définir la fonction `estPertinent`$(\sigma s_c, s_v)$ pour MC-Von. Cette estimation garantit ainsi $sup_{p\in[0,1]}\widetilde{RR}_p(\hat{p}) \leq \epsilon$.

Théorème 3.1. *L'algorithme 2 s'arrête après un nombre fini de tirages.*

Démonstration. La fin de l'algorithme vient du fait que la fonction $x \mapsto (n+1)b(n, x, n\hat{p})$ est la densité de la loi bêta de paramètres $n\hat{p}+1$ et $n(1-\hat{p})$ et tend vers une mesure de

Données : $D_{KL,obs}$: Divergence observée
Entrées : α^-, α^+, ϵ : seuils et limite pour le risque de ré-échantillonnage
Résultat : $\widehat{p}$: p-valeur estimée
$S = 0$; $n = 0$
tant que $(b(n, \alpha^-, S) > \epsilon/(n+1)) \wedge (b(n, \alpha^+, S) > \epsilon/(n+1))$ **faire**
  $D \sim \mathcal{MH}(C_{s_v}, c(\sigma s_c))$
  si $D_{KL}\left(P_D || P_{s_v}\right) \geq D_{KL,obs}$ **alors**
    $S = S + 1$
  $n = n + 1$
retourner $\widehat{p} = S/n$

Algorithme 2 : Algorithme de décision de MC-Von

Dirac en p lorsque $n \to \infty$. Par conséquent, au moins l'une des deux valeurs $b(n, \alpha^-, S_n)$ et $b(n, \alpha^+, S_n)$ doit tendre vers zéro si $\alpha^- < \alpha^+$. □

La valeur de α^* est la valeur pour laquelle nous allons comparer la p-valeur estimée par l'algorithme 2. Plusieurs choix sont possibles, nous prenons ici la valeur telle que $b(n, \alpha^-, n\alpha^*) = b(n, \alpha^+, n\alpha^*)$ *i.e.*

$$\alpha^* = 1 - \frac{\log\left(\alpha^+/\alpha^-\right)}{\log\left(\dfrac{\alpha^+/(1-\alpha^+)}{\alpha^-/(1-\alpha^-)}\right)} \tag{3.17}$$

Cette valeur correspond à la valeur p qui nécessitera le plus grand nombre de tirages en moyenne d'après les paramètres choisis. En pratique, elle est proche de $\frac{\alpha^- + \alpha^+}{2}$. Sur la figure 3.3b, elle correspond à l'ordonnée à laquelle les bornes inférieure et supérieure se rejoignent.

3.4 Expériences

Dans cette section, nous comparons les modèles de la littérature ainsi que le MC-Von, en terme de taille et de précision. Ensuite, nous étudions les contextes pertinents en fonction du modèle de réseau. Dans la suite, nous utiliserons les données présentées dans la section 2.5.

3.4.1 Précision et taille du réseau

Pour construire les réseaux MC-Von, nous utilisons une valeur standard pour le seuil de confiance $\alpha^- = 10^{-3}$ avec $\alpha^+ = \alpha^- + 2.10^{-3}$ pour contrôler un risque $\epsilon = 0,05$. Cela signifie que nous prenons la bonne décision pour les valeurs p hors $)\alpha^-, \alpha^+($ avec une probabilité d'au moins $1 - \epsilon$. Nous présentons les résultats pour le modèle D_{KL}-Von car il s'agit du principal autre modèle de réseau d'ordre variable existant dans la littérature.

TABLE 3.2 – Comparaison entre les modèles HON sur les quatre jeux de données

Jeu de données	**Réseau**	$\|V\|$	**Ordre**	**Acc** $\pm 2sd$
PORTS	FON$_1$	909	1	13.71 ± 0.73
	FON$_2$	9,437	2	31.73 ± 1.38
	D_{KL}-VON(1.95)	9,559	6	38.56 ± 1.63
	D_{KL}-VON(1)	18K	8	46.48 ± 1.89
	MC-VON(0.001)	9,553	16	42.93 ± 2.22
	MC-VON(0.05)	18K	27	48.17 ± 2.23
AIR	FON$_1$	175	1	19.48 ± 0.09
	FON$_2$	1,716	2	27.44 ± 0.10
	D_{KL}-VON(2.85)	28K	6	36.50 ± 0.15
	D_{KL}-VON(1)	58K	6	39.37 ± 0.19
	MC-VON(0.001)	28K	6	37.11 ± 0.15
	MC-VON(0.29)	58K	6	39.19 ± 0.20
MSNBC	FON$_1$	17	1	13.82 ± 0.07
	FON$_3$	4,061	3	22.18 ± 0.16
	D_{KL}-VON(1.585)	5,774	8	22.04 ± 0.15
	D_{KL}-VON(1)	28K	11	22.29 ± 0.17
	MC-VON(0.001)	5,771	122	22.44 ± 0.17
	MC-VON(0.027)	28K	145	22.43 ± 0.16
WIKI	FON$_1$	100	1	21.48 ± 0.65
	D_{KL}-VON(3.39)	306	4	21.87 ± 0.67
	D_{KL}-VON(1)	2,260	4	23.29 ± 0.64
	MC-VON(0.001)	304	4	22.85 ± 0.65
	MC-VON(0.35)	2,257	12	23.39 ± 0.70

Afin de comparer les contextes retenus par chaque approche en termes de précision, nous déterminons également λ^* tel que D_{KL}-VON(λ^*) contienne un nombre de nœuds équivalent à MC-VON. De même, nous déterminons le α_*^- tel que MC-VON(α_*^-) contienne un nombre de nœuds équivalent à D_{KL}-VON(1), les autres paramètres étant égaux. Nous incluons également les résultats obtenus avec le réseau FON$_1$ et le réseau FON avec l'ordre optimal, présenté dans la section 3.2.

Nous étudions ici la différence entre les réseaux HON de la littérature et la possibilité d'obtenir une précision meilleure ou similaire avec un modèle plus petit en utilisant MC-VON. Le tableau 3.2 fait état des résultats pour chaque réseau construit sur les quatre ensembles de données en utilisant l'ensemble $\mathcal{S}$. La taille des réseaux est représentée par le nombre de nœuds $|\mathcal{V}|$ dans les réseaux. L'ordre correspond à l'ordre maximal du réseau. La dernière colonne fournit la *précision* (équation 3.18) pour chacun des modèles. Pour déterminer celle-ci, l'ensemble $\mathcal{S}$ est divisé en deux parties : 90% pour la construction du réseau et 10% pour calculer la précision (ensemble de test). Nous l'évaluons en considérant

la probabilité moyenne d'identifier les bons états dans $\mathcal{S}_T$:

$$Acc(\mathcal{R}, \mathcal{S}_T) \coloneqq \frac{100}{|\mathcal{S}_T|} \sum_{s \in \mathcal{S}_T} \frac{1}{|s|-1} \sum_{i=1}^{|s|-1} p^{\mathcal{R}}(s_{i+1}|s_1 s_2 \dots s_i) \tag{3.18}$$

où $\mathcal{R}$ correspond au modèle séquentiel choisi (ensemble des contextes retenus selon les méthodes) (définition 2.7). Cette approche est similaire à des scores de type *log-loss* utilisés dans le domaine de la prédiction de séquences [6]. Nous travaillons ici avec une moyenne plutôt qu'un produit car, à la différence des méthodes la prédiction de séquences, nous pouvons avoir une transition de probabilité nulle (correspondant à l'absence de lien dans le réseau). Notons par ailleurs que Xu *et al.* [84] utilise un score similaire mais uniquement calculé en prenant la fin (*i.e.* les trois derniers états) de chaque séquence. Les résultats de précision correspondent à des moyennes sur 50 itérations.

L'augmentation de la précision entre Fon_1 et les autres modèles justifie l'utilisation de modèles d'ordre supérieur. Par exemple, nous pouvons prédire correctement près de la moitié des ports visités par les navires dans l'ensemble de données *PORTS* en utilisant D_{KL}-Von ou MC-Von. Ce score tombe à 13% pour le réseau régulier Fon_1. Cette différence est moins importante pour *WIKI*. Notons par ailleurs que l'ordre optimal de Fon est de 1 pour ce jeu de données[1]. Nous comparons maintenant les réseaux D_{KL}-Von et MC-Von. Pour un ordre donné, l'ensemble des contextes jugés pertinents est différent, même si les paramètres sont choisis de manière à obtenir des ensembles de taille similaire. Cela suggère que la différence entre les deux méthodes n'est pas seulement une question de paramétrage.

Les réseaux MC-Von ont un ordre maximal plus important qui peut être très grand. Ceci est normal puisque le critère utilisé ne pénalise pas intrinsèquement les contextes larges. Les écarts les plus importants sont obtenus avec *MSNBC* et *PORTS*. Notons toutefois que de tels contextes sont rares ; la grande majorité des nœuds-mémoire sont d'ordre 2 ou 3. Nous verrons plus en détail ces différences dans la section suivante. Lorsque l'on compare des réseaux d'ordre variable de taille similaire, MC-Von semble correspondre à D_{KL}-Von en termes de précision. Pour *PORTS* ou *WIKI*, il est nettement plus performant. Pour *MSNBC* et *AIR*, les résultats sont plus proches. Cela confirme l'idée que le critère utilisé pour MC-Von est une bonne alternative aux autres modèles de la littérature pour créer des réseaux parcimonieux.

1. Dans [75], l'ordre optimal pour le même jeu de données est de 2. Les résultats que nous présentons ici sont toutefois conformes avec ceux obtenus avec la nouvelle version du code des auteurs (`www.pathpy.net`, v2.2.0)

3.4.2 Comparaison entre les contextes pertinents des modèles

Puisque les modèles produisent des résultats de précision différents pour une taille donnée, on sait de fait que D_{KL}-VOM et MC-VOM prennent des décisions différentes sur la pertinence des contextes.

Nous cherchons ici à savoir

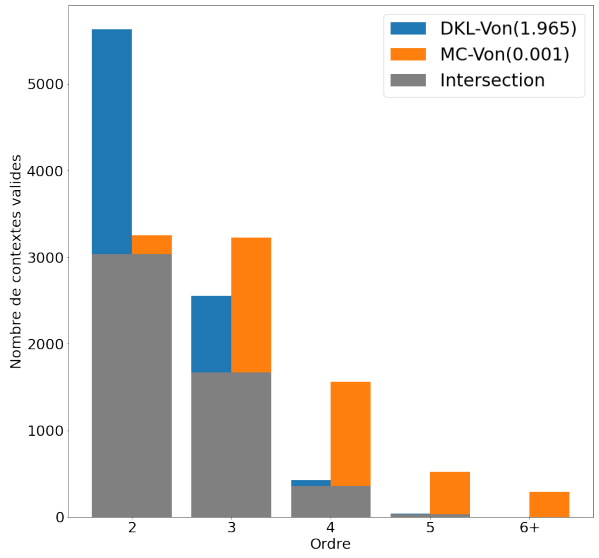
1. Comment se répartissent les contextes en fonction de l'ordre ?
2. À quel point les contextes pertinents sont communs entre les modèles ?
3. Comment se répartissent les contextes en fonction de leur fréquence (leur nombre d'occurrences) dans les jeux de données en entrée ?

La figure 3.4 permet de répondre à la première et deuxième question et la figure 3.5 à la troisième. Les conclusions que l'on peut tirer de ces graphiques valent que l'on considère la paire de réseau de plus petite taille (D_{KL}-VOM(λ^*), MC-VOM(0.001)) ou de plus grande taille (D_{KL}-VOM(1), MC-VOM(α^*)). Nous ne ferons donc pas cette distinction par la suite.

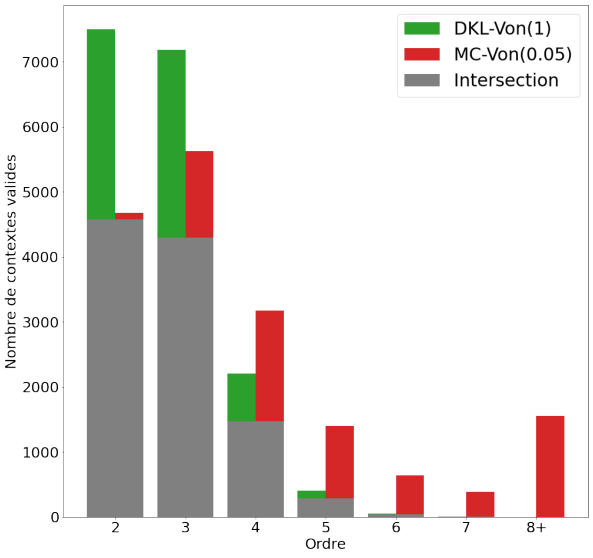
Pour la première question, la majorité des contextes pertinents des deux modèles sont des contextes d'ordre bas (*i.e.* soit d'ordre 2, 3 ou 4 selon le jeu de données). Dans le cas de MC-VOM, on retrouve quelques contextes d'ordre élevé voire un nombre non négligeable dans le cas de *MSNBC*. Toutefois, ceux-ci sont généralement rares.

Pour la deuxième question, il apparaît que les deux modèles ont un socle de contextes pertinents en commun plus ou moins important selon le jeu de données. Pour *AIR*, Les deux modèles sont très similaires à la différence de *MSNBC* où les modèles ont des comportements plus différents. Il serait compliqué d'expliquer cette différence. Mais on peut conclure que cette différence entre les deux approches n'est probablement pas juste une question de choix des paramètres.

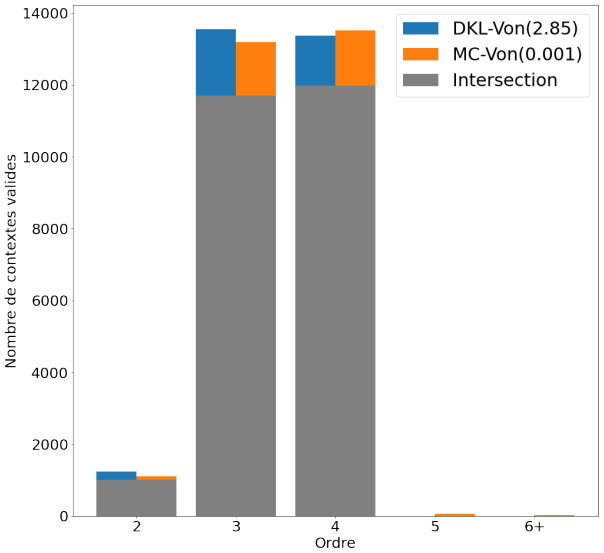
Enfin, pour la troisième question, il apparaît qu'en moyenne les contextes pertinents de D_{KL}-VOM ont un nombre plus faible d'occurrences. C'est particulièrement le cas pour les ordres bas. Effectivement, dans ce cas, les contextes peu fréquents peuvent avoir une valeur D_{KL} importante qui dépasse facilement le seuil de pertinence. L'effet inverse est aussi vrai ; les contextes d'ordre élevé doivent avoir une haute fréquence d'apparition pour être acceptés. Rappelons qu'il existe un risque de sur-apprentissage si les contextes sont peu rencontrés dans les jeux de données. Le fait que D_{KL}-VOM juge plus facilement les contextes peu fréquents explique peut-être la plus faible précision de ce modèle par rapport à MC-VOM.



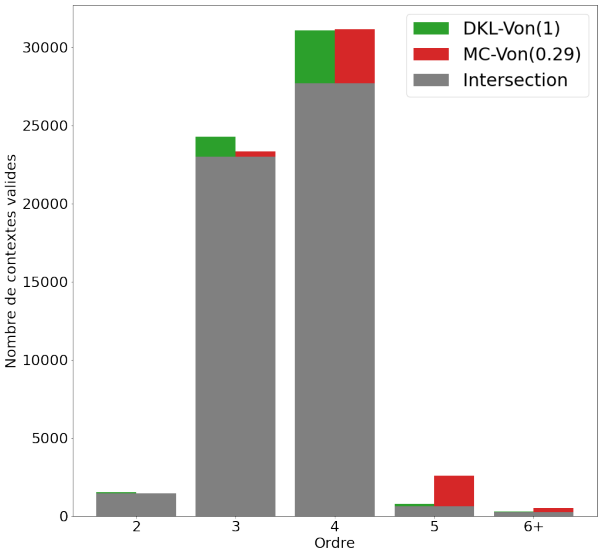
(a) *PORTS*



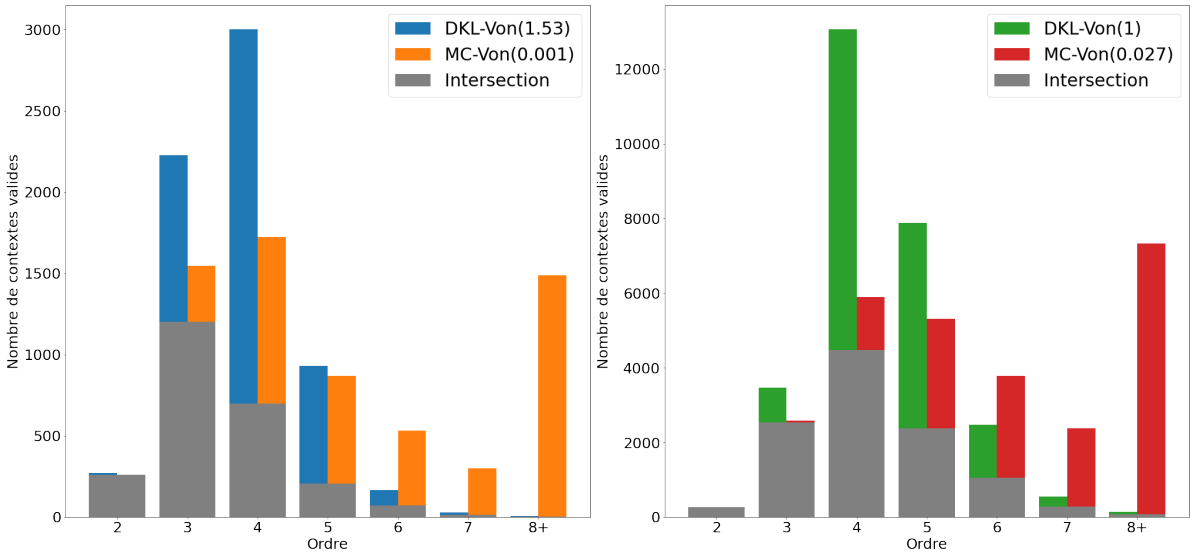
(b) *PORTS*



(c) *AIR*

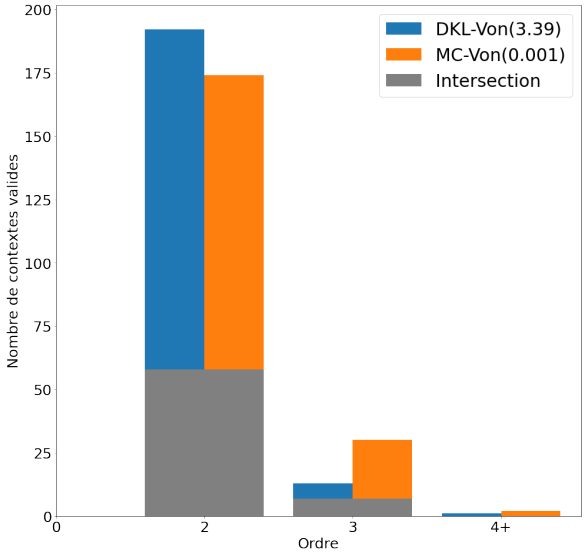


(d) *AIR*

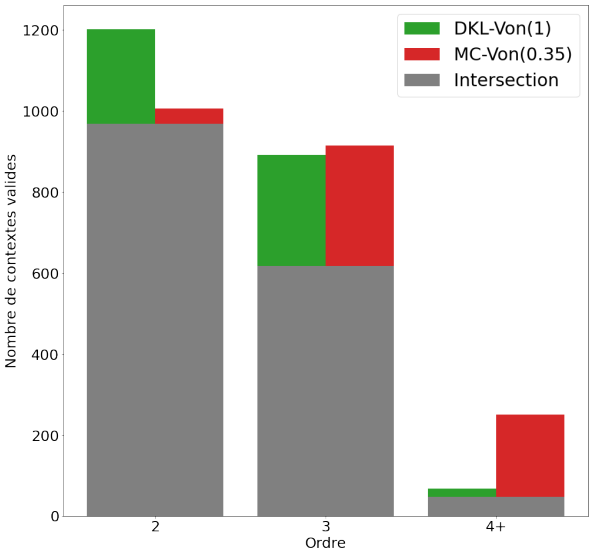


(e) *MSNBC*

(f) *MSNBC*

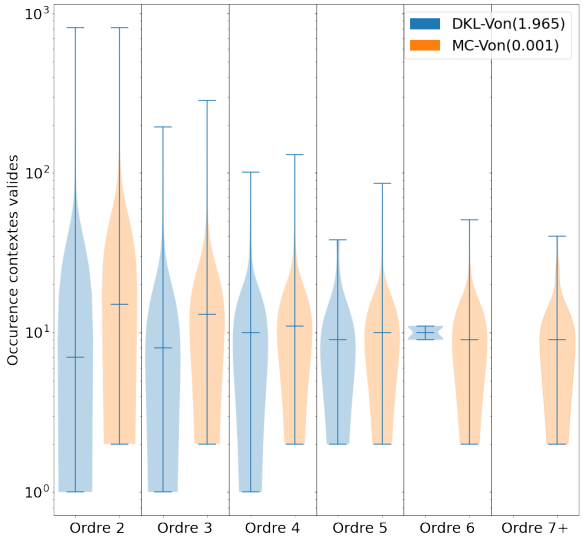


(g) *WIKI*

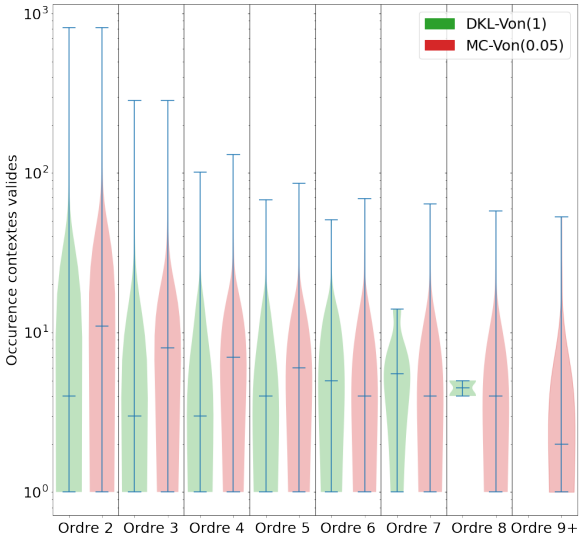


(h) *WIKI*

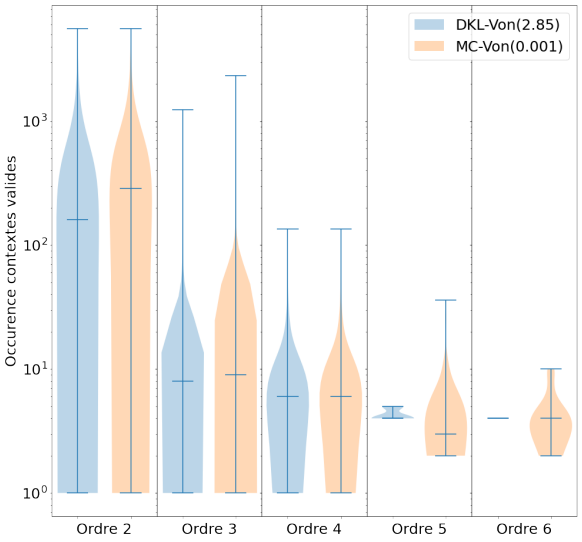
FIGURE 3.4 – Comparaison du nombre de contextes pertinents en fonction de l'ordre entre D_{KL}-VON et MC-VON. On compare les modèles de taille similaire, donc D_{KL}-VON(1)/MC-VON(α) (première colonne) et D_{KL}-VON(α)/MC-VON(0.001) (seconde colonne). La partie grise représente les nombres de contextes communs entre les deux modèles. Les barres colorées représentent le nombre de contextes restants du modèle. On représente les ordres en commun entre les deux modèles. Le dernier ordre représente la somme cumulée du reste des contextes du modèle qui dépasse l'ordre maximum en commun.



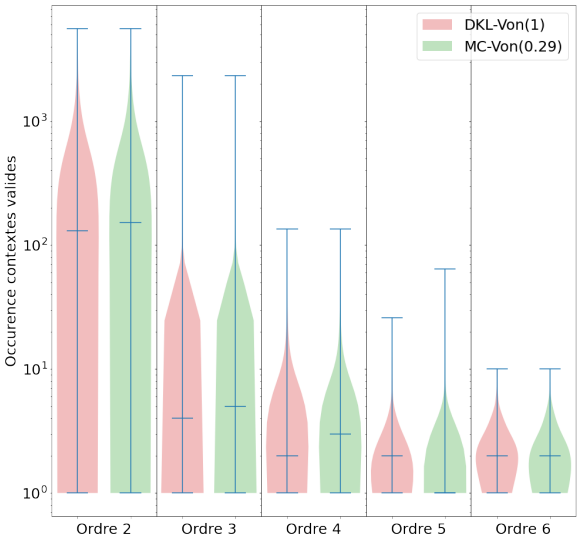
(a) *PORTS*



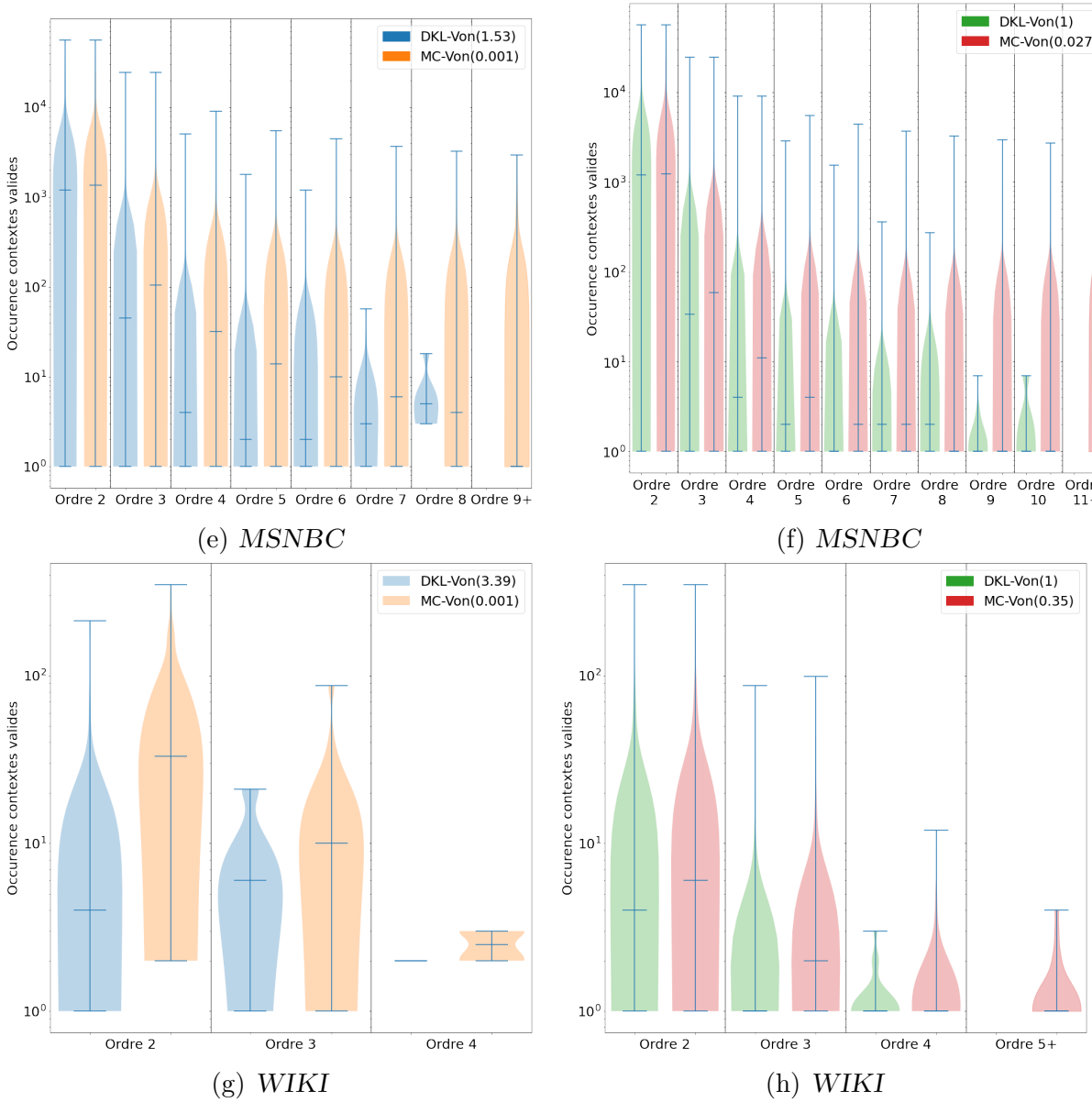
(b) *PORTS*



(c) *AIR*



(d) *AIR*



(e) *MSNBC* (f) *MSNBC*

(g) *WIKI* (h) *WIKI*

FIGURE 3.5 – Comparaison des occurrences des contextes pertinents en fonction de l'ordre entre D_{KL}-VON et MC-VON. On compare les modèles de taille similaire, donc D_{KL}-VON(1)/MC-VON(α) (première colonne) et D_{KL}-VON(α)/MC-VON(0.001) (seconde colonne). On représente les ordres en commun entre les deux modèles. Le dernier ordre représente la somme cumulée du reste des occurrences du modèle qui dépasse l'ordre maximum en commun. Sur le graphique, on représente le minimum (en bas), le maximum (en haut) et la barre du milieu représente la médiane.

3.5 Conclusion et Discussion

Dans ce chapitre, nous avons fait un panorama des différentes stratégies de sélection des nœuds-mémoires à intégrer dans un réseau d'ordre supérieur. Nous nous sommes concentrés sur les modèles d'ordre variable, qui permettent d'obtenir des réseaux plus

parcimonieux sans perdre voire en améliorant la précision par rapport aux réseaux d'ordre fixe. Pour les construire, nous définissons un algorithme générique qui réduit la différence entre les modèles au choix de la définition de pertinence. Nous proposons également un modèle d'ordre variable (MC-Von) alternatif au principal modèle de la littérature (D_{KL}-Von).

Au vu des résultats du modèle MC-Von, nous soutenons que ce modèle est une alternative viable au modèle D_{KL}-Von. Comme on l'a vu, les modèles peuvent être très différents en termes de nœuds-mémoire retenus. Il est important de noter que ces différences ne garantissent pas des résultats d'algorithmes de fouille différents. Dans le chapitre suivant, nous étudions l'impact du choix du modèle sur une tâche importante d'analyse de réseau : le calcul de mesure de centralité.

Il est toutefois important de soulever les limitations du modèle que nous proposons, à savoir le choix du paramètre et le temps de calcul discutés dans la section 3.5.1. Une limite plus générale concerne les choix implicites liés de l'algorithme 1 de recherche de contextes pertinents que nous discuterons en section 3.5.2.

3.5.1 Limites du modèle MC-Von

Le modèle D_{KL}-Von auquel nous comparons MC-Von est considéré comme *sans-paramètre*. Cependant comme soulevé dans la section 3.3.1, l'absence de paramètre du modèle est discutable et le seuil choisi est un choix arbitraire de la part des auteurs. De ce fait, nous avons proposé un modèle dont le paramètre est explicable, l'idée étant de quantifier l'improbabilité d'une extension.

Cependant, déterminer automatiquement le paramètre « optimal » est une piste qui pourrait être empruntée. Une idée est d'observer la fluctuation de la taille et de la précision en fonction du paramètre α, le but étant d'étudier l'espace des solutions pour déterminer s'il existe un α optimal, qui pourrait être un bon compromis taille/précision. La difficulté réside dans le fait de définir le paramètre « optimal » pour trouver le compromis « optimal ».

Le parcours de l'espace de solution peut être confronté à un autre problème que nous n'avons pas mentionné auparavant, celui des temps de calculs pour construire MC-Von. Par exemple, dans le cas de *MSNBC*, alors que la construction de D_{KL}-Von prend quelques secondes, il faut près d'une demi-heure pour MC-Von. Notons que les tâches d'analyse de réseau s'accompagnent rarement de contraintes sur le temps de calcul (dans la limite du raisonnable). Par ailleurs, une fois le réseau construit, il n'y a pas de différence

pour les algorithmes de fouille de réseaux qui seront discutés dans les chapitres suivants. Néanmoins, des travaux futurs pourraient se porter sur l'amélioration du calcul de l'approximation de la p-valeur. La méthode utilisée ici est conçue pour obtenir une solution stable contrôlant le caractère aléatoire de la méthode de Monte-Carlo. Nous pensons que des approximations plus rapides et aussi stables sont possibles, par exemple en utilisant des techniques de Monte-Carlo séquentielles [30].

3.5.2 Réflexion sur la construction des réseaux

Notre définition et la procédure générique de construction des réseaux Von (section 3.2.2) inclut des hypothèses que nous n'avons pas discutées. Le fait d'évaluer la pertinence en fonction du plus proche contexte suffixe pertinent mène à des situations qui peuvent sembler en contradiction avec notre objectif de parcimonie.

En effet, les réseaux Hon, comme précisé dans la section 3.2.2.1, doivent inclure les préfixes des contextes jugés comme pertinents. Ainsi, si un contexte `abc` est jugé comme pertinent alors le réseau final doit impérativement intégrer le nœud `ab`. Si `ab` n'est effectivement pas pertinent par rapport à `b`, on ajoute au modèle un nœud qui augmente la taille du réseau sans pour autant rajouter de l'information. Toutefois, l'ajout de ces contextes non-pertinents mais indispensables n'est pas pris en compte pour juger les autres contextes dans le cadre de Von. Ainsi, lorsque la pertinence de `cab` sera testée, la distribution de $P_{\texttt{cab}}$ sera comparée à $P_{\texttt{b}}$ et non à $P_{\texttt{ab}}$. Cependant, dans le cas où $P_{\texttt{cab}}$ et $P_{\texttt{ab}}$ sont similaires, on aurait pu se contenter de `ab`.

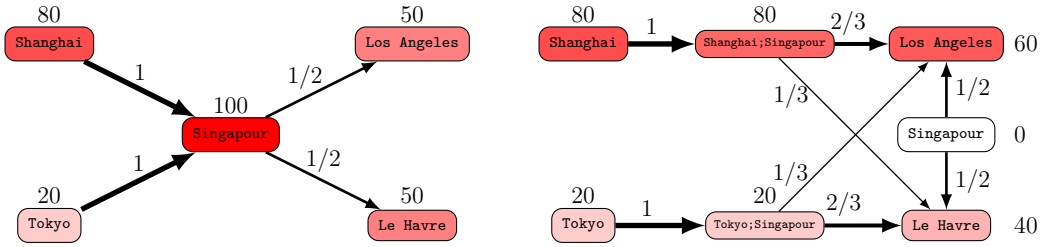
Une telle simplification pourrait être intéressante mais suppose toutefois de concevoir un algorithme qui évaluera plusieurs fois chaque contexte. En effet, `ab` est ajouté au réseau lorsqu'une de ces extensions par la droite (*e.g.* `abc`) est jugé pertinente. Cependant, si `cab` a déjà été évaluée, il faudra effectuer un nouveau test. L'algorithme 1 garantit en revanche au plus un test pour un contexte donné. Cela ne signifie toutefois pas que la recherche de contextes pertinents utilisés ici est la seule possible. D'autres solutions algorithmiques pourraient être envisagées. Nous pensons toutefois qu'il faut pour cela redéfinir la façon dont la pertinence est évaluée. Par exemple, il pourrait être intéressant de tenir compte des divergences avec les contextes suffixes autres que le dernier contexte pertinent.

Chapitre 4

Centralité dans les réseaux d'ordre supérieur

4.1 Introduction

Dans le chapitre précédent, nous avons introduit différents modèles de réseaux d'ordre supérieur. La construction de ces modèles permet d'obtenir de l'information sur le système sous-jacent et la dynamique de flux. Par exemple, le nombre de représentations d'un état dans un VON peut être intéressant pour l'expert. Toutefois, l'intérêt de la modélisation par des graphes est d'extraire de la connaissance à partir de la structure du réseau (topologie) en tenant compte des relations indirectes. Définir les probabilités de transitions et construire le graphe n'est que la première étape de notre chaîne d'analyse. Nous allons ainsi dans ce chapitre nous intéresser à la seconde étape avec l'évaluation de l'importance des états au travers des mesures de *centralité*. Le chapitre suivant sera lui centré sur le *clustering* des HON.



(a) FON$_1$ (b) FON$_2$

FIGURE 4.1 – Illustration de la diffusion de l'importance dans un réseau. Le chiffre à côté des nœuds indique leur importance dans le réseau. On considère que les ports de `Shanghai` et `Tokyo` ont une importance arbitraire de 80 et 20 respectivement. Cette importance est diffusée aux autres nœuds proportionnellement aux poids des liens.

Les mesures de centralité permettent de simplifier l'information en quantifiant « l'importance » de certaines parties du réseau en ne se limitant généralement pas au voisinage direct. Un nœud peut ainsi se voir attribuer une grande importance même s'il n'est pas connecté à beaucoup d'autres nœuds. Dans l'exemple 4.1, on peut considérer que l'importance initiale de `Shanghai` ou `Tokyo` vient du fait que ces ports capturent à l'échelle locale les flux avant de les rediriger vers `Singapour` qui bien qu'étant la destination de seulement deux ports sera considéré comme très central. Les mesures de centralité sont nombreuses dans la littérature de l'analyse de réseaux et chacune repose sur une définition différente de l'importance. Celles qui nous intéressent ici se basent sur les marches aléatoires et la « diffusion de l'importance ». Par exemple, dans le cas de la mesure PageRank [14], on dira qu'un nœud est important s'il peut être atteint à partir de nœuds importants.

Ce mécanisme est illustré dans la figure 4.1 avec des quantités arbitraires d'importance issues de nœuds sources qui se diffusent au reste du réseau. Dans FON$_1$ (figure 4.1a), les deux destinations finales sont considérées comme étant de même importance. Prendre en

compte des dépendances séquentielles aboutit à un résultat différent. Dans FON_2 (figure 4.1b), un nœud bénéficie davantage de l'asymétrie de la distribution de l'importance au départ. Les mesures de la centralité sont en pratique définies pour des cas plus complexes que cet exemple. Les premières applications des HON dans la littérature ont consisté à déterminer l'influence des dépendances séquentielles sur la centralité des états.

Dans le cas de FON_1, il y a une bijection entre les nœuds du graphe et les états représentés. Ce n'est pas le cas dans les HON. Les travaux de la littérature (section 4.2) se basent sur des transformations intuitives *i.e.* considérer la centralité d'un état comme la somme de la centralité de ses représentations. Dans la figure 4.1b, on retrouve bien la même valeur de 100 en prenant la somme de l'importance des représentations de `Singapour`. Cette approche semble *a priori* très prometteuse car elle suggère que les mesures et algorithmes existants pour les réseaux FON_1 pourront directement être appliqués sur les HON (c'est d'ailleurs un avantage souligné par Xu *et al.* [84] en décrivant leur modèle). L'objet de ce chapitre est de questionner cette hypothèse dans le cadre de la mesure PageRank.

Dans la section 4.3, nous montrons que ne pas tenir compte de l'hétérogénéité des représentations dans les HON mène à une évaluation biaisée de la centralité des états qui renforce d'autant plus l'importance des états déjà importants. Nous proposons une correction de ce biais pour la mesure PageRank. Les expérimentations que nous détaillons dans la section 4.4 montrent que la prise en compte des dépendances séquentielles va, à l'inverse, homogénéiser les écarts de centralité. Nous analysons également l'influence du choix des modèles de HON définis dans le chapitre 3. Il apparaît, dans ce cadre, que les différences dans les classements issus de PageRank sont faibles malgré les différences dans les mesures.

4.2 État de l'art

Nous présentons brièvement, dans la section 4.2.1, la littérature abondante concernant les mesures de centralité. Nous discuterons également des applications connues de ces mesures de centralité aux HON (section 4.2.2). Nous finissons par une discussion sur l'existence d'une « vérité terrain » proposée par Scholtes [75] pour évaluer une mesure de centralité dans le cadre de données séquentielles.

4.2.1 Mesures de centralité en analyse de réseaux

D'après Bloch *et al.* [9], la centralité permet d'identifier dans le réseau les nœuds les plus « influents », « centraux » ou bien « prestigieux ». La définition et la mesure ainsi choisies seront dépendantes du cas d'études et des hypothèses. Par exemple, Padgett *et*

al. [60] montre que la mesure d'« intermédiarité » (présentée ci-dessous) est la plus adaptée pour expliquer l'ascension des Medici dans la république de Florence. Mais, d'après Koschützki *et al.* [48], le terme « centralité » n'est pas clairement défini. Pour la même question, il va exister plusieurs réponses qui s'appuient sur des intuitions différentes. Les définitions des mesures à partir de ces intuitions varient ensuite en fonction de la métrique choisie, du formalisme utilisé voire des contraintes algorithmiques [42]. Un large ensemble de mesures de centralité ont ainsi été étudiées, critiquées et comparées dans la littérature [55]. Nous allons nous focaliser ici sur des mesures globales qui utilisent les relations indirectes pour tirer parti des dépendances séquentielles dans les HON. Dans ce cadre, nous discutons de deux grandes catégories de mesures de centralité ; les mesures basées sur les plus-courts-chemins et les mesures « spectrales ».

Les mesures basées sur les plus-courts-chemins tentent de capturer la place d'un nœud dans un réseau où les échanges s'effectuent de la manière la moins coûteuse possible. On définit dans ce cadre la distance entre deux nœuds comme le nombre minimum d'arcs devant être suivis pour passer d'un nœud à l'autre. Dans ce cadre, les plus-courts-chemins peuvent tenir compte de la longueur des arcs mais cet outil n'est *a priori* pas adapté aux réseaux représentant des flux comme les réseaux étudiés ici. De telles mesures incluent la mesure de proximité (*closeness centrality*) qui est l'inverse des sommes des distances entre un nœud et les autres nœuds du réseau. Une autre mesure populaire est la mesure d'« intermédiarité » (*betweenness centrality)* qui se base sur le nombre de plus-courts-chemins passant par un nœud. Un nœud important dans ce cadre est par exemple un nœud servant de « pont » entre différentes parties du réseaux peu connectées par ailleurs.

Les mesures « spectrales » se basent sur une mécanique de diffusion de l'importance telle qu'illustrée dans la figure 4.1. Dans le cadre de la « centralité spectrale » (*eigenvector centrality*), chaque nœud émet autant d'importance et la diffuse uniformément à ses voisins. Si ce processus est répété un grand nombre de fois, on aboutit à une distribution stationnaire qui peut être définie récursivement : un nœud sera alors d'autant plus central que ses voisins le sont. La plupart des mesures « spectrales » sont des variantes normalisées de ce principe. Comme indiqué plus haut, elles ont un intérêt pour nous car elles permettent de tenir compte des dépendances séquentielles intégrées dans les HON. La mesure *PageRank*, qui nous intéresse dans ce chapitre, fait partie de cette catégorie. *PageRank* a été mise en œuvre dans le moteur de recherche de Google par ses inventeurs Brin et Page [14] pour classer les résultats d'une recherche. Elle a l'avantage d'intégrer un mécanisme de téléportation qui permet le calcul de l'importance dans le cas de graphe orienté. Nous donnerons une définition plus précise de cette mesure dans la section 4.3.1.

4.2.2 La centralité des états avec les réseaux d'ordre supérieur

Scholtes *et al.* [76] se concentrent sur les mesures basées sur les plus-courts-chemins notamment l'intermédiarité (*betweenness*) et la proximité (*closeness*). Ils généralisent ces méthodes aux réseaux construits à partir de réseaux temporels (présentés dans la section 2.5.6). Dans ce cas, les auteurs n'utilisent pas les probabilités de transitions entre états. Les expérimentions montrent que ces mesures sont de bonnes approximations de mesures similaires mais calculées directement sur les réseaux temporels.

Dans le reste des travaux sur la centralité appliquée aux HON, la mesure utilisée est le *PageRank*. Il existe des applications pour les modèles présentés précédemment : SN_2 [70], FON_{opt} [75] et D_{KL}-VON_k [84]. Dans tous les cas, les auteurs ramènent les valeurs de *PageRank* aux états en faisant la somme de valeurs de *PageRank* de toutes leurs représentations. Cette transformation est discutée dans la section 4.3.2.

Pour SN_2, Rosvall *et al.* [70] s'intéressent à l'effet de l'ajout de la mémoire sur les classements, notamment dans le cas des classements de journaux scientifiques issus de réseaux de citations entre articles (la construction est également présentée dans la section 2.5.6). Les auteurs décrivent le phénomène de « fuite » du flux, dans le sens où un journal multidisciplinaire redistribuera le flux reçu dans des journaux de domaines différents (et ce même s'il est peu probable que des lecteurs fassent le pas). Les journaux qui bénéficient de ce genre de « fuite » seront alors avantagés en utilisant PageRank sur le réseau SN_1. Ils notent aussi que les classements des nœuds des réseaux d'ordre 1 sont plus sensibles aux manipulations liées à l'auto-citation. De ce fait, ils concluent que les modèles comme SN_2 sont plus robustes pour classer les journaux.

Pour les FON_{opt}, Scholtes [75] utilise le *PageRank* comme critère de validation pour la sélection de l'ordre optimal (décrit dans la section 3.2.1). Les résultats montrent que les classements issus de PageRank dans FON_{opt} sont davantage corrélés à cette « vérité-terrain ».

Pour les D_{KL}-VON_k, Xu *et al.* [84] étudient la différence des classements *PageRank* sur des données de navigation de sites internet par rapport à FON_1. En utilisant D_{KL}-VON, les auteurs remarquent que la valeur PageRank d'une grande majorité des pages diminue, tandis que les autres obtiennent un gain non négligeable. La structure du site permet aux auteurs d'expliquer pourquoi le fait d'ajouter les dépendances séquentielles peut changer le score *PageRank*. Sur le D_{KL}-VON, on remarque en effet que le réseau permet d'encoder des comportements logiques d'utilisateur sur le site ne pouvant être capturés par FON_1.

4.2.3 Une « vérité-terrain » pour évaluer l'importance des états ?

Scholtes [75] évalue l'ordre optimal de FON_{opt} en utilisant les « probabilités de visite » (*visitation probabilities*) des états (voir équation 4.1).

$$\psi(\sigma) = \frac{c(\sigma)}{\sum_{s \in \mathcal{S}} |s|} \tag{4.1}$$

L'hypothèse est que cette mesure sera plus proche du PageRank calculé sur le réseau d'ordre fixe optimal. Elle est donc utilisée comme une « vérité-terrain » pour valider le choix de FON_{opt}. Intuitivement, on peut en effet voir les séquences $\mathcal{S}$ comme des marches qui, évidemment, sont au moins aussi fidèles à elles-mêmes qu'une marche aléatoire sur le réseau (même d'ordre supérieur).

Cette hypothèse est pour nous discutable. En effet, la modélisation de réseaux d'ordre supérieur permet d'intégrer deux types de relations indirectes entre états comme discuté dans la section 1.4. Les « probabilités de visite » vont ici tenir compte du premier type de relations (deux états appartiennent à la même séquence) mais pas du second.

Cette observation est d'autant plus claire si on considère le cas « extrême » suivant : les séquences $\mathcal{S}$ ne contiennent que des paires $(u, v) \in \mathcal{A} \times \mathcal{A}$. Il n'y a dans ce cas pas de dépendances séquentielles, toutes les relations indirectes sont du second type et on peut analyser ces données en utilisant le réseau FON_1. Considérons le graphe multiple non-orienté $G_1 = (\mathcal{A}, E)$ obtenu en oubliant l'orientation des arcs dans FON_1. La « probabilité de visite » d'un $\sigma \in \mathcal{A}$ est alors égale à $\frac{deg(\sigma)}{2|E|}$. Cette probabilité est statistiquement proche du PageRank calculé sur G_1 [62] et seulement égale si G_1 est *régulier* (*i.e.* les nœuds de G_1 ont tous le même degré) [32].

Nous pouvons modifier l'hypothèse de Scholtes [75] en proposant que la meilleure mesure de centralité pour un graphe orienté doit être au plus proche des « probabilités de visite ». Dans ce cadre, la « meilleure » centralité serait toute mesure parfaitement proportionnelle au nombre d'arcs impliquant chaque état (*i.e.* le degré dans G_1). Une telle conclusion est en contradiction avec la littérature existante sur la centralité. En particulier, elle donne une grande importance à des nœuds connectés avec des nœuds de faible degré ce qui est précisément le type de situation que PageRank cherche à éviter.

Nous n'utilisons pas ici de « vérité-terrain » mais discutons les différences entre les classements PageRank obtenus selon différents modèles dans la section 4.4.1.2. Nous montrons d'ailleurs que les classements issus de ψ peuvent être plus proches des classements PageRank sur des réseaux FON_1 volontairement biaisés pour favoriser les états avec davantage de représentations.

4.3 Adaptation de la mesure *PageRank* aux réseaux d'ordre supérieur

Dans cette section, nous définissons tout d'abord la mesure du PageRank (aussi noté PR) et son application directe à Von. Nous discutons de l'effet de la distribution du nombre de répétitions sur la distribution des probabilités du PR. Afin d'isoler cet effet et évaluer son importance, nous avons introduit un modèle de PR de réseau de premier ordre biaisé. Nous présentons ensuite une méthode pour annuler ce biais.

4.3.1 Modèle PageRank standard sur $\mathbf{Fon}_1$

Le PR est équivalent à l'état stable d'un « surfeur aléatoire » (noté SA) qui suit un processus de Markov sans mémoire. Ce surfeur peut suivre les liens du réseau avec une probabilité de τ ou se téléporter uniformément vers un nœud du réseau avec une probabilité de $(1-\tau)$ (il se téléportera également à partir de n'importe quel nœud puits (*i.e.* sans arc sortant). Ces téléportations garantissent que le SA ne peut pas être bloqué dans une sous-partie du réseau et que la distribution des probabilités à l'état stable est unique. Le paramètre τ est parfois appelé le « facteur d'amortissement » et est généralement fixé à $\tau = 0.85$.

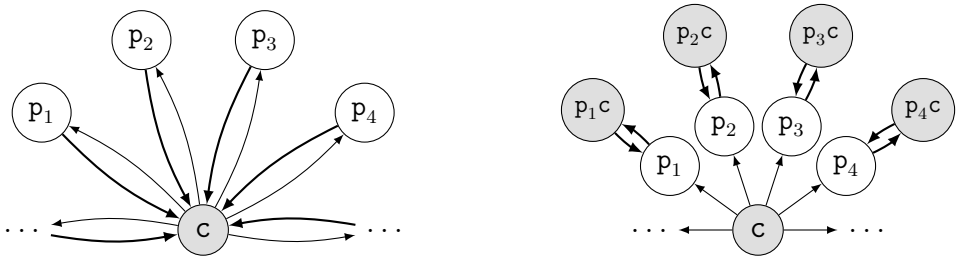
Définition 4.1. *(PageRank.) Pour un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, le PageRank (noté PR) est le vecteur $[\pi(v)]_{v\in\mathcal{V}}$ tel que*

$$\pi(v) = (1-\tau)\boldsymbol{e}(v) + \tau \sum_{u\in\mathcal{N}^-(v)} w(u,v)\pi(u) \tag{4.2}$$

où $\tau \in (0,1)$ et $\boldsymbol{e}(v) = \frac{1}{|\mathcal{V}|}$ est la probabilité uniforme pour le surfeur de se téléporter sur le nœud v.

Dans Fon_1, l'état $\sigma \in \mathcal{A}$ est représenté par un unique nœud σ. Le PR associé à l'état σ dans Fon_1 est noté $\Pi_1(\sigma) = \pi(\sigma)$. Nous notons K_1 le classement par ordre décroissant des états en fonction de Π_1.

Le vecteur $\boldsymbol{e}$ est parfois appelé « vecteur de personnalisation » (*personalisation vector).* Dans l'exemple de graphe non-orienté discuté dans la section 4.2.3, le PR sera égal à la « probabilité de visite » ψ précisément lorsque $\boldsymbol{e} = \psi$ (Corollaire 1 de [32]) *i.e.* lorsque le SA se téléportera préférentiellement sur les nœuds avec un degré plus important. Le vecteur $\boldsymbol{e}$ sera par la suite utilisé pour évaluer et corriger le biais de PR directement évalué sur les Hon définis dans la section suivante.



(a) Von sans aller-retour (b) Von avec aller-retour

FIGURE 4.2 – Réseaux Von obtenus pour deux dynamiques de flux entre un état central $\mathtt{c}$ et des états « périphériques » $(\mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3, \ldots)$. En (a), en quittant $\mathtt{c}$, un marcheur va uniformément vers n'importe lequel des satellites $\mathtt{p}_i$. Le modèle Von optimal devrait ici correspondre à un modèle Fon_1. En (b), un marcheur venant de $\mathtt{p}_i$ retourne toujours à $\mathtt{p}_i$ après être passé par $\mathtt{c}$. Le modèle Von optimal inclut une représentation de $\mathtt{c}$ pour chaque état périphérique.

4.3.2 PageRank directement appliqué à un Hon

Bien que le surfeur aléatoire corresponde à un processus de Markov sans mémoire, dans le cas des Hon, ce processus va indirectement simuler un modèle d'ordre supérieur. Dans ce cadre, un état peut être représenté par plusieurs nœuds. Comme indiqué dans la section 4.2.2, la littérature suggère que le PR d'un état σ dans un Hon noté $\Pi_{\text{Hon}}(\sigma)$ peut être défini comme la somme de PR des nœuds calculée de la même façon que sur un réseau de premier ordre *i.e.*

$$\Pi_{\text{Hon}}(\sigma) = \sum_{v \in \mathcal{V}(\sigma)} \pi(v) \tag{4.3}$$

L'équation 4.3 s'interprète comme la probabilité stationnaire de visiter *une des* représentations de l'état σ. Nous désignons par K_{Hon} le classement issu de Π_{Hon}.

Puisque nous utilisons un surfeur aléatoire, on peut noter que plus l'état σ a de représentations, plus la probabilité de se téléporter vers l'une d'entre elles est élevée. Puisque le PR d'un état Π_{Hon} (équation 4.3) est défini comme la somme sur les représentations de l'état σ, la mécanique de téléportation va induire un biais dans la mesure Π_{Hon}. Nous pouvons illustrer cet effet par un exemple simple illustré dans la figure 4.2. Notons que pour le cas 4.2a, on aura $\Pi_{\text{Hon}} = \Pi_1$ puisque le Von ne contient ici pas de nœud-mémoire. Pour l'état $\mathtt{c}$, on a ainsi

$$\Pi_1(\mathtt{c}) = \frac{n\tau + 1}{(n+1)(\tau+1)} \tag{4.4}$$

$$\Pi_{\text{Hon}}(\mathtt{c}) = \frac{n(1+\tau) + 1}{(2n+1)(1+\tau)} \tag{4.5}$$

où n est le nombre d'états périphériques. Avec $\tau < 1$, le PR de $\mathtt{c}$ dans le cas 4.2b $\Pi_{\text{Hon}}(\mathtt{c})$ sera toujours supérieur à $1/2$ alors que $\Pi_1(\mathtt{c}) < 1/2$. L'égalité est atteinte lorsque $\tau = 1$

(c'est-à-dire lorsqu'il n'y a pas de téléportation).

La caractérisation de cet effet comme un « biais » peut être discutée. En effet, l'exemple de la figure 4.2 correspond à des dynamiques différentes. Toutefois, il nous semble compliqué de considérer que, dans le cas 4.2b, l'état `c` est « plus central » si la dynamique de flux implique des aller-retours. Il serait en fait plus facile de justifier l'inverse puisque, dans le cas 4.2a, `c` permet des échanges entre les différentes parties du réseau. Dans tous les cas, la différence ici ne repose que sur le mécanisme de téléportation et non sur la dynamique sous-jacente.

4.3.3 PageRank biaisé sur Fon_1

Afin d'isoler le biais dû aux téléportations, nous allons supposer que les probabilités de transition associées aux représentations associées à chaque état sont toutes égales. Chaque état aura ainsi autant de représentations mais le HON n'encodera aucune dépendance séquentielle. Sans en faire la preuve ici, le PR obtenu sera égal au PR calculé sur FON_1 en utilisant un vecteur de personnalisation non-uniforme préférentielle $\boldsymbol{e}_{\text{B}}$ dépendant du nombre de représentations $N_{\mathcal{V}}$ des états dans le HON :

$$\boldsymbol{e}_{\text{B}}(\sigma) = \frac{N_{\mathcal{V}}(\sigma)}{\sum_{\sigma \in \mathcal{A}} N_{\mathcal{V}}(\sigma)} = \frac{N_{\mathcal{V}}(\sigma)}{|\mathcal{V}|} \tag{4.6}$$

Les valeurs PR des éléments associées à ce modèle et le classement qui en résulte sont notés Π_1^{B} et K_1^{B} respectivement. Dans l'exemple de la figure 4.2, Π_1^{B} calculé sur la figure 4.2a est ainsi égal à Π_{HON} calculé sur la figure 4.2b.

De manière générale, il est naturel de supposer que la distribution $\boldsymbol{e}_{\text{B}}$ n'est pas arbitraire mais dépend du nombre de fois où cet état a été vu dans $\mathcal{S}$: en effet, par construction, un état peu fréquent va être peu représenté. Nous verrons dans les expérimentations que K_1^{B} est proche du classement obtenu en utilisant ψ (équation 4.1).

4.3.4 PageRank non-biaisé sur un Von

Afin d'éliminer le biais évoqué ci-dessus, une modification du vecteur de personnalisation est également utilisée. Bien que plusieurs corrections soient possibles, celle choisie correspond à redéfinir le processus stochastique du surfeur aléatoire. Intuitivement, la téléportation est supposée être le début d'une nouvelle marche. Il n'est donc possible de se téléporter uniformément que vers les nœuds de premier ordre puisqu'on considère que la mémoire des étapes précédentes n'est pas importante. Cela correspond à utiliser le vecteur de personnalisation $\boldsymbol{e}_{\text{U}}$ (voir équation 4.7).

Mesure	Classement	Description
Π_1	K_1	PR calculé sur FON_1
Π_{HON}	K_{HON}	PR calculé sur un D_{KL}-VON(1)
Π_1^{B}	K_1^{B}	PR calculé sur FON_1 avec téléportations dépendantes de $N_{\mathcal{V}}$
$\Pi_{\text{HON}}^{\text{U}}$	$K_{\text{HON}}^{\text{U}}$	PR non-biaisé calculé sur un D_{KL}-VON(1)
$N_{\mathcal{V}}$	$K_{\mathcal{V}}$	Nombre de représentations des états
ψ	K_{ψ}	« Probabilité de visite » (équation 4.1).

TABLE 4.1 – Résumé des mesures et classements comparés.

$$\boldsymbol{e}_{\text{U}}(v) = \begin{cases} 1/|\mathcal{A}| & \text{if } |v| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Il est facile de voir que l'état d'équilibre du SA est toujours unique en utilisant ce vecteur de téléportation (voir théorème 4.1). Les valeurs PR des éléments associées à ce modèle et le classement qui en résulte sont notés $\Pi_{\text{HON}}^{\text{U}}$ et $K_{\text{HON}}^{\text{U}}$ respectivement.

Théorème 4.1. *(Existence et unicité de Π_{HON}^{U}) Pour $\tau < 1$, Π_{HON}^{U} existe et est unique.*

Démonstration. Il nous suffit de montrer que le surfeur aléatoire se téléportant sur les nœuds $\mathcal{V}$ avec probabilité v_{U} peut atteindre n'importe quel $v \in \mathcal{V}$. On note $p_U(v|u)$ la probabilité que le surfeur atteigne v à partir de $u \in \mathcal{V}$. On a $p_U(v|u) = (1-\tau)p(v|u)+\tau v_{\text{U}}$. Si $|v| = 1$, alors on a $\forall u \in \mathcal{V}, p_U(v|u) > 0$. Sinon, soit $v = \sigma_1\sigma_2 \dots \sigma_m$, d'après le théorème 2.1, v est accessible depuis σ_1 et $\forall u \in \mathcal{V}, p_U(\sigma_1|u) > 0$. □

Si on poursuit avec l'exemple de la figure 4.2, les valeurs de $\Pi_{\text{HON}}^{\text{U}}$ seront ici les mêmes pour les deux dynamiques de flux bien que les réseaux aient des topologies différentes.

4.4 Résultats expérimentaux

Dans la section 4.4.1, nous montrons que l'effet de biais est non négligeable lorsque l'on compare les différentes mesures de PR présentées dans la section 4.3. Nous comparons ensuite les PR non-biaisés obtenus selon les modèles de HON discutés dans le chapitre 3.

4.4.1 Influence du biais

Pour les quatre ensembles de séquences étudiées, nous calculons les différentes variantes de PR avec $\tau = 0,85$ ainsi que les classements (en ordre décroissant) correspondants. Nous allons analyser l'effet de la présence ou de l'absence du biais dans un HON avec le modèle D_{KL}-VON(1). Les notations des mesures et classements sont données dans le tableau 4.1. Les états avec la même mesure sont classés dans le même ordre. Outre les quatre modèles décrits dans la section précédente, nous utilisons également le nombre de représentations

$N_{\mathcal{V}}$ ainsi que la « probabilité de visite » ψ (équation 4.1) discutée dans la section 4.2.3.

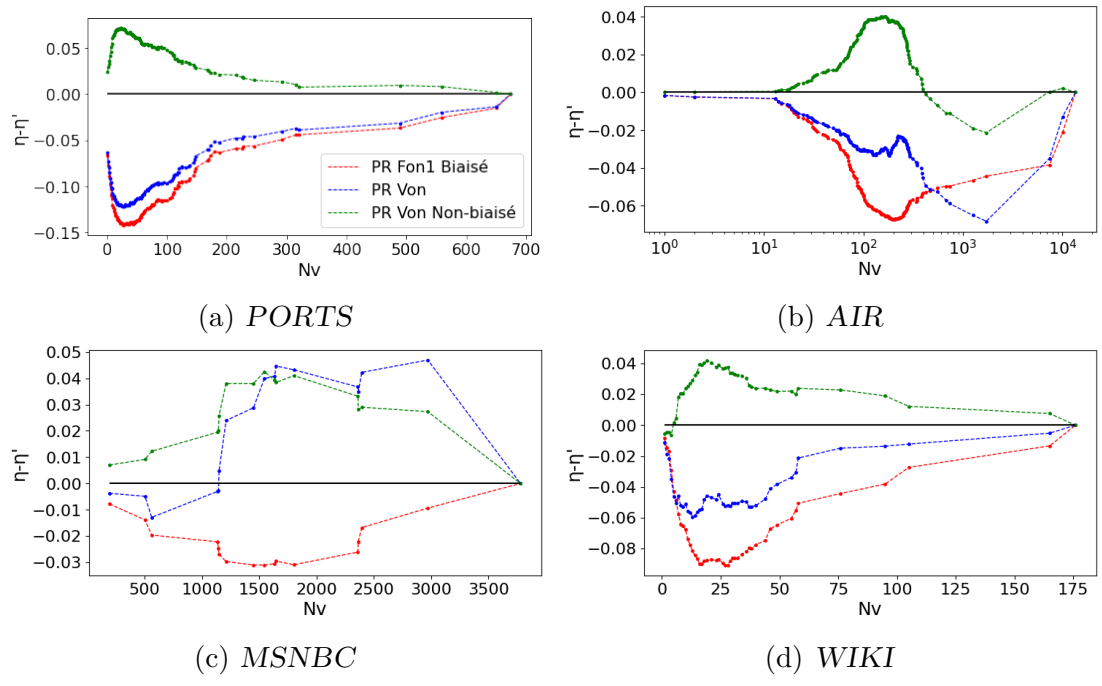
Nous montrons dans cette section que l'effet de biais est effectivement important lorsque l'on examine les valeurs de Π même s'il semble plus marginal si on considère les classements. De plus, ces observations restent vraies lorsque des valeurs différentes de τ sont utilisées.

4.4.1.1 Évolution des valeurs de PR en fonction de $N_{\mathcal{V}}$.

Pour une mesure de PR Π, nous notons $\eta(\Pi, k)$ la probabilité qu'un surfeur aléatoire visite un état ayant au plus k représentations *i.e.*

$$\eta(\Pi, k) = \sum_{\sigma \in \mathcal{A}} \Pi(\sigma) \text{ avec } N_{\mathcal{V}}(\sigma) \leq k \tag{4.8}$$

L'impact de $N_{\mathcal{V}}$ sur les probabilités de PR est quantifié par l'augmentation relative des PR $(\eta - \eta^{'})$ où $\eta^{'} = \eta(\Pi_1, .)$. On mesure donc la différence par rapport au PR calculé dans FON$_1$. Les évolutions de $(\eta - \eta^{'})$ pour Π_{HON}, Π_1^{B} et $\Pi_{\text{HON}}^{\text{U}}$ sont rapportées dans la figure 4.3.



(a) *PORTS* (b) *AIR*

(c) *MSNBC* (d) *WIKI*

FIGURE 4.3 – Variation du PageRank $(\eta - \eta^{'})$ (voir équation 4.8) en fonction de $N_{\mathcal{V}}$ pour trois mesures de PR, avec $\tau = 0,85$.

Les mesures Π_{HON} et Π_1^{B} semblent pénaliser davantage les états ayant un nombre faible de représentations. Dans le cas de *PORTS* ou *WIKI*, les distributions de $(\eta - \eta^{'})$ sont très proches. Un comportement différent apparaît pour *MSNBC*. Toutefois, le faible nombre d'états rend le comportement des mesures plus erratique. À l'inverse, $\Pi_{\text{HON}}^{\text{U}}$ donne une

plus grande importance aux états peu représentés, y compris par rapport à Π_1. On peut en conclure que la prise en compte des dépendances temporelles semble davantage uniformiser les valeurs de PR tandis que le biais lié aux nombre de représentations renforce les états déjà très centraux aux dépens des périphériques. Cela s'explique par le fait que ces derniers ont généralement un faible nombre de représentations et sont pénalisés par la téléportation biaisée.

Cependant, le fait que les valeurs PR sont plus ou moins homogènes ne signifie pas nécessairement que les classements issus des mesures PR seront différents. Il faut pour cela analyser les différences dans les classements obtenus pour chaque mesure.

4.4.1.2 Comparaison des classements

Nous avons quantifié les similitudes entre les paires de classements de PR en utilisant les coefficients de corrélation de Spearman r_s et de Kendall r_τ (voir tableau 4.2).

TABLE 4.2 – Coefficients de Spearman (r_s) et Kendall (r_τ) entre les classements de PR, ainsi que $K_\mathcal{V}$ et K_ψ (voir tableau 4.1).

	K_1		K_1^{B}		K_{HON}		$K_{\mathrm{HON}}^{\mathrm{U}}$	
	r_s	r_τ	r_s	r_τ	r_s	r_τ	r_s	r_τ
	a) *PORTS*							
K_1	-	-	0.962	0.851	0.947	0.814	0.955	0.825
K_1^{B}	-	-	-	-	0.983	0.894	0.916	0.754
K_{HON}	-	-	-	-	-	-	0.986	0.908
K_ψ	0.944	0.821	**0.992**	**0.933**	0.972	0.869	0.901	0.741
	b) *AIR*							
K_1	-	-	0.999	0.981	0.978	0.883	0.978	0.898
K_1^{B}	-	-	-	-	0.980	0.889	0.976	0.894
K_{HON}	-	-	-	-	-	-	0.932	0.786
K_ψ	**0.998**	**0.971**	0.997	0.960	0.979	0.880	0.983	0.906
	b) *MSNBC*							
K_1	-	-	0.995	0.971	0.939	0.838	0.975	0.897
K_1^{B}	-	-	-	-	0.946	0.868	0.983	0.926
K_{HON}	-	-	-	-	-	-	0.983	0.941
K_ψ	**0.975**	**0.912**	0.963	0.882	0.900	0.780	0.939	0.838
	d) *WIKI*							
K_1	-	-	0.921	0.783	0.810	0.657	0.900	0.750
K_1^{B}	-	-	-	-	0.928	0.783	0.842	0.673
K_{HON}	-	-	-	-	-	-	0.892	0.735
K_ψ	0.665	0.498	**0.715**	**0.552**	0.606	0.447	0.551	0.399

Malgré les différences dans les valeurs de PR, nous observons des grands scores de corrélations entre tous les classements. On peut toutefois constater une plus grande simi-

litude entre les classements K_1^{B} et K_{HON} qu'entre K_1^{B} et $K_{\mathrm{HON}}^{\mathrm{U}}$ (sauf pour le jeu de données *MSNBC*). Ceci peut indiquer que le biais du nombre de représentations risque d'occulter les différences induites par les dépendances séquentielles. Ce phénomène se retrouve lorsque l'on se focalise sur le « Top 10 » des états, qui est une utilisation populaire de la mesure PR. Les différents Top 10 sont donnés dans le tableau 4.3. Par exemple, dans le cas de *PORTS*, le Top 10 de K_{HON} correspond au Top 10 de $K_{\mathcal{V}}$. Notons cependant que les classements PR sont généralement proches du classement $K_{\mathcal{V}}$. Ceci est compréhensible : un état avec un nombre important de représentations est probablement impliqué dans de nombreuses transitions.

Dans la section 4.2.3, nous mentionnons l'utilisation de la « probabilité de visite » ψ et le classement correspondant K_ψ comme une « vérité de terrain » utilisée par Scholtes [75] pour sélectionner l'ordre optimal FON_{opt}. Le tableau 4.2 rapporte les corrélations entre K_ψ et les autres classements PR (les corrélations les plus élevées sont en gras). Notons que, pour *PORTS* et *WIKI*, K_ψ est plus proche de K_1^{B} que des autres classements. Dans les deux autres cas (*AIR* et *MSNBC*), K_ψ est plus proche de K_1 que des classements tenant compte des dépendances séquentielles. L'ordre optimal détecté dans chaque cas est pourtant supérieur à 1. Si nous devions utiliser K_ψ comme méthode de sélection de la mesure PR, nous sélectionnerions probablement le modèle de PR biaisé FON qui n'inclut aucune dépendance d'ordre supérieur. Cela souligne le fait que K_ψ n'est, selon nous, pas une mesure pertinente dans le cadre de l'évaluation des mesures PR.

4.4.1.3 Dépendance du biais $N_{\mathcal{V}}$ avec le facteur d'amortissement τ.

Le biais lié à la téléportation dépend du facteur d'amortissement de PR τ. Nous avons utilisé une valeur standard de $\tau = 0.85$ mais l'utilisation d'autres valeurs a été étudiée dans la littérature [22]. L'évolution des coefficients de Spearman r_s en fonction des variations de τ est présentée dans la figure 4.4 pour $\tau \in (0.75, 0.99)$. Les résultats relatifs à l'évolution du coefficient de corrélation de Kendall r_τ ne sont pas présentés car ils sont similaires.

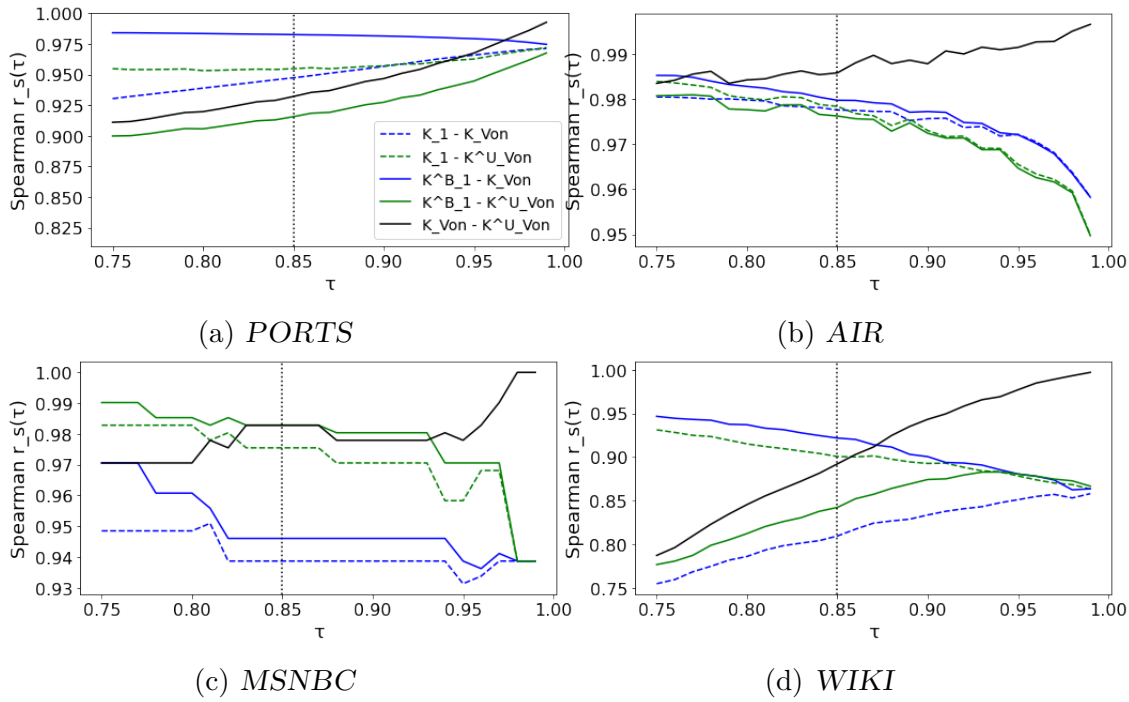
En augmentant τ, les classements issus des deux mesures sur FON_1 se rapprochent, de même que ceux issus de D_{KL}-VON. Ceci est normal car la seule différence entre ces mesures est le vecteur de téléportation utilisé (équation 4.2 et 4.6). On peut également noter pour que les classements biaisés K_1^{B} et K_ψ sont plus proches entre eux que pour K_1^{B} et $K_{\mathrm{HON}}^{\mathrm{U}}$, ce dernier étant plus proche de K_1. Ce phénomène est plus apparent dans les jeux de données *PORTS* ou *WIKI* et n'est pas valable pour *MSNBC*. De manière générale, les conclusions tirées de la comparaison des classements pour $\tau = 0.85$ sont toujours valables en modifiant le facteur d'amortissement.

TABLE 4.3 – Top 10 des classements dans chaque jeu de données. Les états en gras sont les états entrant dans le Top 10 par rapport au classement directement à gauche.

	K_1			K_1^{B}			K_{HON}			$K_{\mathrm{HON}}^{\mathrm{U}}$		
	PORTS											
Rank	Port	$K_{\mathcal{V}}$	K_{ψ}	Port	$K_{\mathcal{V}}$	K_{ψ}	Port	$K_{\mathcal{V}}$	K_{ψ}	Port	$K_{\mathcal{V}}$	K_{ψ}
1	Singapore	2	2	Singapore	2	2	Hong Kong	1	1	Singapore	2	2
2	Hong Kong	1	1	Hong Kong	1	1	Singapore	2	2	Busan	4	4
3	Rotterdam	5	7	Shanghai	3	3	Shanghai	3	3	Hong Kong	1	1
4	Busan	4	4	Busan	4	4	Busan	4	4	Rotterdam	5	7
5	Shanghai	3	3	Rotterdam	5	7	Rotterdam	5	7	Shanghai	3	3
6	Hamburg	8	10	Port Klang	6	6	Port Klang	6	6	Hamburg	8	10
7	Port Klang	6	6	Kaohsiung	7	5	Kaohsiung	7	5	Antwerp	10	12
8	Antwerp	10	12	Hamburg	8	10	Hamburg	8	10	**Bremerhaven**	12	19
9	Bremerhaven	12	19	Antwerp	10	12	Antwerp	10	12	**Surabaya**	45	36
10	Kaohsiung	7	5	**Jebel Ali**	9	11	Jebel Ali	9	11	Port Klang	6	6
	AIR											
Rank	Airport	$K_{\mathcal{V}}$	K_{ψ}	Airport	$K_{\mathcal{V}}$	K_{ψ}	Airport	$K_{\mathcal{V}}$	K_{ψ}	Airport	$K_{\mathcal{V}}$	K_{ψ}
1	DFW	1	1	DFW	1	1	DFW	1	1	DFW	1	1
2	ORD	2	2	ORD	2	2	ORD	2	2	ORD	2	2
3	STL	3	3	STL	3	3	STL	3	3	STL	3	3
4	LAX	4	4	LAX	4	4	LAX	4	4	LAX	4	4
5	MIA	5	5	MIA	5	5	MIA	5	5	BOS	6	6
6	BOS	6	6	BOS	6	6	BOS	6	6	MIA	5	5
7	SJU	7	9	SJU	7	9	SJC	8	8	LGA	10	7
8	LGA	10	7	LGA	10	7	SJU	7	9	SJU	7	9
9	SJC	8	8	SJC	8	8	LGA	10	7	SJC	8	8
10	SFO	13	10	SFO	13	10	**SAN**	15	12	SAN	15	12
	MSNBC											
Rank	Page	$K_{\mathcal{V}}$	K_{ψ}	Page	$K_{\mathcal{V}}$	K_{ψ}	Page	$K_{\mathcal{V}}$	K_{ψ}	Page	$K_{\mathcal{V}}$	K_{ψ}
1	frontpage	1	1	frontpage	1	1	frontpage	1	1	frontpage	1	1
2	news	2	2	news	2	2	news	2	2	news	2	2
3	misc	5	5	misc	5	5	local	3	4	local	3	4
4	on-air	4	3	on-air	4	3	misc	5	5	misc	5	5
5	local	3	4	local	3	4	on-air	4	3	on-air	4	3
6	sports	6	6	sports	6	6	sports	6	6	sports	6	6
7	business	7	7	business	7	7	business	7	7	business	7	7
8	tech	8	9	tech	8	9	tech	8	9	tech	8	9
9	msn-news	13	8	msn-news	13	8	health	9	12	health	9	12
10	living	10	11	health	9	12	**opinion**	11	15	**msn-news**	13	8
	WIKI											
Rank	Article	$K_{\mathcal{V}}$	K_{ψ}	Article	$K_{\mathcal{V}}$	K_{ψ}	Article	$K_{\mathcal{V}}$	K_{ψ}	Article	$K_{\mathcal{V}}$	K_{ψ}
1	United States	1	1	United States	1	1	United States	1	1	United States	1	1
2	Europe	2	2	Europe	2	2	Europe	2	2	Europe	2	2
3	United Kingdom	3	3	United Kingdom	3	3	United Kingdom	3	3	Microsoft	74	65
4	Africa	16	6	Africa	16	6	World War II	6	8	Africa	16	6
5	Earth	4	4	Earth	4	4	Earth	4	4	World War II	6	8
6	Computer	19	12	North America	5	7	Africa	16	6	United Kingdom	3	3
7	Microsoft	74	65	England	7	5	North America	5	7	Computer	19	12
8	World War II	6	8	World War II	6	8	England	7	5	**Agriculture**	85	24
9	England	7	5	Computer	19	12	Computer	19	12	Earth	4	4
10	North America	5	7	**Atlantic Ocean**	8	16	**Microsoft**	74	65	**Internet**	42	21

4.4.2 Classements PageRank non-biaisés selon les modèles Hon

Nous disposons maintenant d'une mesure de centralité permettant de classer et comparer l'importance des états dans les HON. Cette mesure tient compte des dépendances séquentielles détectées par le modèle sans pour autant être affectée par le nombre de représentations des états. Toutefois, comme mentionné plusieurs fois, la prise en compte de dépendances séquentielles ne signifie pas que l'importance des états sera différente. Et, comme nous l'avons vu dans la section précédente, des mesures différentes n'impliquent pas forcément des classements avec de grandes différences. Nous comparons ici les classements PageRank non-biaisés obtenus selon les différents modèles discutés dans le chapitre



(a) *PORTS* (b) *AIR*

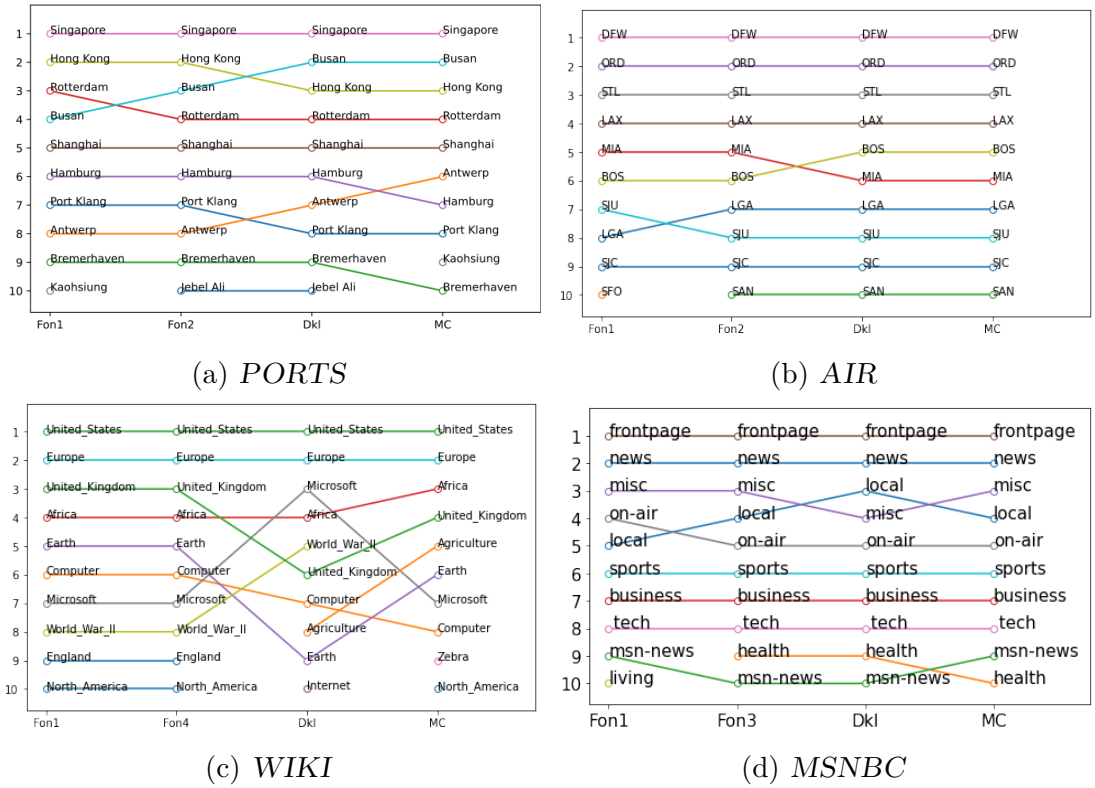
(c) *MSNBC* (d) *WIKI*

FIGURE 4.4 – Évolution des corrélations de Spearman $r_s(\tau)$ selon τ. Les lignes pleines (lignes en pointillés) correspondent aux corrélations entre K_1^{B} (K_1) et les classements de K_{HON} (en bleu) et $K_{\mathrm{HON}}^{\mathrm{U}}$ (en vert).

TABLE 4.4 – Coefficients de Spearman (r_s) et Kendall (r_τ) entre les classements de PageRank non-biaisés.

	$K^U_{\mathrm{FON}_{opt}}$		$K^U_{D_{KL}\text{-VON}}$		$K^U_{\mathrm{MC\text{-}VON}}$	
	r_s	r_τ	r_s	r_τ	r_s	r_τ
	a) *PORTS*					
K_1	0.960	0.844	0.955	0.825	0.979	0.888
$K^U_{\mathrm{FON}_{opt}}$	-	-	0.990	0.919	0.961	0.852
$K^U_{D_{KL}\text{-VON}}$	-	-	-	-	0.956	0.833
	b) *AIR*					
K_1	0.996	0.964	0.978	0.898	0.972	0.879
$K^U_{\mathrm{FON}_{opt}}$	-	-	0.988	0.918	0.981	0.899
$K^U_{D_{KL}\text{-VON}}$	-	-	-	-	0.992	0.927
	b) *MSNBC*					
K_1	0.990	0.964	0.975	0.898	0.985	0.879
$K^U_{\mathrm{FON}_{opt}}$	-	-	0.988	0.918	0.990	0.899
$K^U_{D_{KL}\text{-VON}}$	-	-	-	-	0.983	0.928
	d) *WIKI*					
K_1	1.	1.	0.900	0.750	0.963	0.848
$K^U_{D_{KL}\text{-VON}}$	-	-	-	-	0.938	0.810

3 à savoir FON_1, FON_{opt}, D_{KL}-VON(1) et MC-VON(0.001) respectivement. Nous notons les classements correspondants K_1, $K^U_{\mathrm{FON}_{opt}}$, $K^U_{D_{KL}\text{-VON}}$, $K^U_{\mathrm{MC\text{-}VON}}$ respectivement.



(a) *PORTS* (b) *AIR*

(c) *WIKI* (d) *MSNBC*

FIGURE 4.5 – Différences entre les dix premiers états en termes de PageRank selon le modèle de réseau (FON$_1$, FON$_{opt}$, D_{KL}-VON(1) et MC-VON(0.001)).

Le tableau 4.4 contient les scores de corrélations de Spearman r_s et de Kendall r_τ entre les différents classements. La figure 4.5 illustre les différences entre les dix états les plus importants selon les modèles. Nous observons des similarités fortes entre tous les classements y compris K_1. Pour chaque coefficient calculé, l'hypothèse d'indépendance peut être en effet rejetée avec une faible p-valeur. Nous pouvons en conclure que la prise en compte de dépendances séquentielles, peu importe la méthode, ne bouleverse pas les hiérarchies existantes entre les ports, aéroports ou pages web. Une conséquence, dans ce cadre, est que la mesure de PageRank sur FON$_1$ est toujours pertinente même si des dépendances séquentielles existent dans le système. Paradoxalement, c'est lorsque l'existence de dépendances séquentielles semble la moins claire, par exemple avec *WIKI*, que les corrélations entre K_1 et les autres classements semblent les plus faibles. Cela peut s'expliquer par le fait que les mesures de PageRank sont plus homogènes pour ce jeu de données. Notre conclusion ne s'appliquera probablement pas toujours dans ces cas.

Bien que les classements soient statistiquement très similaires, il y a toutefois des différences notamment dans les Top 10 (figure 4.5). Ces différences « locales » ne sont pas négligeables dans le cas de mesure de PageRank peu homogènes. Par exemple, pour *PORTS*, les modèles VON accordent tous deux une place plus importante au port de Pusan par rapport à ceux de Hong Kong ou Rotterdam.

4.5 Discussion et Perspectives

Dans ce chapitre, nous avons étudié l'utilisation de la mesure PageRank pour évaluer l'importance des états dans les réseaux d'ordre supérieur. Nous avons montré que, bien que ces réseaux puissent être modélisés par des graphes similaires aux $\textsc{Fon}_1$, ne pas tenir compte des différences de représentations des états peut mener à un biais dans la mesure. En effet, l'hétérogénéité du nombre de représentations des états dans les HON peut avoir un impact sur la valeur de PageRank en renforçant les états avec plus de représentations. Nous proposons une correction de la mesure PageRank, qui oblige le surfeur à se téléporter uniquement vers les nœuds de premier ordre.

Nous avons ensuite analysé l'impact de la correction du biais et du choix du modèle présenté dans le chapitre 3 sur la mesure. Les analyses expérimentales montrent toutefois que les différences de classements obtenus à partir des mesures PageRank ne sont pas si importantes. Toutefois, la mesure PageRank peut être utilisée dans le cadre d'algorithme de clustering de graphe pour comparer les qualités des partitions [69]. Dans le chapitre suivant, vous verrons que les différences de PageRank peuvent avoir un impact plus important dans ce cadre.

De plus, la mesure de centralité PageRank a d'autres applications. Une méthode récente basée sur la matrice Google (la matrice stochastique qui modélise le surfeur aléatoire), appelée *matrice Google réduite*, a montré son efficacité pour déduire les liens cachés entre un ensemble de nœuds d'intérêt [29], par exemple en étudiant les réseaux Wikipedia [20]. En utilisant les traces des utilisateurs sur le site web, nous pourrons également étudier la généralisation de cet outil aux HONs.

Ce chapitre s'est focalisé sur la mesure PageRank. Il existe toutefois différentes mesures reposant sur les marches aléatoires qui pourraient être appliquées aux réseaux d'ordre supérieur. On peut notamment évoquer la centralité « de second ordre » développée par Kermarrec *et al.* [46]. Dans cette approche, l'importance d'un nœud dépend du temps qu'un marcheur aléatoire met à revenir à ce nœud. Le « second ordre » n'est pas ici lié à la longueur de séquences mais correspond au moment d'ordre 2 de la distribution de ces temps de retour. Un avantage de cette mesure est de ne pas impliquer de mécanisme de téléportation, il serait donc intéressant de la généraliser aux HON en tenant compte du temps de retour du marcheur à un *état* donné.

Chapitre 5

Partitionnement des réseaux d'ordre supérieur

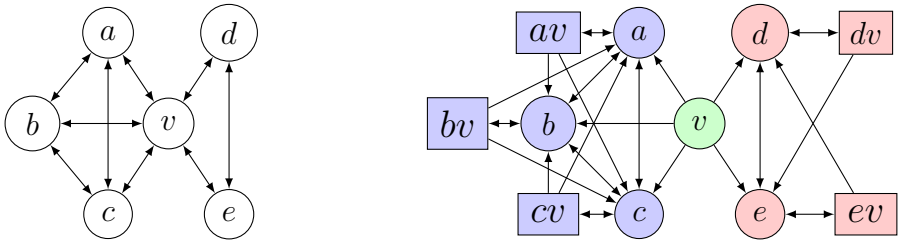
5.1 Introduction

Les algorithmes de *clustering* sont des outils répandus dans l'analyse et la fouille de réseau. Le *clustering* de graphe permet de grouper les nœuds en *cluster*, communauté ou bien module, afin que les nœuds du *cluster* soient plus proches entre eux qu'avec les autres nœuds du graphe. Ce sont des outils particulièrement utiles lorsque les réseaux construits à partir de données réelles indiquent la présence de communautés sous-jacentes. On parle alors de problèmes de « détection de communautés ». Trouver ces communautés permet notamment d'étudier les comportements du réseau plus facilement. Nous nous intéressons ici à ces méthodes, qui doivent être différenciées des problèmes de découpage de graphe en un nombre pré-déterminé de groupes.

Il existe une littérature très importante sur la détection de communautés et le *clustering* de graphe. D'après Dao *et al.* [24], il existe plusieurs raisons pour lesquelles les résultats de détection de communautés varient tant entre les méthodes ; des notions différentes de proximité de nœuds, d'hypothèse sur la structure d'une communauté, malgré une définition de structure de communauté similaire une différence de formalisme et d'algorithmique qui peut impacter le résultat final, un compromis temps/complexité qui diffère d'auteurs à d'autres et qui influe sur les résultats, etc. Le choix de la méthode qui correspond au système étudié est en soit un problème difficile ; aucune réponse et choix ne saura satisfaire l'intégralité des cas d'usage.

Une hypothèse souvent posée en détection de communautés est que les nœuds appartiennent uniquement à une communauté. C'est alors une *partition* des nœuds qui est recherchée. Cependant, cette simplification ne fait pas état du cas où les nœuds peuvent avoir plusieurs rôles dans différentes communautés. Dans ces cas, un *clustering chevauchant* peut s'avérer plus fidèle à la dynamique sous-jacente de certains réseaux, notamment ceux construits à partir de données réelles. Des exemples se retrouvent dans des domaines aussi divers que la représentation de textes [61], le traitement de données médicales [47] ou le marketing [3]. Dans la suite, nous utiliserons surtout le terme « *clustering* » pour parler de *clusterings* chevauchants et le terme « partitionnement » dans le cas où on impose l'absence de superposition de communautés.

Dans le cadre des données séquentielles qui nous intéressent, on peut également poser l'hypothèse que des communautés d'états peuvent se chevaucher mais également que la dynamique des flux est liée à cette structure communautaire. Sous ces hypothèses, l'utilisation des Hon peut permettre une recherche plus efficace de ces structures. En effet, différentes représentations d'un même état peuvent être partagées entre différents groupes. Cela correspond à l'idée que ces représentations encodent des comportements différents.



(a) Réseau d'ordre 1 (b) Réseau VON$_2$

FIGURE 5.1 – Exemple de dynamique de flux avec deux cliques. Supposons un graphe **(a)** composé de deux cliques $\{v, a, b, c\}$ et $\{v, d, e\}$. Les séquences $\mathcal{S}$ sont construites comme suit : un marcheur choisit au hasard n'importe quelle arête sortante. Cependant, s'il atteint v, il retournera à un autre noeud de la clique d'où il vient. Le réseau VON$_2$ **(b)** comprend alors une représentation de v pour chaque autre nœud et toutes les arêtes quittant le même nœud ont le même poids. La couleur indique la partition qui devrait être trouvée par l'algorithme de partitionnement de graphe. En projetant sur les états, on obtient le *clustering* chevauchant $\{\{a, b, c, v\}, \{v\}, \{v, d, e\}\}$. L'état v est présent dans 3 *clusters* (1 est trivial).

À partir d'un partitionnement des nœuds $\mathcal{V}$, on aboutit alors à un *clustering* chevauchant des états $\mathcal{A}$.

Illustrons cette idée avec l'exemple donné dans la figure 5.1b : les différentes représentations de l'emplacement v capturent parfaitement la dynamique du flux (*i.e.* le marcheur ne peut jamais quitter une clique donnée). Un algorithme de partitionnement devrait renvoyer la partition des nœuds correspondant aux trois couleurs formant trois composantes fortement connexes. À partir de cette partition, nous pouvons construire un *clustering* en assignant à chaque état l'ensemble des *clusters* où au moins une de ses représentations a été trouvée, résultant en un *clustering* chevauchant des *états*.

Ces observations ont déjà été notées par les chercheurs ayant développé des modèles de HON. Dans ces cas, un algorithme de partitionnement basé sur des marches aléatoires (*Infomap*) a été utilisé avec peu [70] ou aucune modification [84]. À l'instar du chapitre 4, ce chapitre vise à examiner cette approche dans le cas du *clustering*. Nous analysons en particulier les différences de résultat de *clustering* en fonction de la méthode de construction de réseaux. Nous pouvons en effet imaginer des modèles HON plus petits qui peuvent aussi bien représenter la dynamique de flux. Même s'ils contiennent des caractéristiques similaires, nous montrons que, même dans le cas où la dynamique de flux est modélisée à l'aide d'un processus de second ordre, la différence dans le nombre de représentations des *états* a des effets importants sur les résultats de *clustering*.

Dans la section 5.2, nous commençons par faire un état de l'art des algorithmes de partitionnement et de *clustering* de graphe en insistant sur l'applicabilité des méthodes

aux réseaux d'ordre supérieur. Nous détaillons par la suite l'algorithme *Infomap* dans la section 5.3 et discutons des principaux problèmes liés à son utilisation sur les HON dans la section 5.4. Nous proposons dans cette dernière section un modèle de HON agrégé nommé AGG-VON. L'intérêt d'utiliser un tel modèle est montré dans la section 5.5 sur des benchmarks synthétiques. Nous montrons par ailleurs que le *clustering* de ce modèle mène à des résultats significativement différents sur des données réelles. Ces résultats ainsi que des pistes pour des recherches futures sont discutées dans la section 5.6.

Nos résultats renforcent l'idée déjà suggérée dans le chapitre précédent, à savoir que, même si une dynamique d'ordre supérieur peut être encodée dans un graphe pondéré classique, les outils d'exploration conçus pour les réseaux sans mémoire peuvent ne pas être adaptés pour capturer cette dynamique. Les futurs algorithmes de réseau devraient plutôt être spécifiquement conçus pour prendre en compte la multiplicité des représentations des réseaux d'ordre supérieur.

5.2 État de l'art

Nous allons nous intéresser, dans la section 5.2.1, à la littérature abondante concernant les méthodes de *clustering* de graphe « classiques ». Dans la section 5.2.2, nous constatons la grande différence dans le nombre de travaux disponibles entre les réseaux d'ordre supérieur markovien et les autres. Nous discuterons également des méthodes d'évaluation des *clustering* (section 5.2.3).

5.2.1 Partitionnement et *clusterings* chevauchants des graphes $\mathbf{Fon}_1$

Il est très compliqué de faire un panorama complet des méthodes de *clustering* de graphe, le domaine ayant fait l'objet d'un nombre incroyable de travaux au fil du temps se basant sur des méthodes et de mesures différentes. Il existe toutefois plusieurs études comparatives des méthodes de *clustering*, on pourrait notamment parler de l'étude de Fortunato [27] ou bien du travail de Dao *et al.* [24]. Ces derniers font un état des lieux plus exhaustif du domaine ainsi que des expériences très approfondies. Néanmoins, nous nous attachons ici à présenter les méthodes usuelles de partitionnement et de *clustering* chevauchant afin de discuter de leur intérêt dans le cadre de la prise en compte des dépendances d'ordre supérieur.

Il est tout d'abord difficile de parler de la détection de communautés sans mentionner une approche très répandue à savoir l'optimisation de la mesure de modularité. La

modularité permet d'évaluer la qualité d'un partitionnement de graphe en comparant les connections directes intra-module (*cluster*) par rapport aux connections directes avec le reste du réseau. L'idée est de faire la différence entre la proportion d'arêtes à l'intérieur d'un *cluster* par rapport à la proportion d'arêtes qu'on s'attendrait à trouver dans un réseau aléatoire avec la même distribution du degré. L'optimisation de cette mesure permet ainsi de trouver des modules sans en connaître le nombre *a priori* ce qui est un élément important dans le cadre de la détection de communautés.

La maximisation de la modularité est un problème difficile [13] et des heuristiques sont employées afin d'obtenir une solution approchée. Dans ce cadre, la méthode dite « de Louvain » [10] est sans doute l'approche la plus répandue et permet de traiter ce problème pour des graphes de très grande taille. Comme présenté dans le chapitre 3, l'intérêt principal des HON repose sur la prise en compte des dépendances séquentielles. Or, la modularité n'évalue que le flux direct entre nœuds. Il existe cependant d'autres mesures de qualité de partition comme celle de l'algorithme *Infomap* que nous allons présenter plus en détail dans la section 5.3. Bien que la mesure optimisée (la *Map Equation*) soit *a priori* très différente de la modularité, l'algorithme d'optimisation est proche de la méthode de Louvain.

Comme souligné dans la section 5.1, il existe des méthodes permettant de trouver des nœuds en superposition sur des *clusters* : les *clusterings* chevauchants. Les travaux de Baadel *et al.* [4] donnent quelques méthodes existantes pour identifier des *clusterings* chevauchants. Une autre étude de Xie *et al.*[83] fait un état de l'art plus complet du domaine, notamment sur les algorithmes mais aussi les méthodes d'évaluations de résultats.

Dans le reste du chapitre nous nous intéressons à l'algorithme *Infomap*, Rosvall *et al.* [70] ont généralisé cette méthode à la recherche de *clusterings* chevauchants avec le *Fuzzy Infomap* [80]. Si on considère des exemples simples tels que la figure 5.1a, on pourrait considérer que la recherche d'un *clustering* chevauchant dans FON_1 suffit pour trouver une solution satisfaisante. Il est toutefois possible de concevoir des exemples où la séparation est moins claire sans tenir compte des dépendances séquentielles. Au demeurant, comparer les résultats de partitionnement des HON à des *clusterings* chevauchants de FON_1 pourrait faire l'objet d'une recherche plus approfondie dans le futur.

5.2.2 *Clustering* de réseaux d'ordre supérieur

Comme évoqué dans la section 5.1, il existe peu d'applications de *clustering* sur les modèles de réseaux markoviens (celles-ci sont évoquées dans la section 5.2.2.2). Cependant, les travaux sur les autres réseaux d'ordre supérieur non-markoviens sont bien plus courants

(section 5.2.2.1). La suite ne saurait représenter la diversité des travaux sur le *clustering* de réseaux d'ordre supérieur. Notre but est de montrer le contraste entre les démarches.

5.2.2.1 *Clustering* de modèles non-markoviens

Les travaux sur le *clustering* d'hypergraphes sont nombreux bien qu'il n'existe pas, à notre connaissance, de revue détaillée de ce sujet. On peut toutefois noter qu'une approche courante consiste à transformer l'hypergraphe en un graphe « classique », puis d'y appliquer des méthodes de *clustering* connues [1]. Par exemple, Zhou *et al.* [85] se base sur une généralisation de la méthode spectrale aux hypergraphes. Ils commencent par construire un graphe à partir de l'hypergraphe en utilisant l'expansion de clique (un hyperlien donnera une clique). Puis, les auteurs utilisent une généralisation aux hypergraphes de la *Normalized Cut* (une mesure populaire pour évaluer la segmentation d'image).

Benson *et al.* [7] fournissent un framework pour le *clustering* tenant compte des motifs dans un graphe. Dans ce travail, ils définissent une contrainte : le partitionnement de nœuds doit éviter de couper les motifs dont font partie les nœuds. De ce fait, le but est de trouver un ensemble de nœuds qui minimise la *conductance du motif* (c'est-à-dire le rapport entre les motifs coupés et les nombres de motifs au total).

Il apparaît que nombre de méthodes de *clustering* des réseaux d'ordre supérieur cherchent, à l'instar des travaux évoqués ci-dessus, à se ramener à un problème de *clustering* de graphe « simple ». Toutefois, la recherche et l'évaluation des solutions tient compte de l'« ordre supérieur » rajoutée. Ceci est à contraster avec la littérature sur le *clustering* de réseau d'ordre supérieur markovien.

5.2.2.2 *Clustering* de modèles markoviens

En considérant la récence du domaine, il n'est pas étonnant de trouver peu de travaux sur la détection de communautés adaptée et appliquée exclusivement aux réseaux Hon. Toutefois, à la différence des approches évoquées dans la section précédente, les algorithmes de *clustering* connus sont appliqués directement aux Hon (car ils sont structurellement similaire à des réseaux « classiques ») avec peu ou pas d'adaptations.

Dans la littérature, l'algorithme *Infomap* a été appliqué à Von [84] construit à partir d'un jeu de données issu de la même source que *PORTS* mais sur une période différente. Une application développée par les auteurs est la prédiction de l'invasion d'un écosystème par des espèces aquatiques non-indigènes. Le rejet d'eau de ballast ou bien du *biofouling* (*e.g.* accumulation de micro-organismes sur la coque des navires) constitue en effet un facteur important de ces changements. Dans ce cadre, le fait de trouver des *clusters* de

ports très connectés entre eux permet de révéler les endroits où il est probable que des espèces qui ne sont pas autochtones soient introduites à cause des échanges maritimes. Une différence identifiée dans les résultats avec l'utilisation de VON est la place des ports internationaux, qui ont beaucoup plus de représentations et font donc partie de plusieurs *clusters* différents. À l'inverse il est moins probable d'observer des invasions dans les ports locaux au vu du nombre plus limité d'états dans des *clusters* différents.

Cette application a été par la suite développée, en partie avec les mêmes auteurs, par Saebi *et al.* [72]. Cette fois, le VON construit tient compte d'un risque d'invasion entre un port A et B dépendant de la probabilité d'avoir des espèces non indigènes à B dans A (en fonction de l'écorégion de deux ports) ou la probabilité que ces espèces présentes en A s'établissent dans la zone de B. Ce risque d'invasion est couplé aux données séquentielles pour construire un réseau qui représente les trajets de bateau mais avec le risque d'invasion comme probabilité de transition. Les résultats du *clustering* permettent de déterminer quel port est associé à des flux d'espèces non indigènes plus important.

Le *clustering* des réseaux SN_2 avec l'algorithme *Infomap* a été également envisagé [70]. Les auteurs notent que la présence de la mémoire, même uniquement jusqu'à l'ordre 2, affectent la répartition et le chevauchement. Ils évoquent notamment un possible biais lié à l'utilisation directe de l'algorithme. Ce biais ainsi que la correction proposée sont discutés dans la section 5.4. Les *clusters* trouvés par leur méthode sont plus petits mais plus informatifs, révélant des comportements qui n'étaient pas explicites dans les résultats sur les réseaux FON_1. Le *clustering* appliqué aux données *AIR* (sur une période différente de la nôtre) permet d'identifier une différence entre les aéroports de transit et les aéroports qui permettent aux voyageurs de rentrer chez eux.

Dans la section 5.4, nous étudierons le choix de la sélection de modèle d'ordre supérieur sur les résultats de *clustering* de l'algorithme *Infomap*. En particulier, nous étudions l'effet de l'utilisation d'un modèle plus clairsemé (*sparse*) obtenu en fusionnant les nœuds mémoire ayant des probabilités de transition similaires. Une idée similaire a été suggérée précédemment dans les travaux de Jaaskinen *et al.* [41], ce qui a donné lieu à un modèle appelé *« sparse Markov chain »* (Chaîne de Markov clairsemée). L'objectif des auteurs était d'améliorer le taux de compression et la classification des séquences d'ADN et des données protéiques.

5.2.3 Évaluations des résultats de *clustering* de graphe

Cependant, il n'existe pas de métrique universelle pour comparer des algorithmes de *clustering*. Une approche courante est de comparer le résultat d'un algorithme sur des

benchmarks, c'est-à-dire des graphes dont on connaît déjà la structure communautaire (la « vérité terrain »). Dans cette optique, Girvan et Newman [31] ou Holland [37] proposent de générer des graphes aléatoires avec une structure communautaire.

Un autre *benchmark* sur lequel nous allons baser nos expériences est le benchmark *LFR* [54] développé par Lancichinetti *et al.*. Les auteurs soulèvent en effet que le *benchmark* de Newman et Girvan[31] ne propose que des réseaux avec des degrés similaires, des communautés de même taille et des réseaux globalement petits. De ce fait, il est difficile de considérer ce type de *benchmark* comme une bonne approximation de réseaux réels et donc, de l'utiliser pour tester si un algorithme donne un résultat de qualité. Le *benchmark* LFR se rapproche des *vrais* réseaux, avec une distribution de degrés et de taille de communautés hétérogènes. Il permet aussi de créer des réseaux des nœuds qui appartiennent à plusieurs communautés, ce qui permet aussi d'évaluer les *clusterings* chevauchants.

Il n'existe toutefois pas de *benchmark* pour évaluer le *clustering* d'un réseau avec une dynamique de flux suivant une structure communautaire. Pour résoudre ce problème, il sera nécessaire de concevoir un protocole *ad hoc*, qui permet de créer des réseaux qui suivent la dynamique de flux similaire à la figure 5.1. Nous proposons un protocole allant dans ce sens dans la section 5.5.1. Notre approche est toutefois limitée à une dynamique de flux limitée à l'ordre de 2. Il serait intéressant de la généraliser afin d'inclure des dynamiques différentes.

5.3 Algorithme de Partitionnement *Infomap*

Dans le reste du chapitre, nous utilisons l'algorithme *Infomap* [69]. C'est une méthode de clustering basée sur la marche aléatoire qui maximise la *Map Equation*. Notre but étant d'évaluer l'utilisation de cette méthode pour le clustering des réseaux d'ordre supérieur nous en donnons ici une description détaillée mais pas complète.

L'algorithme se base sur le principe de « Longueur de description minimale » (*MDL, Minimum Description Length*), permettant de minimiser la longueur de l'information permettant d'encoder les données. La partition en communautés devient un problème de compression visant à minimiser le nombre de bits nécessaires pour écrire une marche aléatoire sur le graphe. La qualité du codage va correspondre au nombre moyen de *bits* par pas effectué. Une façon de faire est d'utiliser le codage de Huffman, qui assigne des codes binaires plus courts aux nœuds souvent visités et longs aux nœuds rarement visités.

Ce codage peut toutefois être amélioré en utilisant un encodage à deux niveaux en attribuant des codes binaires d'entrée/sortie aux clusters et aux nœuds. La meilleure par-

tition de $\mathcal{V}$ correspond alors au meilleur encodage à deux niveaux des marches aléatoires dans G.

Pour un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, les codes assignés aux nœuds seront fonction de la probabilité stationnaire d'un marcheur aléatoire. Celle-ci est estimée avec les valeurs de *PageRank* $\{\pi_n\}_{n\in N}$. Toutefois il est inutile dans notre cas de chercher vraiment ce codage optimal. En effet, d'après le théorème de Shannon [77], la longueur optimale en moyenne par pas est donnée par $H(\{\pi_n\}_{n\in N})$, l'entropie de ces probabilités stationnaires. La *Map Equation* $L(\mathcal{C})$ (aussi appelé *code length*) correspond à cet optimun dans le cas d'un codage à deux niveaux utilisant la partition des nœuds $\mathcal{C} = (C_1, C_2, \ldots, C_m)$:

$$L(\mathcal{C}) = q_{\curvearrowright} H(\mathcal{C}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{C}_i) \tag{5.1}$$

La valeur de $L(\mathcal{C})$ est décomposée en une somme de deux termes : premièrement, $q_{\curvearrowright} H(\mathcal{C})$ le nombre attendu de bits utilisés pour le code d'entrée/sortie des clusters et deuxièmement $\sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{C}_i)$, le nombre attendu de bits pour les mots de code associés aux nœuds.

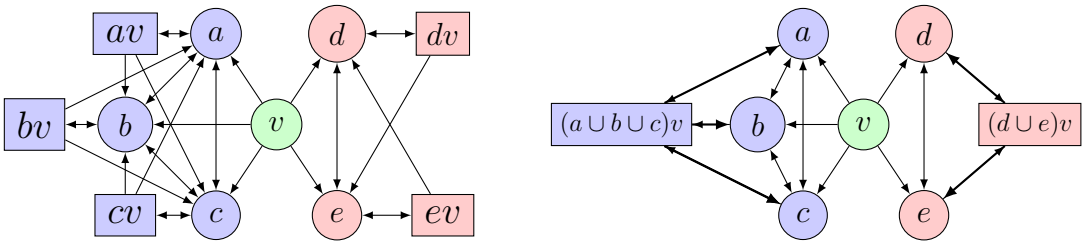
L'algorithme *Infomap* vise à minimiser la fonction L à l'aide d'une procédure rapide et gloutonne à plusieurs niveaux qui suit le schéma suivant, qui ressemble à la méthode de Louvain évoquée dans la section 5.2. La première phase de l'algorithme correspond à la trame suivante :

1. Chaque nœud est assigné à son propre groupe.
2. — Dans un ordre aléatoire, chaque nœud est déplacé vers le cluster voisin si le résultat donne la plus grande diminution de la valeur $L(\mathcal{C})$. Cependant, s'il n'y a pas d'augmentation, le nœud reste dans son groupe d'origine.

   — L'étape précédente est réappliquée jusqu'à ce que plus aucune diminution de la valeur ne soit constatée.
3. Les clusters sont alors considérés comme les nœuds d'un nouveau graphe.
4. Répéter (1) jusqu'à ce qu'un minimum de $L(\mathcal{C})$ soit atteint.

La seconde phase consiste en une succession d'ajustements. Le premier ajustement permet de déplacer les nœuds entre les clusters. Le deuxième permet de considérer un cluster comme un nouveau réseau et d'y appliquer à nouveau la première partie d'*Infomap*, le but étant d'améliorer le résultat et d'éviter les minima locaux. Pour les différentes expériences, nous avons utilisé la bibliothèque python *Infomap* développée par Rosvall *et al.* [1].

1. `www.mapequation.org/infomap/` (version 1.3.0)



(a) Réseau Von$_2$ (b) Réseau Min-Von$_2$

FIGURE 5.2 – Illustration du modèle Min-Von$_2$ reprenant l'exemple de la figure 5.1. Dans le réseau **(b)**, les représentations de v sont fusionnées conformément à la dynamique de flux connue et ne comprennent que trois représentations de v. Ce réseau divise également les sommets en deux composantes fortement connectées.

5.4 Clustering de Hon en utilisant *Infomap*

Comme indiqué précédemment, il a été suggéré que nous pouvions directement appliquer l'algorithme Infomap sur D_{KL}-Von [84] (présenté dans la section 3.2.2.2) sans aucune adaptation *i.e.* nous pouvons, dans ce contexte, le considérer comme un réseau de premier ordre. Cependant, comme nous l'avons vu avec la mesure *PageRank* dans le chapitre 4, la multiplicité des représentations peut avoir des effets importants sur l'algorithme.

Nous allons ici montrer que c'est également le cas pour le clustering utilisant l'algorithme Infomap en nous appuyant sur un cas de figure impliquant un clustering connu et une dynamique de flux relativement simple. Il sera par la suite utilisé dans des expérimentations (section 5.5.1). Dans l'exemple introductif (figure 5.1), un autre réseau peut être construit avec moins de représentations (voir figure 5.2b). Nous appelons cet autre modèle d'ordre supérieur *minimal* et le nommons Min-Von$_2$ (défini ci-dessous).

Définition 5.1 (Réseau Minimal Min-Von$_2$)**.** *Soit un clustering chevauchant $\mathcal{C}$ sur l'ensemble d'états $\mathcal{A}$ et un réseau Von$_2$ $\mathcal{G}$, le réseau minimal Min-Von$_2$ de $\mathcal{G}$ est obtenu en fusionnant, pour chaque $y \in \mathcal{A}$ et chaque groupe $C \in \mathcal{C}$, l'ensemble de nœuds $\{xy \in \mathcal{V}_2, x \in C\}$ en un nœud-mémoire composite représentant l'évènement "le marcheur a visité un des états $\{x \in C\}$ avant d'aboutir à l'état y".*

Si on suppose que le réseau d'ordre variable 2 (figure 5.2a) modélise une dynamique de flux similaire à celle détaillée en introduction, alors Min-Von$_2$ est un modèle aussi précis mais plus parcimonieux de cette dynamique. En effet, si l'état suivant un état chevauchant est purement déterminé par le *groupe* visité à l'étape précédente alors on peut juste considérer des contextes fusionnés par groupe plutôt qu'énumérer toutes les possibilités.

Dans ce cadre, on peut s'attendre à ce que le résultat d'un algorithme corresponde au même clustering par chevauchement pour Von$_2$ et Min-Von$_2$. Dans le cas général, nous

soutenons que le nombre de représentations « supplémentaires » du premier a trois effets potentiels que nous détaillons ici.

1- Effet sur le nombre de codes par groupe

Lorsque Infomap est appliqué à un réseau d'ordre supérieur, deux représentations du même état appartenant aux mêmes groupes reçoivent des codes différents. Cela peut rendre plus difficile la détection de groupes contenant de nombreuses représentations du même état. Ce problème a déjà été identifié pour les réseaux SN_2 [70]. Les auteurs ont proposé une modification d'Infomap afin d'attribuer le même code aux représentations d'un même état. En effet, plus un état est représenté dans un cluster, plus la contribution de ce cluster à la *Map Equation* sera importante. Par exemple, dans le réseau VON_2 de la figure 5.1b, nous devrions utiliser trois codes pour chacun des nœuds $\{av, bc, cv\}$.

2- Effet sur le taux d'utilisation des codes

Les probabilités stationnaires d'un marcheur aléatoire sont utilisées pour calculer le taux d'utilisation des codes dans la *Map Equation*. Cette probabilité stationnaire est donnée par le *PageRank* des nœuds. Or, comme vu dans le chapitre 4, le mécanisme de téléportation va créer des divergences dans les valeurs de *PageRank* selon que l'on utilise le modèle VON_2 ou MIN-VON_2. Cela est vrai même en supposant que le premier effet soit corrigé *i.e.* nous utilisons un code similaire pour les représentations d'un lieu qui appartient aux mêmes grappes. Dans l'exemple de la figure 5.1, en supposant que nous corrigions le premier effet, la probabilité qu'un marcheur aléatoire utilise le code associé aux nœuds (av, bc, cv) après une téléportation est de $\frac{3}{10} = 0,3$ pour VON_2 et de $\frac{1}{8} = 0,125$ pour MIN-VON_2.

Ce second effet peut donc également rendre difficile l'identification de clusters plus importants. En effet, plus un cluster contient de représentations, plus ses codes d'entrée, de sortie et de nœuds qu'il contient sont utilisés. Dans le modèle minimal, les taux de π sont également biaisés en raison du nombre différent de représentations, mais à un degré moindre. Même si un surfeur aléatoire est plus susceptible de visiter une représentation d'un lieu appartenant à différents groupes, il n'y a, par définition, qu'une seule représentation de chaque état par groupe.

3- Effet sur l'exploration de l'espace de solution

L'algorithme *Infomap* suit une procédure gloutonne, en commençant par une partition où chaque représentation est assignée à un seul cluster. L'algorithme *Infomap* doit donc effectuer davantage de fusions dans VON_2 par rapport à MIN-VON_2. Le risque pour l'algorithme de tomber sur des minima locaux est donc plus élevé.

Nous concluons qu'il y a plusieurs raisons pour lesquelles nous devrions essayer de comparer les *clusterings* obtenus par VON_2 et ceux obtenus sur un modèle de réseau plus parcimonieux. Le fait que la *Map Equation L* soit affectée par le nombre de représentations

utilisées ne signifie pas que les résultats d'*Infomap* seront moins bons. Cependant, les expériences détaillées dans la section 5.5.1 montrent que c'est le cas même lorsque l'on ne considère qu'une dynamique de second ordre simple comme celle de notre exemple introductif (figure 5.1).

5.4.1 Modèle agrégé Agg-Von$_2$

Dans la section précédente, le modèle minimal discuté est construit à partir d'un clustering sous-jacent connu. Afin d'évaluer la pertinence des modèles parcimonieux dans des études de cas réelles, nous définissons ici un modèle agrégé de VON$_2$ appelé AGG-VON$_2$. L'hypothèse sous-jacente que nous utilisons pour nous rapprocher d'un modèle minimal est que les représentations ayant des probabilités de transition de sortie similaires appartiendront aux mêmes groupes. Cela apparaît clairement dans l'exemple de la figure 5.2. Comme pour l'étude de cas précédente, nous supposons que la dynamique du flux est capturée à un ordre maximum de 2. Nous discuterons davantage cette hypothèse dans la section 5.6.1. Nous définissons d'abord le concept de fusion des représentations et introduisons ensuite les critères permettant ces fusions pour aboutir à la définition de l'algorithme 3.

Définition 5.2. ***Représentation fusionnée*** *Pour un état $v \in \mathcal{A}$, nous appelons représentation fusionnée un sous-ensemble $X \subseteq \mathcal{V}_2(v)$ et pour $\sigma \in \mathcal{A}$ nous définissons $c(X\sigma) = \sum_{x_1x_2 \in X} c(x\sigma)$ (i.e. le nombre de fois que σ est observé après un élément de X) et*

$$p(\sigma|X) = \frac{c(X\sigma)}{\sum_{\sigma' \in \mathcal{A}} c(X\sigma')} \tag{5.2}$$

comme la probabilité de transition de X vers σ. Comme dans la définition 2.6, P_X (respectivement C_X) désigne la distribution de probabilité (respectivement les occurrences) associée aux symboles suivant une séquence quelconque dans X.

Cette forme généralisée de probabilité de transition peut être interprétée comme la probabilité d'arriver à l'emplacement σ compte tenu du fait que nous nous trouvons à l'emplacement v après avoir visité *l'un* des emplacements x tels que $xv \in X$.

Définition 5.3. ***Agrégation possible.*** *Soit X, Y deux représentations fusionnées disjointes de v. On dit que (X, Y) peuvent être fusionnées ou $X \boxplus Y$ si toutes les conditions suivantes sont pertinentes*

$$D_{KL}\left(P_X||P_{X\cup Y}\right) < \frac{2}{\log_2(c(X)+1)} \tag{5.3}$$

$$D_{KL}\left(P_Y||P_{X\cup Y}\right) < \frac{2}{\log_2(c(Y)+1)} \tag{5.4}$$

$$D_{KL}\left(P_{X\cup Y}||P_v\right) > \frac{2}{\log_2(c(X\cup Y)+1)} \tag{5.5}$$

Un exemple de fusion possible est donné dans la figure 5.2 avec la fusion des représentations dv et ev en une seule représentation $(d \cup e)v$. Le but des conditions ci-dessus est de réutiliser le critère de « pertinence » défini pour le modèle VON (Eq. 3.4). En effet, nous avons $dv \boxplus ev$ lorsque savoir que le marcheur vient de d ou e est pertinent (condition 5.5), mais que l'information supplémentaire « il vient en fait de d (ou e) » ne l'est pas (conditions 5.3 et 5.4).

Théorème 5.1. ***Non-Transitivité des fusions.*** *Soit X, Y, Z des représentations fusionnées disjointes de v alors*

$$(X \boxplus Y) \wedge (Y \boxplus Z) \nRightarrow (X \boxplus (Y \cup Z)) \tag{5.6}$$

Démonstration. Nous pouvons construire un contre-exemple

$$\begin{aligned} C_X &= (n, 0, 0, \ldots, 0) \\ C_Y &= (n, n, 0, \ldots, 0) \\ C_Z &= (0, n, 0, \ldots, 0) \\ C_v &= (2n, 2n, N, 0, \ldots, 0) \end{aligned}$$

En considérant que $n \ll N$, nous pouvons supposer que la condition (5.5) est toujours vérifiée. Nous avons $D_{KL}(P_X, P_{X\cup Y\cup Z}) = 1$ donc pour tout $n > 2$ nous avons $\neg\,(X \boxplus (Y \cup Z))$. Cependant, on a

$$\begin{aligned} D_{KL}(P_X, P_{X\cup Y}) &= D_{KL}(P_Z, P_{Y\cup Z}) = \log_2(3) - 1 \\ D_{KL}(P_Y, P_{X\cup Y}) &= D_{KL}(P_Y, P_{Y\cup Z}) = \log_2(3) - \frac{3}{2} \end{aligned}$$

Par conséquent, pour $2 < n \leq 9$, $(X \boxplus Y) \wedge (Y \boxplus Z)$ est vrai tandis que $(X \boxplus (Y \cup Z))$ est faux. □

Une conséquence du théorème 5.1 est qu'un ensemble minimum de représentations fusionnées ne peut pas être trouvé en effectuant itérativement toutes les fusions possibles puisque les résultats peuvent être arbitraires. Nous utilisons donc une procédure d'agrégation hiérarchique qui donnera la priorité aux fusions de représentations les plus similaires

(en termes de distance par rapport à leur distribution d'union). Cependant, comme toutes les représentations ne peuvent pas être fusionnées, la procédure d'agrégation n'a pas besoin de condition d'arrêt et le nombre de représentations fusionnées renvoyées dépend des seuils de définition 5.3. Cette opération est donc sans paramètre. Son inconvénient est similaire à celui lié à la construction des réseaux VON : elle repose sur la définition de la pertinence selon [71].

Require: $\mathcal{V}_2(v)$ (représentations d'ordre 2de v)
Ensure: $\mathcal{R}$ (partition de $\mathcal{V}_2(v)$)
1: $\mathcal{R} \leftarrow \{X \in \mathcal{V}_2(v)\}$
2: $M \leftarrow \{(X,Y) \in \mathcal{R} \times \mathcal{R} : X \boxplus Y\}$
3: **while** $M \neq \emptyset$ **do**
4: $\quad (X,Y) \leftarrow argmin_{(X',Y') \in M} D_{KL}\left(P_{X'} || P_{Y' \cup Y'}\right) + D_{KL}\left(P_{Y'} || P_{X' \cup Y'}\right)$
5: $\quad \mathcal{R} \leftarrow \mathcal{R} \setminus X \setminus Y \cup (X \cup Y)$
6: $\quad M \leftarrow \{(X,Y) \in \mathcal{R} \times \mathcal{R}, X \boxplus Y\}$
7: **end while**
8: **return** $\mathcal{R}$

Algorithme 3 : Agrégation des représentations de second ordre de v

La procédure exacte est détaillée dans l'algorithme 3. L'algorithme utilisé est similaire à un *clustering* hiérarchique [56] avec deux différences principales. Premièrement, tester $x \boxplus y$ et calculer la similarité entre x et y a une complexité temps de $O(|\mathcal{A}|)$. En supposant que $N = |\mathcal{V}_2(v)|$ et, étant donné que $N \leq |\mathcal{A}|$, la complexité en temps de l'algorithme 3 est donc de $O(N^2|\mathcal{A}|)$. Deuxièmement, comme toutes les fusions ne sont pas possibles, l'ensemble M peut être peu dense et nécessite $O(N^2)$ d'espace. La complexité de l'espace de la procédure correspond donc à $O(N|\mathcal{A}|)$ nécessaire pour stocker les valeurs d'occurrences de symboles suivants c. Les temps de calcul pour des ensembles de données réels sont examinés dans la section 5.5.2.

La troisième condition (équation 5.5) garantit que les représentations fusionnées sont toujours des extensions pertinentes des séquences de premier ordre. Après avoir identifié toutes les représentations fusionnées de chaque emplacement à l'aide de l'algorithme 3, nous construisons AGG-VON$_2$ en fusionnant les nœuds de second ordre de VON$_2$ qui appartiennent au même groupe. Les probabilités de transition des nœuds fusionnés sont définies à l'aide de l'équation 5.2. La fusion des nœuds préserve la propriété 2.1, bien qu'un marcheur aléatoire utilise la dernière estimation des probabilités de transition. En effet, pour une représentation fusionnée X de v et $\sigma \in \mathcal{A}$, le suffixe le plus long s^* dans l'équation 2.8 est similaire pour chaque $xv \in X$ car $s^*\sigma$ est soit σ soit $v\sigma$. De plus, pour $xv \in X$ et $yv \in X$ il n'y a pas $s \in \mathcal{V}$ tel que $(s \rightarrow xv) \in \mathcal{E}$ et $(s \rightarrow yv) \in \mathcal{E}$ car cela signifierait que s est une représentation à la fois de x et de y.

Plusieurs tests doivent être effectués pour évaluer la pertinence de AGG-VON$_2$. Premièrement, le réseau agrégé produit doit être significativement plus petit en terme de nombre de nœuds. Deuxièmement, il doit représenter la dynamique des flux presque aussi bien que le réseau VON$_2$. Cette condition est cruciale puisque l'intérêt d'utiliser le regroupement basé sur la marche aléatoire sur des réseaux d'ordre supérieur était de tirer parti de leur capacité à reproduire les séquences observées. Troisièmement, si ces deux hypothèses sont vérifiées, nous nous attendons à ce que les *clusterings* trouvés sur le réseau agrégé soient différents de ceux obtenus sur le réseau VON$_2$ pour les raisons décrites dans la section précédente.

5.5 Évaluation et résultats

Nous présentons dans cette section les expériences réalisées pour tester les différentes hypothèses formulées dans la section 5.4. Tout d'abord, dans la section 5.5.1, nous démontrons sur des jeux de données synthétiques qu'un VON$_2$ avec un nombre minimal de représentations est plus efficace pour l'identification des groupes connus. Ensuite, dans la section 5.5.2, nous montrons que le modèle AGG-VON$_2$ produit un réseau plus parcimonieux et plus précis. Nous discutons également des différences obtenues avec le *clustering Infomap* sur quatre ensembles de données de séquences réelles (*AIR*, *PORTS*, *WIKI* et *MSNBC*).

5.5.1 Benchmarks synthétiques

Nous nous concentrons sur l'impact du nombre de mots de code par cluster sur les résultats d'*Infomap*. En considérant des réseaux synthétiques, nous mesurons l'effet de la réduction du nombre de représentations dans le résultat du clustering.

Ces expériences sont réalisées à l'aide de graphes aléatoires présentant des clusters et des distributions de degrés de nœuds plus complexes que celles de la figure 5.1a. Nous utilisons le benchmark LFR [54] pour générer des graphes dirigés ainsi qu'un clustering chevauchant. Le but des algorithmes est de retrouver ce clustering. Dans ce benchmark, le paramètre *« Mixing »* correspond au pourcentage d'arcs sortants inter-clusters. Le paramètre *« overlap »* est le pourcentage de nœuds présents dans plus d'un cluster. Le paramètre *« Taille clusters »* indique la taille minimale et maximale des clusters. Les autres paramètres prennent les valeurs suivantes trouvées dans la littérature [83] : $N = 1000$ (nombre de nœuds), $om = 2$ nombre de clusters différents dont les nœuds qui se chevauchent sont membres, $\bar{k} = 10$ (degré moyen), $\max(k) = 50$ (degré maximal), $\tau_1 = 2$ (exposant de la distribution des degrés), $\tau_2 = 1$ (exposant de la distribution de la taille

des clusters).

Afin de comparer les clusters entre eux, nous utiliserons la NMI *(Normalized Mutual information)* [57], qui permet de calculer la similarité entre des *clusterings* chevauchants. Une valeur de 0 correspond à deux *clusterings* dissimilaires et une valeur de 1 à deux *clusterings* identiques. Cette mesure est une correction d'une mesure similaire développée par Lancichinetti et al. [53] qui surestime la similarité entre les *clusterings*.

Nous utilisons une dynamique de flux similaire à l'exemple de la figure 5.2 : si un marcheur venant d'un cluster $\mathcal{C}_i$ arrive à un nœud de chevauchement v dans $\mathcal{C}_i$, il retournera à un voisin aléatoire non chevauchant de v dans $\mathcal{C}_i$. Sinon, il suivra un arc sortant au hasard. Dans cette situation, le marcheur est capable de se déplacer d'un cluster à l'autre en suivant des arcs inter-clusters (leur nombre dépend du paramètre *« Mixing »*). Connaissant le graphe, le clustering « vérité-terrain » et la dynamique des flux, nous pouvons directement construire les réseaux «idéaux» VON$_2$ et MIN-VON$_2$ et essayer de retrouver le regroupement original à l'aide d'*Infomap*. Pour tester l'impact des effets décrits à la fin de la section 5.4.1, nous comparons quatre entrées d'*Infomap* différentes :

— Avec le réseau VON$_2$ sans correction du premier effet décrit dans la section 5.4 *i.e.* l'utilisation de mots de code différents pour toutes les représentations de lieux

— Réseau VON$_2$ lorsque l'on corrige le premier effet *i.e.* en utilisant le même mot de code pour les représentations d'un lieu qui appartiennent aux mêmes clusters

— Réseau MIN-VON$_2$ lorsque les représentations sont fusionnées en fonction des groupes auxquels elles devraient appartenir (comme dans l'exemple de la figure 5.2b). Nous corrigeons également le premier effet.

— Réseau SN$_2$ lorsque l'on corrige le premier effet. Cela correspond à ce qui est utilisé par Rosvall *et al.* [70] dans le cadre du clustering de HON.

TABLE 5.1 – Différence dans le nombre de nœuds dans D_{KL}-VON$_2$ et MIN-VON$_2$ selon les différents paramètres du benchmark LFR

Overlap / Mixing	Taille Clusters	Médiane Nb. Nœuds (VON$_2$ / MIN-VON$_2$ / Diff)
15% / 15%	20 - 50	2248 / 1295 / - 953
	50 - 100	2113 / 1297 / - 816
15% / 30%	20 - 50	1944 / 1293 / - 651
	50 - 100	1916 / 1292 / - 624
30% / 15%	20 - 50	2928 / 1575 / - 1353
	50 - 100	2886 / 1574 / - 1312
30% / 30%	20 - 50	2579 / 1592 / - 1017
	50 - 100	2501 / 1555 / - 946

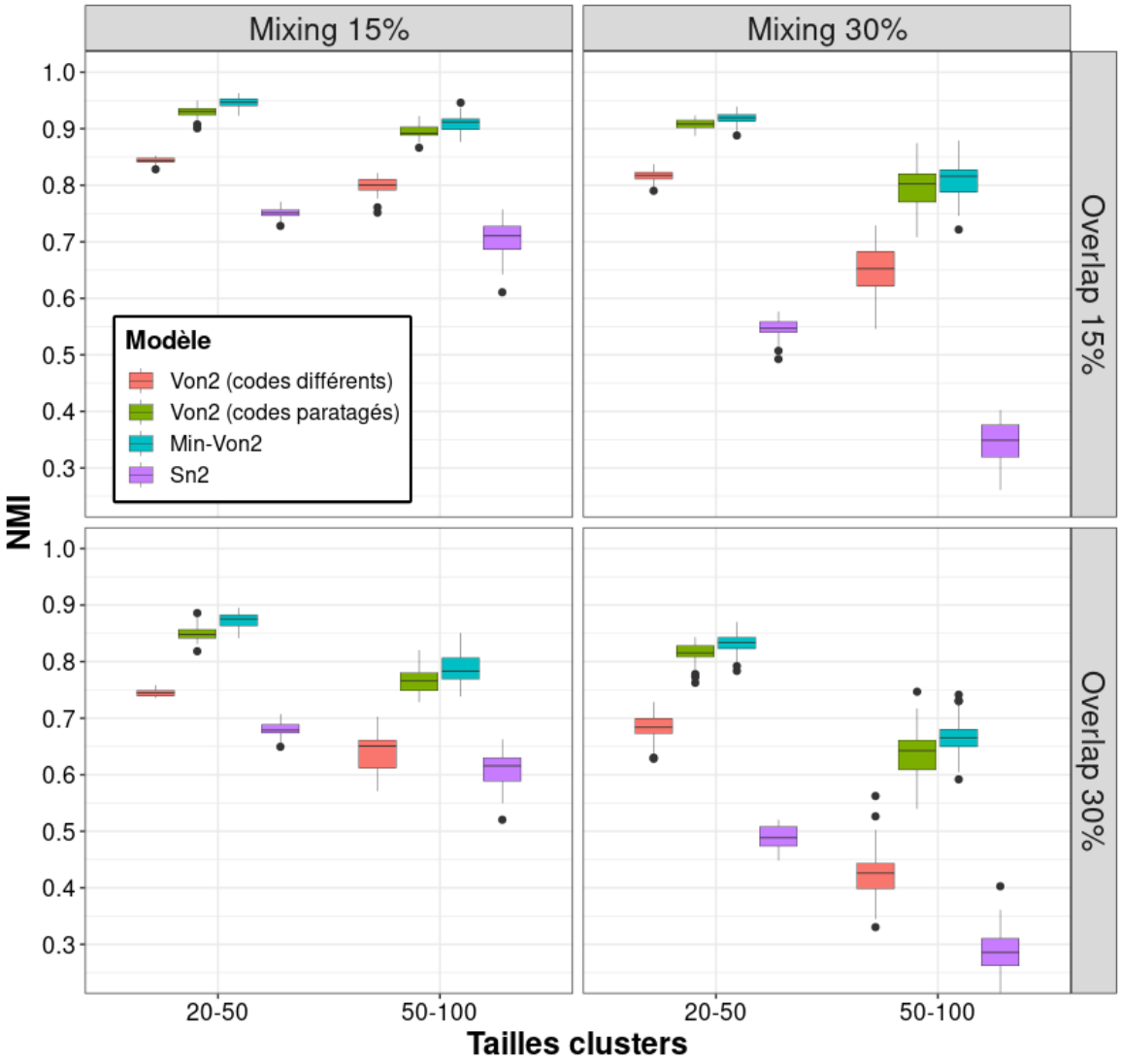


FIGURE 5.3 – NMI entre le clustering *Infomap* trouvé pour chaque cas de test et le clustering de la vérité terrain tel que généré par le benchmark LFR. Une valeur de 1 indique une identification parfaite des clusters réels [57]. Chaque diagramme en boîte correspond à la distribution sur 50 tests.

Le nombre de nœuds obtenus selon les paramètres utilisés pour les réseaux VON$_2$ et MIN-VON$_2$ est indiqué dans le tableau 5.1. Étant donné que le nombre maximum de communautés par état est de 2, chaque lieu se chevauchant devrait avoir 3 représentations dans MIN-VON$_2$ *e.g.* en prenant un chevauchement de 15% et 1000 états (quatre premières lignes dans tableau 5.1), MIN-VON$_2$ devrait contenir 1300 nœuds. Les divergences avec les résultats rapportés proviennent de l'algorithme LFR qui peut avoir besoin d'assouplir certaines contraintes *i.e.* le nombre de nœuds qui se chevauchent peut varier. Remarquons que le nombre de nœuds dans VON$_2$ est toutefois plus faible lorsque les valeurs de mélange sont plus élevées. En effet, les voisins inter-clusters d'un emplacement qui se chevauche ne génèrent pas de nœuds-mémoires supplémentaires.

La différence entre les clusterings trouvés et la « vérité-terrain » est présentée dans la figure 5.3. Les distributions des valeurs NMI suggèrent que la correction du premier effet est importante car elle améliore considérablement la détection du véritable clustering dans toutes les situations. L'amélioration obtenue en utilisant le MIN-VON$_2$ n'est pas aussi importante. Nous pouvons toutefois constater un écart notable lorsque la différence de nombre de représentations entre VON$_2$ et MIN-VON$_2$ est la plus grande (ratio de mélange de 15% et 30% de nœuds qui se chevauchent). Dans cette situation, il semble que trop de clusters soient identifiés avec VON$_2$ et SN$_2$. Il apparaît également que les réseaux d'ordre variable sont toujours plus performants que les réseaux SN$_2$. Cela montre que la correction du premier effet suggérée par Rosvall *et al.* [70] n'est pas suffisante.

5.5.2 Données réelles

Même si la dernière étude de cas est révélatrice, la dynamique de flux utilisée est plutôt simple et le benchmark LFR peut ne pas refléter les systèmes réels dans lesquels les séquences observées se produisent. En outre, les réseaux construits précédemment correspondent à des *scenarii* idéaux où les probabilités de transition ne sont pas estimées et les extensions pertinentes ne sont pas extraites d'un ensemble de séquences. Il est en effet possible que le choix du modèle n'ait pas d'impact sur les *clusterings* des états trouvés pour des ensembles de données réelles. Il est donc important de comparer les résultats sur des ensembles de données réelles. Dans ce contexte, nous utilisons le modèle AGG-VON$_2$ (au lieu de MIN-VON$_2$) construit à partir de D_{KL}-VON$_2(1)$.

Nous montrons tout d'abord que notre modèle agrégé est plus parcimonieux et conserve un bon pouvoir représentatif. En outre, cette précision n'est pas atteinte lorsque l'on construit des D_{KL}-VON$_2(\lambda)$ réduits avec différentes valeurs de seuil. Nous montrons ensuite que les dépendances séquentielles peuvent conduire à des *clusterings Infomap* présentant des variations notables. En particulier, lorsque l'on utilise le modèle AGG-VON$_2$, les *clusterings* tendent à contenir moins de *clusters* et de chevauchements entre ces *clusters*.

5.5.2.1 Paramètres expérimentaux

Dans cette partie, nous utilisons quatre des jeux de données présentés dans la section 2.5 : *AIR*, *PORTS*, *WIKI* et *MSNBC*. Pour rappel, nous avons supprimé les répétitions consécutives d'états dans chaque séquence.

Pour chaque réseau construit, nous analyserons le nombre de nœuds (nombre total de représentations) et le nombre de représentations par état $N_{\mathcal{V}}$ (définition 2.20). Pour évaluer la capacité des réseaux à modéliser la dynamique du flux, nous utilisons à nouveau la procédure décrite dans la section 3.4.1 en utilisant 90% des séquences pour construire les

différents modèles et 10% pour tester la précision Acc (équation 3.18). À la différence des résultats discutés dans la section 3.4.1, nous comparons ici des modèles limités à l'ordre 2. Il est donc probable que les modèles soient moins performants.

Puisque que le modèle Sn_2 a également été proposé pour le *clustering* dans la littérature, nous l'incluons aussi dans cette analyse. Dans ce cas, on considère que l'estimation $p(\sigma|s_0)$ d'ordre 1 est obtenue en prenant une représentation de s_0 tirée aléatoirement. Le calcul de la précision est donc équivalent au calcul de la précision avec le modèle Fon_2. Pour le modèle agrégé Agg-Von_2, on aura $p(\sigma|s_0s_1) = p(\sigma|X)$ où X est la représentation fusionnée à laquelle appartient s_0s_1.

Afin de comparer des modèles de taille similaire et en sachant que le modèle Agg-Von est plus parcimonieux que $\text{Von}_2(\alpha)$ pour $\alpha = 1$, nous allons, à l'instar de la procédure décrite dans la section 3, utiliser une valeur α plus élevée pour construire des réseaux plus parcimonieux, puisque cette variable a pour effet de diminuer le nombre de nœuds-mémoire. Pour chaque jeu de données, nous trouvons le paramètre λ^* (section 3.2.2.2) tel que le nombre total de représentations dans D_{KL}-$\text{Von}_2(\lambda^*)$ soit aussi proche que possible de celui de Agg-Von_2. Nous comparons donc quatre réseaux, pour chaque ensemble de données : D_{KL}-$\text{Von}_2(1)$, D_{KL}-$\text{Von}_2(\lambda^*)$, Agg-Von_2 et Sn_2. Pour chaque ensemble de données et chaque modèle, le test a été effectué 50 fois et nous indiquons la valeur moyenne de précision *Acc* et l'écart-type de cette mesure.

Infomap étant un algorithme non déterministe, nous l'avons appliqué 50 fois sur chaque réseau (construit à partir de l'ensemble des séquences $\mathcal{S}$) et conservons le *clustering* chevauchant $\mathcal{C}$ associé à la plus petite longueur de code. Nous indiquons le nombre de groupes $N_{\mathcal{C}} : \mathcal{A} \to \mathbb{N}^+$ pour chaque état.
Nous calculons également la diminution de la longueur du code ΔL par rapport à l'absence de *clustering*. Une valeur ΔL proche de 0 suggère que le *clustering* n'est pas un bon résumé de la dynamique du flux. Cette mesure (ainsi que les valeurs absolues de L) ne peut pas être utilisée pour comparer directement les modèles de réseau entre eux, car la longueur du code sera mécaniquement plus élevée avec le nombre de nœuds, qui peut varier. Nous reportons cependant le NMI [57] entre les *clusterings* trouvés.

5.5.2.2 Discussion des résultats

Nous commençons par répondre aux deux premières questions posées à la fin de la section 5.4 : le réseau agrégé est-il plus parcimonieux et, dans l'affirmative, représente-t-il suffisamment bien la dynamique des flux ?

Les statistiques relatives à la taille des réseaux et leur précision sont présentées dans le

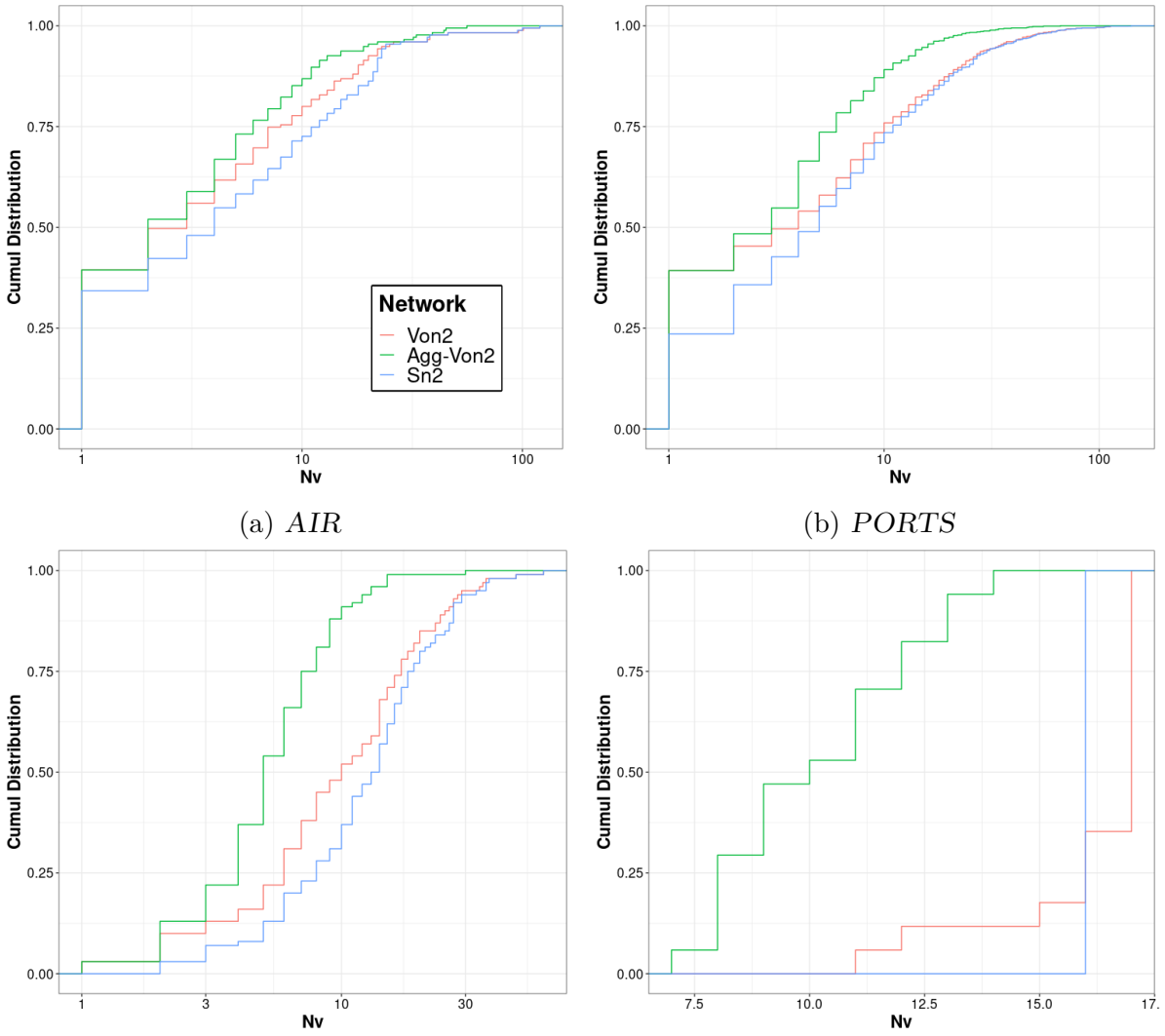
TABLE 5.2 – Comparaison de la précision des modèles

Données	**Réseau**	**Tps Const.**	$\lvert\mathcal{V}\rvert$	**moy** $N_\mathcal{V}$	**max** $N_\mathcal{V}$	**Acc** $\pm 2sd$
AIR	D_{KL}-VON$_2$(1)	2.23s	1356	7.75	120	27.07% ± 0.12
	D_{KL}-VON$_2$(2.8)	2.62s	959	5.48	90	25.41% ± 0.14
	SN$_2$	1.30s	1598	9.13	120	27.44% ± 0.12
	AGG-VON$_2$	7.65s	951	5.43	56	26.79% ± 0.13
PORTS	D_{KL}-VON$_2$(1)	0.76s	8005	8.81	136	32.02% ± 1.24
	D_{KL}-VON$_2$(3.05)	0.80s	4391	4.83	116	27.36% ± 1.18
	SN$_2$	0.80s	8755	9.63	141	32.50% ± 1.24
	AGG-VON$_2$	4.27s	4398	4.84	66	30.92% ± 1.25
WIKI	D_{KL}-VON$_2$(1)	0.2s	1267	12.67	60	23.08% ± 0.73
	D_{KL}-VON$_2$(2.4)	0.16s	596	5.96	39	22.07% ± 0.70
	SN$_2$	0.1s	1488	14.88	60	23.36% ± 0.72
	AGG-VON$_2$	0.66s	607	6.07	30	22.87% ± 0.75
MSNBC	D_{KL}-VON$_2$(1)	2.69s	273	16.06	17	21.72% ± 0.14
	D_{KL}-VON$_2$(2.5)	2.93s	173	10.18	17	20.29% ± 0.14
	SN$_2$	1.24s	272	16.00	16	21.75% ± 0.14
	AGG-VON$_2$	2.79s	173	10.18	14	21.36% ± 0.15

tableau 5.2. Les distributions cumulées des valeurs de $N_\mathcal{V}$ sont données dans la figure 5.4. Nous pouvons tout d'abord remarquer que le nombre total de nœuds dans les réseaux agrégés D_{KL}-VON$_2$ est nettement inférieur à celui des réseaux D_{KL}-VON$_2$ ou SN$_2$. À noter que la baisse des valeurs de $N_\mathcal{V}$ a surtout un impact sur les états fortement représentés. Pour le jeu de données *MSNBC* (figure 5.4d), les valeurs de $N_\mathcal{V}$ sont distribuées de manière plus uniforme pour les réseaux D_{KL}-VON$_2$ et SN$_2$.

Nous constatons par ailleurs que les modèles D_{KL}-VON$_2$ et SN$_2$ ont des scores de précision proches. Il y a en effet relativement peu de contextes jugés non-pertinents par D_{KL}-VON à l'ordre 2. Des valeurs de précision inférieures sont observées avec le modèle AGG-VON$_2$. La perte semble toutefois négligeable par rapport aux différences dans les valeurs de N_V. En outre, les résultats en matière de précision sont nettement moins bons pour D_{KL}-VON$_2(\lambda^*)$, même si la différence n'est pas nécessairement importante, comme c'est le cas pour *AIR* ou *MSNBC*. Nous pouvons conclure que notre stratégie d'agrégation est efficace à cet égard et que nos deux premières hypothèses sont vérifiées sur ces jeux de données.

Nous examinons maintenant les regroupements obtenus avec l'algorithme *Infomap* en utilisant les différents modèles. Les statistiques pertinentes sont rapportées dans le tableau 5.3, les distributions cumulées pour $N_\mathcal{C}$ (nombre de *clusters* par état) sont données dans la figure 5.5. La colonne $\lvert\mathcal{C}\rvert_{>1}$ rapporte le nombre de *clusters* de taille supérieure à 1 permettant d'éliminer les cas de noeuds d'ordre 1 laissés isolés (voir l'exemple 5.1). La similarité entre les regroupements (selon la NMI) peut être trouvée dans le tableau 5.4.



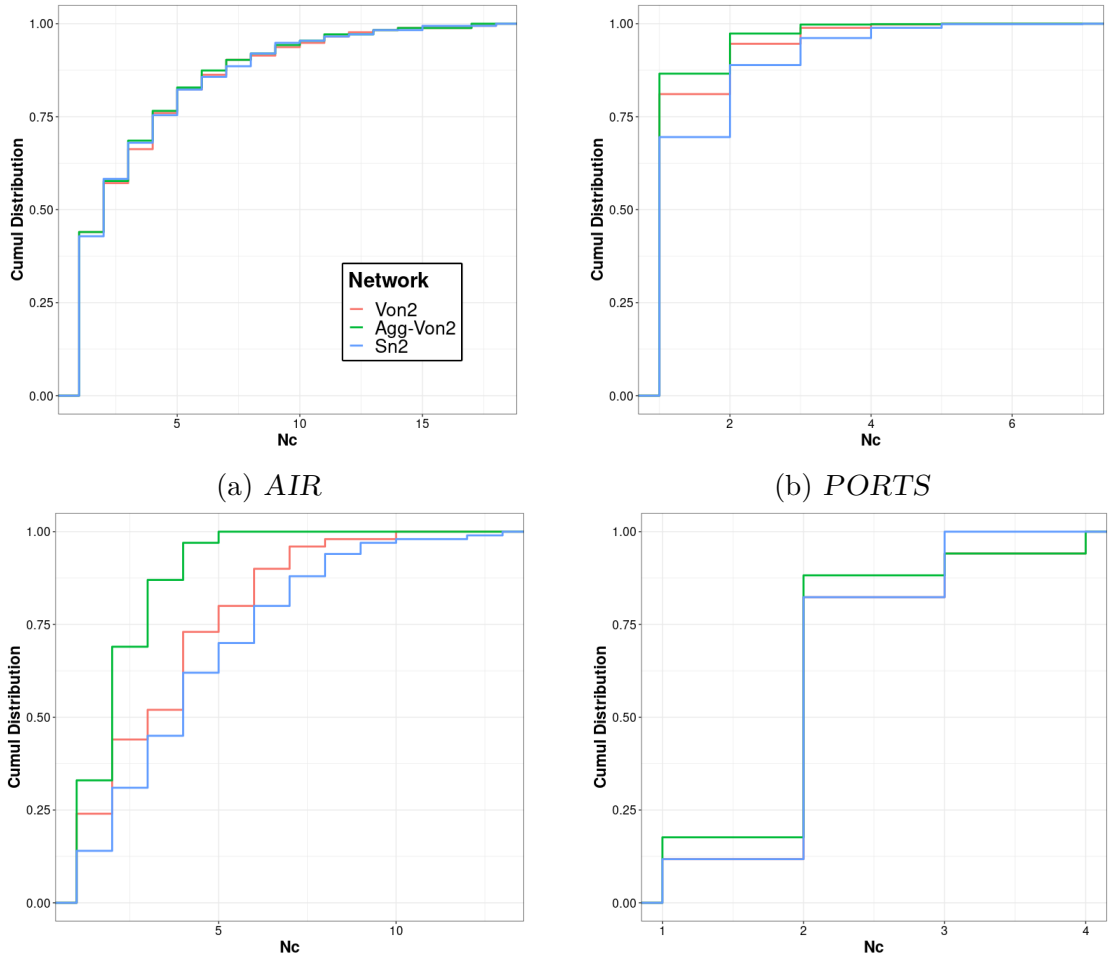
(a) *AIR* (b) *PORTS*

(c) *WIKI* (d) *MSNBC*

FIGURE 5.4 – Distribution cumulative du nombre de représentations $N_{\mathcal{V}}$ pour les états. Pour chaque panneau, $y(x)$ donne le ratio d'états ayant au plus x représentations.

La similarité entre les *clusterings* calculés est fluctuante selon le jeu de données. Ils sont notamment similaires dans le cas de *AIR* mais bien plus faibles pour *PORTS* ou *WIKI*. Par ailleurs, dans le cas de *MSNBC*, le *clustering* obtenu avec AGG-VON$_2$ est plus éloigné des deux autres. Les $N_{\mathcal{C}}$ sont généralement plus faibles pour la plupart des jeux de données avec le modèle agrégé. Cela signifie qu'il y a moins de chevauchements entre les *clusters*. Cela se produit même lorsque le nombre de *clusters* est similaire. Par ailleurs, il est possible de noter que des *clusters* de taille 1 ne se rencontrent que dans *PORTS* et *MSNBC*.

Même si ces ensembles de données du monde réel n'incluent pas de vérité-terrain, nous pouvons conclure que l'utilisation du modèle AGG-VON$_2$ peut conduire à des *clusterings* significativement différents. En particulier, ils contiennent moins de groupes et moins de



(a) *AIR* (b) *PORTS*

(c) *WIKI* (d) *MSNBC*

FIGURE 5.5 – Distribution cumulative du nombre de groupes N_C pour les états. Pour chaque panneau, $y(x)$ donne le ratio d'états trouvés dans au plus x *clusters*. L'étendue de l'axe y peut varier d'un panneau à l'autre.

chevauchements. Cela suggère que l'impact du nombre de représentations sur les algorithmes basés sur la marche aléatoire n'est pas marginal.

Nous reportons dans les tableaux 5.2 et 5.3 les temps de calcul pour la construction de chaque réseau et le temps moyen pris par l'algorithme *Infomap* pour produire un *clustering*. Sans surprise, il est possible de constater que l'agrégation des représentations ralentit significativement la construction du réseau. Le temps supplémentaire requis dépend évidemment du nombre de nœuds dans le réseau D_{KL}-VON$_2$. Toutefois, le nombre réduit de nœuds de AGG-VON$_2$ permet d'accélérer le partitionnement avec *Infomap*. Cet algorithme de partitionnement n'étant pas déterministe, il doit être exécuté plusieurs fois (50 ici). Seul le temps moyen d'une seule exécution est indiqué ici dans le tableau 5.3.

TABLE 5.3 – Comparaison des résultats de *clustering*.

Données	**Réseau**	**Tps Clust(s)**	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{C}\rvert_{>1}$	**moy** $N_\mathcal{C}$	**max** $N_\mathcal{C}$	ΔL
AIR	VON$_2$(1)	0.43	94	82	3.69	18	48.24%
	AGG-VON$_2$	0.20	83	68	3.22	17	42.75%
	SN$_2$	0.40	88	81	3.28	18	48.77%
PORTS	VON$_2$(1)	1.86	8	8	1.25	5	55.81%
	AGG-VON$_2$	1.09	8	8	1.17	5	50.46%
	SN$_2$	1.92	44	44	1.47	7	57.07%
WIKI	VON$_2$(1)	0.21	40	40	3.45	10	44.99%
	AGG-VON$_2$	0.14	25	25	2.14	5	36.54%
	SN$_2$	0.24	69	69	4.24	13	48.36%
MSNBC	VON$_2$(1)	0.17	5	4	2.18	4	47.67%
	AGG-VON$_2$	0.08	4	3	2	4	41.36%
	SN$_2$	0.12	4	4	2.06	3	48.01%

TABLE 5.4 – NMI entre les *clusterings* trouvés sur D_{KL}-VON$_2$, AGG-VON$_2$ et SN$_2$.

Données	**Réseau**	AGG-VON$_2$	SN$_2$
AIR	VON$_2$(1)	0.882	0.924
	AGG-VON$_2$	-	0.868
PORTS	VON$_2$(1)	0.552	0.330
	AGG-VON$_2$	-	0.273
WIKI	VON$_2$(1)	0.345	0.514
	AGG-VON$_2$	-	0.228
MSNBC	VON$_2$(1)	0.318	0.914
	AGG-VON$_2$	-	0.635

5.6 Discussion

Dans ce chapitre, nous nous sommes intéressés au clustering sur les HON, plus précisément à un algorithme basé sur la marche aléatoire, *Infomap*. Malgré le fait qu'il soit possible d'appliquer des algorithmes directement aux modèles HON, nous montrons que le faire sans adaptation peut également mener à un biais, dû à la multiplicité des représentations. Pour réduire ce biais, nous proposons un modèle AGG-VON$_2$, plus parcimonieux, qui respecte la dynamique de flux. Grâce à des expérimentations sur des *benchmarks* synthétiques, nous montrons que ces modèles minimaux permettent d'obtenir des résultats de clustering très différents des modèles VON classiques.

Au vu des résultats validant notre théorie, il nous semble important de pouvoir généraliser notre approche à des ordres supérieurs à 2. Il serait ainsi possible de concevoir une extension de notre modèle agrégé. Cette perspective est discutée dans la section 5.6.1. Une autre perspective est de développer un algorithme tenant mieux compte de la variété de représentations des états. Nous discutons une adaptation possible de l'algorithme *Walktrap* [63] dans la section 5.6.2.

5.6.1 Extension du modèle agrégé à tout ordre

Les résultats présentés dans cet article sont valables dans le cas limité où l'ordre maximal de représentations est de 2. Le modèle AGG-VON$_2$ montre que nous pouvons obtenir des résultats de clustering très différents en utilisant un modèle de réseau plus parcimonieux qui capture presque aussi bien la dynamique des flux. Cependant, nous avons vu dans le Chapitre 3 que des dépendances à des ordres supérieurs à 2 existent pour les différents jeux de données considérés ici.

En effet, les réseaux VON généraux contiennent beaucoup plus de représentations de chaque état que les réseaux VON$_2$. Nous pouvons nous attendre à ce que les effets sur le clustering basé sur *Infomap* soient non seulement toujours présents, mais renforcés.

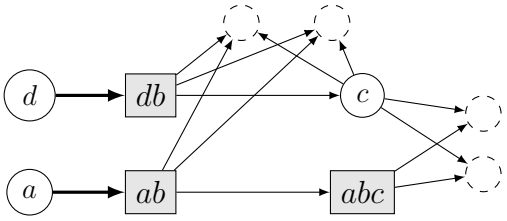


FIGURE 5.6 – Exemple de cas ambigu lorsque l'on tente de construire un réseau agrégé avec des représentations d'ordre supérieur à 2. Nous supposons que les distributions P_{db} et P_{ab} sont similaires. Dans ce contexte, la fusion des noeuds correspondants brisera la relation $ab \to abc$ qui est nécessaire pour encoder ce contexte d'ordre 3.

Nous pourrions dans un premier temps utiliser l'algorithme 3 pour fusionner les nœuds dans un VON$_k$ avec $k > 2$. Faire cela peut introduire des ambiguïtés dans les dépendances séquentielles encodées. En effet, les relations entre les nœuds-mémoires sont un moyen de contraindre un marcheur aléatoire à des destinations spécifiques, comme l'exprime la propriété 2.1. Dans l'exemple donné dans la figure 5.6, deux représentations d'ordre 2 sont similaires en termes d'emplacement visité suivant, mais elles ne doivent pas être fusionnées si on veut prendre en compte une dépendance séquentielle plus grande. Cette situation ne se produit jamais avec un ordre maximal de 2, comme montré dans la section 5.4.1.

Ce problème limite la perspective de généraliser notre méthode. Cette transformation est utile pour VON$_2$ car elle traite les deuxième et troisième effets discutés dans la section 5.4. En effet, pour le réseau de la figure 5.2a, la probabilité pour un surfeur aléatoire de se téléporter vers l'un des $\{av, bc, cv\}$ ou $\{dv, ev\}$ est respectivement de $\frac{3}{10}$ ou $\frac{2}{10}$, alors qu'elle est de $\frac{1}{8}$ pour les deux nœuds agrégés dans la figure 5.2b. Cela atténue le deuxième effet, car la probabilité d'utiliser le code pour une représentation de v est réduite. De plus, la fusion de ces nœuds dans la figure 5.2b correspond à la contrainte de toujours les considérer comme faisant partie des mêmes clusters. Cela annule le troisième effet.

Une solution possible pour le clustering *Infomap* des réseaux Von (quel que soit l'ordre) consiste à continuer d'utiliser les valeurs de *PageRank* calculées par *a priori*, mais en utilisant désormais des taux de téléportation non uniformes correspondant aux représentations fusionnées. Par exemple, la probabilité de se téléporter vers l'un des réseaux $\{av, bc, cv\}$ serait de $\frac{1}{3}\frac{1}{8}$. En outre, les groupes de représentations doivent être déplacés ensemble d'un cluster à l'autre pendant la recherche de la meilleure partition. Cette dernière contrainte nécessiterait des modifications importantes de l'algorithme *Infomap*. Il est à noter que, dans cette étude, nous n'avons modifié que l'entrée de l'algorithme et utilisé les paramètres déjà disponibles.

5.6.2 Adaptation d'autres algorithmes : *Walktrap*

L'autre piste que nous pouvons envisager ne consiste pas à adapter le modèle mais les algorithmes comme cela a été fait pour la mesure PageRank dans le chapitre précédent. En effet, les outils d'exploration de Hon devraient tenir compte de la variabilité de ces modèles. Cela représente un défi pour les chercheurs travaillant dans ce domaine.

Une piste possible est de concevoir des algorithmes tenant compte des dépendances séquentielles, mais travaillant au maximum dans l'espace des états. Pour tenir compte de ces dépendances, les approches reposant sur les marches aléatoires semblent appropriées.

Comme autre algorithme de clustering reposant sur la marche aléatoire, nous pourrions citer comme candidat, l'algorithme *Walktrap* [63], qui est un algorithme agglomératif basé sur le calcul d'une distance entre nœuds. Celle-ci mesure la différence entre l'ensemble des nœuds atteignables à l'aide de courtes marches aléatoires (entre 3 et 5 pas) et n'utilise donc pas de mécanisme de téléportation. Il sera donc intéressant de l'adapter aux réseaux Von, car il permet de tenir compte des principales dépendances séquentielles qui, comme déjà vu, ne dépassent généralement pas l'ordre 4. Par ailleurs, pour rester dans l'espace des états, une idée serait de généraliser cette distance pour évaluer la différence entre l'ensemble des *états* (et non leurs représentations) atteignables après de courtes marches aléatoires.
Walktrap, en tant qu'algorithme agglomératif, fusionne successivement les nœuds puis clusters les plus proches. La meilleur partition est sélectionnée en cherchant la coupe de la hiérarchie obtenue maximisant la modularité. Il serait possible, dans les cas des réseaux d'ordre supérieur, de développer une variation de la *Map Equation* permettant d'annuler le premier et deuxième effets identifiés dans la section 5.4.

Pour tester la validité de la méthode finale, il faudra ainsi vérifier que :

1. l'algorithme fournit de bons résultats : le clustering obtenu est proche de l'attendu pour des benchmarks comme celui proposé dans la section 5.5.1 ;
2. l'algorithme permet d'éliminer le biais des représentations : les clusterings obtenus avec des modèles plus ou moins parcimonieux, mais encodant les mêmes dépendances (comme ici $\text{VON}_2(1)$ et MIN-VON_2) fournissent, sont très proches.

Comme noté dans la section 5.2.1, il nous faudrait comparer les résultats obtenus avec des algorithmes de clusterings chevauchants (tel que *Fuzzy Infomap*) appliqués sur le réseau FON_1.

Chapitre 6

Conclusion

Ce travail a été divisé en deux problématiques principales : une première sur la représentation des données séquentielles par des réseaux et une deuxième qui questionne l'application d'algorithmes de fouille à ces réseaux.

Dans le chapitre 3, nous nous sommes intéressés à la recherche d'un modèle optimal, qui permet d'obtenir un bon compromis entre la précision du modèle et sa taille. Pour cela, nous avons proposé un modèle à ordre variable MC-Von alternatif au principal modèle la littérature, D_{KL}-Von. Ce modèle est aussi performant, voire plus, en restant plus parcimonieux. Cependant, pour construire le modèle il est nécessaire de définir le paramètre α, qui quantifie l'«improbabilité minimale» d'une extension. De plus, le modèle est très long à calculer contrairement à D_{KL}-Von.

Dans les deux autres chapitres 4 et 5, nous avons remis en cause l'application directe d'algorithmes de fouille sur les réseaux Hon, en se focalisant sur la centralité (*PageRank*) et le clustering (*Infomap*). Dans le cas de *PageRank*, nous proposons un changement qui permet de limiter le biais lié à la multiplicité des représentations. Nous obtenons des résultats différents, cependant, cela ne se traduit pas par un bousculement majeur des classements des états. Pour *Infomap*, nous proposons un modèle agrégé de Von plus parcimonieux. Pour ce réseau, les résultats de clustering sont différents entre des clustering obtenus avec les modèles Von classiques.

Dans les conclusions des différents chapitres, nous avons discuté les résultats et évoqué les perspectives de recherches futures dans le cadre fixé dans l'introduction de ce manuscrit. Le but est ici de présenter différentes perspectives et questionnements sur le cadre dans lequel cette recherche a été effectuée. Nous questionnons en particulier la trame d'analyse utilisée (et reprise de la littérature) à savoir : Données brutes → Séquences d'états → Réseau d'ordre supérieur → Algorithme de fouille de graphe → Projection sur les états.

Les deux prochaines sections questionnent l'intérêt de construire un réseau d'ordre supérieur indépendamment de l'information que l'expert cherche à obtenir sur les états dans le cas de la fouille (section 6.1) ou de l'apprentissage supervisé (section 6.2). Dans

la section 6.3, nous envisageons des pistes possibles pour détecter des dépendances séquentielles à partir de données temporelles sans passer par une première transformation en séquences.

6.1 Limites de l'approche proposée pour la fouille des Hon

Dans ce manuscrit, nous avons abordé la mesure de la centralité des états dans les HON ainsi que le partitionnement de ces réseaux résultant en un clustering chevauchant des états. Toutefois, chaque analyse est centrée sur une mesure ou un algorithme particulier. Nous avons conclu, que ce soit pour la centralité ou le clustering, que les procédures devaient être adaptées à l'algorithme utilisé. Nous remettons ainsi en cause l'indépendance entre les étapes « Algorithme de fouille de graphe » et « Projection sur les états ». Ce besoin d'adaptation a toutefois un coût : il existe de nombreuses mesures de centralité ou de méthodes de partitionnement avec différents avantages et inconvénients. La perspective de pouvoir les utiliser indépendamment du modèle avec relativement peu d'adaptations était donc une perspective intéressante de la littérature.

Dans un premier temps, on pourrait plus généralement se demander si le calcul des mesures ou des clusterings sur un HON « minimalement parcimonieux » n'est pas une approximation suffisante. C'est l'approche qui est proposée dans le chapitre 5 avec le modèle MIN-VON. Dans ce cadre, il faudrait pouvoir étendre le modèle agrégé au-delà de l'ordre 2. Par ailleurs, nous avons développé ce concept en se basant sur le modèle D_{KL}-VON mais d'autres sont également possibles, comme le modèle MC-VON défini dans le chapitre 3.

Dans un second temps, nous pourrions essayer de faire l'inventaire des mesures qui peuvent être adaptées. En effet, les HON correspondent à des graphes mais cela ne veut pas dire que toutes les mesures appliquées sur eux aient un « sens ». Des mesures structurelles se basant sur le degré entrant/sortant, les plus-courts-chemins ou la densité du graphe (ou d'une de ses parties) ont *a priori* une pertinence limitée. En effet, les réseaux présentés ici sont stochastiques ; le poids pour un arc correspond à la probabilité d'aller d'une représentation à l'autre. Une mesure telle que le degré indique à quel point chaque transition est observée au moins une fois. Les probabilités de transition sont donc totalement ignorées, un arc de poids proche de 0 comptant autant qu'un avec un poids proche de 1. Cette observation vaut déjà pour FON_1, qui n'est pas un réseau d'ordre supérieur. Dans le cadre des réseaux FON_k ou VON, le nombre d'arcs et la densité peuvent varier de manière importante selon les paramètres de construction. Toutefois, l'intérêt de ces

mesures est tout aussi limité que pour le réseau FON_1.

Ce constat est toutefois à nuancer si on tient compte des résultats de Scholtes *et al.* [76]. Ces derniers ont en effet montré que des généralisations de mesures de centralité basées sur des plus-courts-chemins sur les réseaux d'ordre supérieur étaient une bonne approximation de leurs équivalents dans des réseaux temporels. Cette conclusion peut être due au fait que même une probabilité de transition faible est toujours importante face à une transition qui n'a jamais été observée. Or les transitions jamais observées sont très nombreuses dans les systèmes étudiés, les réseaux générés sont creux (*sparse*).

Remarquons enfin que bien que cette thèse a permis de mettre en lumière ce besoin d'adaptation des mesures dans le cas de HON, nous nous situons toutefois dans la même approche que celle proposée dans la littérature à savoir que la construction du modèle HON est déconnecté des techniques de fouille. Or si le résultat final (*i.e.* la centralité des états) dépend du modèle, il faudrait que la modélisation soit partie intégrante de l'analyse. Par exemple, on pourrait étudier le PageRank appliqué à MC-VON selon la valeur du paramètre α permettant de juger la pertinence des nœuds-mémoires. On peut dans ce cadre conjecturer qu'un réseau « maximalement » parcimonieux produira le meilleur compromis entre les différents résultats d'analyse.

6.2 Applications aux Réseaux neuronaux en Graphes

Cette thèse a principalement traité de méthodes de fouille de graphe que l'on peut classer en tant que méthodes d'*apprentissage non-supervisé*. Il est toutefois possible que le problème de l'expert corresponde à des tâches d'apprentissage supervisé : par exemple, compléter un étiquetage partiel des états ou classifier des HON construits à partir d'observations sur différentes périodes afin d'identifier des périodes anormales [71].

Pour ces différentes tâches, des outils tels que les « réseaux neuronaux en graphes » [73] (*Graph Neural Networks* ou GNN) peuvent être utilisés et ont donné lieu à une littérature abondante ces dernières années [35]. Une grande partie des méthodes proposées reposent sur le « passage de messages » (*message passing*). Dans ce cadre, chaque nœud du graphe est associé à une « représentation », ici correspondant à un vecteur de caractéristiques (*features*). À chaque itération (correspondant aux couches d'un réseau neuronal), chaque nœud agrège les messages (vecteurs de caractéristiques) dans son voisinage et met à jour sa représentation.

Dans le contexte de graphes construits à partir de séquences, on peut s'attendre à ce que la prise en compte des dépendances séquentielles ait une influence sur la qualité de

l'apprentissage. Cette perspective a été étudiée par Jin *et al.* [43] et Krieg *et al.* [49]. Les seconds proposent notamment une adaptation du mécanisme de *message passing* pour le modèle D_{KL}-VON(λ) menant à des meilleurs résultats que sur le modèle FON$_1$. Par ailleurs, ils mettent en lumière l'influence du paramètre λ (Déf. 3.2) sur les résultats de tâches de classifications.

La dernière observation effectuée dans la section précédente est d'autant plus centrale ici : si le choix du modèle influence les performances d'algorithme CLE (d') apprentissage supervisé alors nous devons tenir compte de cette étape dans la tâche d'apprentissage (c'est également la conclusion de Krieg *et al.* [49]). On pourra dans ce cadre définir le « meilleur » modèle de HON comme étant celui donnant les meilleurs résultats. Ce problème d'inférence de graphe rejoint des problématiques plus générales d'« apprentissage de structures de graphes » (*Graph Structure Learning*) [18]. Une première perspective pour nous est d'évaluer la performance de modèle comme MC-VON dans le cadre de tâches d'apprentissage supervisé mais également d'évaluer l'impact des modèles agrégés proposés dans le chapitre 5.

6.3 Détection de dépendances séquentielles à partir d'interactions temporelles

Nous discutons ici de l'identification de dépendances séquentielles à partir de données temporelles dans la lignée des travaux de Scholtes [76]. Les séquences discrètes d'événements sont adaptées aux cas d'études où les observations correspondent à des trajectoires ou des historiques de l'activité d'un agent (utilisateur d'un site web, navire, *etc.*). Cette représentation n'est pas possible dans certains réseaux d'interactions, on peut notamment penser aux échanges électroniques ou aux interactions entre personnes dans un cadre professionnel. À la place d'un agent, on pourrait imaginer que c'est dans ce cadre une information qui change d'état avant d'être rediffusée voir traitée ou dupliquée puis rediffusée.

Comme discuté dans la section 2.5.6, Scholtes *et al.* [76, 75] cherchent à ramener le problème à celui d'un ensemble de séquences à partir d'un réseau temporel afin de détecter l'existence de dépendances séquentielles. Nous pensons qu'une autre approche est possible en se basant sur les « modèles relationnels d'événements » (*Relation Event Model* ou REM) [74, 8]. Ces modèles statistiques permettent de modéliser l'apparition d'interactions entre des entités (*i.e.* générer un réseau temporel). La fréquence d'interactions entre (u, v) peut dépendre de variables endogènes (*e.g.* nombre d'interactions passées entre u et v) ou exogènes (*e.g.* position spatiale des entités).

Les REM n'incluent pas explicitement de dépendances séquentielles. Dans notre cas, il pourrait être intéressant de formaliser un tel modèle dans lequel le HON serait une variable latente. La fréquence d'interactions serait ici équivalente aux probabilités de transitions. Notre tâche consisterait à trouver le HON expliquant au mieux les interactions temporelles. Nous abandonnerions dans ce cadre la première transformation « Données → Séquences » de la chaîne d'analyse en travaillant directement sur les données temporelles. Cette perspective nous pousse à croire que les applications des travaux proposés dans cette thèse dépassent les cas d'études où seules des séquences discrètes sont disponibles pour l'expert.

BIBLIOGRAPHIE

[1] Sameer AGARWAL, Kristin BRANSON et Serge BELONGIE, « Higher order learning with graphs », in : *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, Pittsburgh, Pennsylvania, USA : Association for Computing Machinery, 2006, p. 17-24, ISBN : 1595933832, DOI : 10.1145/1143844.1143847.

[2] Rakesh AGRAWAL et Ramakrishnan SRIKANT, « Mining sequential patterns », in : *Proceedings of the Eleventh International Conference on Data Engineering*, mars 1995, p. 3-14, DOI : 10.1109/ICDE.1995.380415.

[3] Phipps ARABIE et al., « Overlapping Clustering : A New Method for Product Positioning », in : *Journal of Marketing Research* 18.*3* (1981), p. 310-317, DOI : 10.1177/002224378101800305, eprint : https://doi.org/10.1177/002224378101800305.

[4] Said BAADEL, Fadi THABTAH et Joan LU, « Overlapping clustering : A review », in : *2016 SAI Computing Conference (SAI)*, 2016 SAI Computing Conference (SAI), 2016, p. 233-237, DOI : 10.1109/SAI.2016.7555988.

[5] Federico BATTISTON et al., « Networks beyond pairwise interactions : Structure and dynamics », in : *Physics Reports* 874.*0* (2020), p. 1-92, ISSN : 0370-1573, DOI : 10.1016/j.physrep.2020.05.004.

[6] Ron BEGLEITER, Ran EL-YANIV et Golan YONA, « On Prediction Using Variable Order Markov Models », en, in : *Journal of Artificial Intelligence Research* 22 (déc. 2004), p. 385-421, ISSN : 1076-9757, DOI : 10.1613/jair.1491.

[7] Austin R. BENSON, David F. GLEICH et Jure LESKOVEC, « Higher-order organization of complex networks », in : *Science* 353.*6295* (juill. 2016), p. 163-166, ISSN : 1095-9203, DOI : 10.1126/science.aad9029.

[8] Federica BIANCHI et al., « Relational Event Modeling », in : *Annual Review of Statistics and Its Application* 11.*Volume 11, 2024* (2024), p. 297-319, ISSN : 2326-831X, DOI : 10.1146/annurev-statistics-040722-060248.

[9] Francis BLOCH, Matthew O. JACKSON et Pietro TEBALDI, *Centrality Measures in Networks*, 2021, arXiv : 1608.05845 [physics.soc-ph].

[10] Vincent D BLONDEL et al., « Fast unfolding of communities in large networks », in : *Journal of Statistical Mechanics : Theory and Experiment* 2008.*10* (oct. 2008), P10008, ISSN : 1742-5468, DOI : 10.1088/1742-5468/2008/10/p10008.

[11] Jose Borges et Mark Levene, « Evaluating Variable-Length Markov Chain Models for Analysis of User Web Navigation Sessions », in : *IEEE Transactions on Knowledge and Data Engineering* 19.*4* (avr. 2007), Conference Name : IEEE Transactions on Knowledge and Data Engineering, p. 441-452, ISSN : 1558-2191, DOI : 10.1109/TKDE.2007.1012.

[12] José Borges et Mark Levene, « Generating Dynamic Higher-Order Markov Models in Web Usage Mining », en, in : *Knowledge Discovery in Databases : PKDD 2005*, sous la dir. de David Hutchison et al., t. 3721, Series Title : Lecture Notes in Computer Science, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 34-45, ISBN : 978-3-540-29244-9 978-3-540-31665-7, DOI : 10.1007/11564126_9.

[13] Ulrik Brandes et al., *Maximizing Modularity is hard*, 2006, arXiv : physics/0608255 [physics.data-an], URL : https://arxiv.org/abs/physics/0608255.

[14] Sergey Brin et Lawrence Page, « The anatomy of a large-scale hypertextual Web search engine », en, in : *Computer Networks and ISDN Systems*, Proceedings of the Seventh International World Wide Web Conference 30.*1* (avr. 1998), p. 107-117, ISSN : 0169-7552, DOI : 10.1016/S0169-7552(98)00110-X.

[15] Mattia Bunel et al., « Geovisualizing the sail-to-steam transition through vessel movement data », in : *Advances in Shipping Data Analysis and Modeling*, Routledge, 2017, p. 189-205.

[16] Brenno Caetano Troca Cabella et al., « A numerical study of the Kullback-Leibler distance in functional magnetic resonance imaging », in : *Brazilian Journal of Physics* 38 (2008), p. 20-25, DOI : 10.1590/S0103-97332008000100005.

[17] Samuele Capobianco et al., « Deep Learning Methods for Vessel Trajectory Prediction Based on Recurrent Neural Networks », in : *IEEE Transactions on Aerospace and Electronic Systems* 57.*6* (2021), p. 4329-4346, DOI : 10.1109/TAES.2021.3096873.

[18] Yu Chen et Lingfei Wu, « Graph Neural Networks : Graph Structure Learning », in : *Graph Neural Networks : Foundations, Frontiers, and Applications*, sous la dir. de Lingfei Wu et al., Singapore : Springer Nature Singapore, 2022, p. 297-321, ISBN : 978-981-16-6054-2, DOI : 10.1007/978-981-16-6054-2_14.

[19] Flavio Chierichetti et al., « Are web users really Markovian ? », en, in : *Proceedings of the 21st international conference on World Wide Web - WWW '12*, Lyon, France : ACM Press, 2012, p. 609-618, ISBN : 978-1-4503-1229-5, DOI : 10.1145/2187836.2187919.

[20] Célestin COQUIDÉ, José LAGES et Dima L. SHEPELYANSKY, « World influence and interactions of universities from Wikipedia networks », en, in : *The European Physical Journal B* 92.*1* (jan. 2019), p. 3, ISSN : 1434-6036, DOI : 10.1140/epjb/e2018-90532-7.

[21] Célestin COQUIDÉ, Julie QUEIROS et François QUEYROI, « PageRank computation for Higher-Order Networks », en, in : (sept. 2022), DOI : 10.48550/arXiv.2109.03065.

[22] Célestin COQUIDÉ et al., « Influence of petroleum and gas trade on EU economies from the reduced Google matrix analysis of UN COMTRADE data », en, in : *The European Physical Journal B* 92.*8* (août 2019), p. 171, ISSN : 1434-6036, DOI : 10.1140/epjb/e2019-100132-6.

[23] Benjamin CORNWELL, « Network Analysis of Sequence Structures », in : *Sequence Analysis and Related Approaches : Innovative Methods and Applications*, sous la dir. de Gilbert RITSCHARD et Matthias STUDER, Cham : Springer International Publishing, 2018, p. 103-120, ISBN : 978-3-319-95420-2, DOI : 10.1007/978-3-319-95420-2_7.

[24] Vinh Loc DAO, Cécile BOTHOREL et Philippe LENCA, « Community structure : A comparative evaluation of community detection methods », in : *Network Science* 8.*1* (jan. 2020), p. 1-41, DOI : 10.1017/nws.2019.59.

[25] Dong DING, Axel GANDY et Georg HAHN, « A simple method for implementing Monte Carlo tests », in : *Computational Statistics* 35 (2020), p. 1373-1392, DOI : 10.1007/s00180-019-00927-6.

[26] Tina ELIASSI-RAD et al., « Higher-Order Graph Models : From Theoretical Foundations to Machine Learning (Dagstuhl Seminar 21352) », in : *Dagstuhl Reports* 11.*7* (2021), sous la dir. de Tina ELIASSI-RAD et al., p. 139-178, ISSN : 2192-5283, DOI : 10.4230/DagRep.11.7.139, URL : https://drops.dagstuhl.de/entities/document/10.4230/DagRep.11.7.139.

[27] Santo FORTUNATO, « Community detection in graphs », in : *Physics Reports* 486.*3–5* (fév. 2010), p. 75-174, ISSN : 0370-1573, DOI : 10.1016/j.physrep.2009.11.002.

[28] Philippe FOURNIER-VIGER et al., « A survey of sequential pattern mining », in : *Data Science and Pattern Recognition* 1.*1* (2017), p. 54.

[29] Klaus M. FRAHM, Katia JAFFRÈS-RUNSER et Dima L. SHEPELYANSKY, « Wikipedia mining of hidden links between political leaders », en, in : *The European Physical Journal B* 89.*12* (déc. 2016), p. 269, ISSN : 1434-6036, DOI : 10.1140/epjb/e2016-70526-3.

[30] Axel Gandy, « Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk », in : *Journal of the American Statistical Association* 104.*488* (2009), p. 1504-1511, doi : 10.1198/jasa.2009.tm08368.

[31] M. Girvan et M. E. J. Newman, « Community structure in social and biological networks », in : *Proceedings of the National Academy of Sciences* 99.*12* (2002), p. 7821-7826, doi : 10.1073/pnas.122653799, eprint : https://www.pnas.org/doi/pdf/10.1073/pnas.122653799.

[32] Vince Grolmusz, « A note on the PageRank of undirected graphs », in : *Information Processing Letters* 115.*6* (2015), p. 633-634, issn : 0020-0190, doi : 10.1016/j.ipl.2015.02.015.

[33] Furkan Gursoy et Bertan Badur, « Extracting the signed backbone of intrinsically dense weighted networks », in : *Journal of Complex Networks* 9.*5* (oct. 2021), cnab019, issn : 2051-1329, doi : 10.1093/comnet/cnab019, eprint : https://academic.oup.com/comnet/article-pdf/9/5/cnab019/40545302/cnab019.pdf, url : https://doi.org/10.1093/comnet/cnab019.

[34] Aric A. Hagberg, Daniel A. Schult et Pieter J. Swart, « Exploring Network Structure, Dynamics, and Function using NetworkX », in : *Proceedings of the 7th Python in Science Conference*, sous la dir. de Gaël Varoquaux, Travis Vaught et Jarrod Millman, Pasadena, CA USA, 2008, p. 11-15, url : https://www.osti.gov/biblio/960616.

[35] William L. Hamilton, « The Graph Neural Network Model », in : *Graph Representation Learning*, Cham : Springer International Publishing, 2020, p. 51-70, isbn : 978-3-031-01588-5, doi : 10.1007/978-3-031-01588-5_5, url : https://doi.org/10.1007/978-3-031-01588-5_5.

[36] Ivan Herman, Guy Melancon et M. Scott Marshall, « Graph visualization and navigation in information visualization : A survey », in : *IEEE Transactions on Visualization and Computer Graphics* 6.*1* (2000), p. 24-43, doi : 10.1109/2945.841119.

[37] Paul W. Holland, Kathryn Blackmond Laskey et Samuel Leinhardt, « Stochastic blockmodels : First steps », in : *Social Networks* 5.*2* (1983), p. 109-137, issn : 0378-8733, doi : https://doi.org/10.1016/0378-8733(83)90021-7.

[38] Petter Holme et Jari Saramäki, « Temporal networks », in : *Physics Reports* 519.*3* (2012), Temporal Networks, p. 97-125, issn : 0370-1573, doi : https://doi.org/10.1016/j.physrep.2012.03.001.

[39] Zhen Huang et al., « Group-aware graph neural networks for sequential recommendation », in : *Information Sciences* 670 (2024), p. 120623, issn : 0020-0255, doi : https://doi.org/10.1016/j.ins.2024.120623.

[40] Lorenzo Isella et al., « What's in a crowd ? Analysis of face-to-face behavioral networks », in : *Journal of Theoretical Biology* 271.*1* (2011), p. 166-180, ISSN : 0022-5193, DOI : https://doi.org/10.1016/j.jtbi.2010.11.033.

[41] Väinö Jääskinen et al., « Sparse Markov Chains for Sequence Data », in : *Scandinavian Journal of Statistics* 41.*3* (2014), p. 639-655, DOI : https://doi.org/10.1111/sjos.12053.

[42] Riko Jacob et al., « Algorithms for Centrality Indices », in : *Network Analysis : Methodological Foundations*, sous la dir. d'Ulrik Brandes et Thomas Erlebach, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 62-82, ISBN : 978-3-540-31955-9, DOI : 10.1007/978-3-540-31955-9_4, URL : https://doi.org/10.1007/978-3-540-31955-9_4.

[43] Di Jin et al., « Graph Neural Network for Higher-Order Dependency Networks », in : *Proceedings of the ACM Web Conference 2022*, WWW '22, Virtual Event, Lyon, France : Association for Computing Machinery, 2022, p. 1622-1630, ISBN : 9781450390965, DOI : 10.1145/3485447.3512161.

[44] Honey Jindal et Neetu Sardana, « Web navigation prediction using Markov-based models : an experimental study », en, in : *International Journal of Web Engineering and Technology* (jan. 2017), Publisher : Inderscience Publishers (IEL), DOI : 10.1504/IJWET.2016.081766.

[45] Wang-Cheng Kang et Julian McAuley, « Self-Attentive Sequential Recommendation », in : *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, p. 197-206, DOI : 10.1109/ICDM.2018.00035.

[46] Anne-Marie Kermarrec et al., « Second order centrality : Distributed assessment of nodes criticity in complex networks », in : *Computer Communications* 34.*5* (2011), Special Issue : Complex Networks, p. 619-628, ISSN : 0140-3664, DOI : https://doi.org/10.1016/j.comcom.2010.06.007.

[47] Sina Khanmohammadi, Naiier Adibeig et Samaneh Shanehbandy, « An improved overlapping k-means clustering method for medical applications », in : *Expert Systems with Applications* 67 (jan. 2017), p. 12-18, ISSN : 0957-4174, DOI : 10.1016/j.eswa.2016.09.025.

[48] Dirk Koschützki et al., « Centrality Indices », in : *Network Analysis : Methodological Foundations*, sous la dir. d'Ulrik Brandes et Thomas Erlebach, Berlin, Heidelberg : Springer Berlin Heidelberg, 2005, p. 16-61, ISBN : 978-3-540-31955-9, DOI : 10.1007/978-3-540-31955-9_3.

[49] Steven Krieg et al., « Deep Ensembles for Graphs with Higher-order Dependencies », in : *The Eleventh International Conference on Learning Representations*, 2023, DOI : 10.48550/arXiv.2205.13988.

[50] Steven J. Krieg, Peter M. Kogge et Nitesh V. Chawla, « GrowHON : A Scalable Algorithm for Growing Higher-order Networks of Sequences », en, in : *Complex Networks & Their Applications IX*, sous la dir. de Rosa M. Benito et al., t. 944, Series Title : Studies in Computational Intelligence, Cham : Springer International Publishing, 2021, p. 485-496, isbn : 978-3-030-65350-7 978-3-030-65351-4, doi : 10.1007/978-3-030-65351-4_39.

[51] Dirk P Kroese, Thomas Taimre et Zdravko I Botev, *Handbook of monte carlo methods*, John Wiley & Sons, 2013, doi : 10.1002/9781118014967.

[52] Renaud Lambiotte, Martin Rosvall et Ingo Scholtes, *Understanding Complex Systems : From Networks to Optimal Higher-Order Models*, en, rapp. tech. arXiv :1806.05977, arXiv :1806.05977 [cond-mat, physics :physics] type : article, arXiv, juin 2018.

[53] Andrea Lancichinetti, Santo Fortunato et János Kertész, « Detecting the overlapping and hierarchical community structure in complex networks », in : *New Journal of Physics* 11.*3* (mars 2009), p. 033015, doi : 10.1088/1367-2630/11/3/033015.

[54] Andrea Lancichinetti, Santo Fortunato et Filippo Radicchi, « Benchmark graphs for testing community detection algorithms », in : *Physical Review E* 78.*4* (oct. 2008), doi : 10.1103/PhysRevE.78.046110.

[55] Andrea Landherr, Bettina Friedl et Julia Heidemann, « A Critical Review of Centrality Measures in Social Networks », in : *Business & Information Systems Engineering* 2 (2010), p. 371-385, url : https://api.semanticscholar.org/CorpusID:15720889.

[56] Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge, UK : Cambridge University Press, 2008, isbn : 978-0-521-86571-5, doi : 10.1017/CBO9780511809071.

[57] Aaron F. McDaid, Derek Greene et Neil Hurley, *Normalized Mutual Information to evaluate overlapping community finding algorithms*, 2013, doi : 10.48550/arXiv.1110.2515.

[58] Radosław Michalski, Sebastian Palus et Przemysław Kazienko, « Matching Organizational Structure and Social Network Extracted from Email Communication », in : *Business Information Systems*, sous la dir. de Witold Abramowicz, Berlin, Heidelberg : Springer Berlin Heidelberg, 2011, p. 197-206, isbn : 978-3-642-21863-7, doi : 10.1007/978-3-642-21863-7_17.

[59] Luis Moreira-Matias, Michel Ferreira et João Mendes-Moreira, *Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015*, UCI Machine Learning Repository, 2015, doi : 10.24432/C55W25.

[60] John F. Padgett et Christopher K. Ansell, « Robust Action and the Rise of the Medici, 1400-1434 », in : *American Journal of Sociology* 98.*6* (1993), p. 1259-1319, doi : 10.1086/230190.

[61] Airel Pérez-Suárez et al., « An algorithm based on density and compactness for dynamic overlapping clustering », in : *Pattern Recognition* 46.*11* (2013), p. 3040-3055, issn : 0031-3203, doi : https://doi.org/10.1016/j.patcog.2013.03.022.

[62] Nicola Perra et Santo Fortunato, « Spectral centrality measures in complex networks », in : *Phys. Rev. E* 78 (3 sept. 2008), p. 036107, doi : 10.1103/PhysRevE.78.036107, url : https://link.aps.org/doi/10.1103/PhysRevE.78.036107.

[63] Pascal Pons et Matthieu Latapy, *Computing communities in large networks using random walks (long version)*, 2005, arXiv : physics/0512106 [physics.soc-ph].

[64] Julie Queiros, Célestin Coquidé et François Queyroi, « Toward random walk-based clustering of variable-order networks », in : *Network Science* 10.*4* (2022), p. 381-399, doi : 10.1017/nws.2022.36.

[65] Julie Queiros, François Queyroi et Simon Artus, *HONyx*, version 0.1.1, 10 mai 2024, url : https://pypi.org/project/honyx/.

[66] Julie Queiros, François Queyroi et Samuel Maistre, « Sampling based sequential dependencies discovery in Higher-Order Network Models », in : *Proceedings of the French Regional Conference on Complex Systems*, 2024, p. 125-1936, doi : 10.5281/zenodo.11267401.

[67] Jean Rivière et al., « Les divisions socioprofessionnelles en mouvement d'une métropole attractive. Le cas de l'aire urbaine de Nantes (1975-2015) », in : *Cybergeo : European Journal of Geography* (2021), doi : 10.4000/cybergeo.36572.

[68] Nicolas Robette, *Explorer et décrire les parcours de vie : les typologies de trajectoires*, Collections Du CEPED, 2011, p. 86.

[69] M. Rosvall, D. Axelsson et C. T. Bergstrom, « The map equation », en, in : *The European Physical Journal Special Topics* 178.*1* (nov. 2009), p. 13-23, issn : 1951-6401, doi : 10.1140/epjst/e2010-01179-1.

[70] Martin Rosvall et al., « Memory in network flows and its effects on spreading dynamics and community detection », en, in : *Nature Communications* 5.*1* (août 2014), Number : 1 Publisher : Nature Publishing Group, p. 4630, issn : 2041-1723, doi : 10.1038/ncomms5630.

[71] Mandana Saebi et al., « Efficient modeling of higher-order dependencies in networks : from algorithm to application for anomaly detection », en, in : *EPJ Data Science* 9.*1* (déc. 2020), Number : 1 Publisher : SpringerOpen, p. 1-22, issn : 2193-1127, doi : 10.1140/epjds/s13688-020-00233-y.

[72] Mandana Saebi et al., « Network analysis of ballast-mediated species transfer reveals important introduction and dispersal patterns in the Arctic », en, in : *Scientific Reports* 10.*1* (déc. 2020), p. 19558, ISSN : 2045-2322, DOI : 10.1038/s41598-020-76602-4.

[73] Franco Scarselli et al., « The Graph Neural Network Model », in : *IEEE Transactions on Neural Networks* 20.*1* (2009), p. 61-80, DOI : 10.1109/TNN.2008.2005605.

[74] David R. Schaefer et Christopher Steven Marcum, « Modeling Network Dynamics », in : *The Oxford Handbook of Social Networks*, Oxford University Press, jan. 2021, ISBN : 9780190251765, DOI : 10.1093/oxfordhb/9780190251765.013.19, URL : https://doi.org/10.1093/oxfordhb/9780190251765.013.19.

[75] Ingo Scholtes, « When is a Network a Network ? Multi-Order Graphical Model Selection in Pathways and Temporal Networks », en, in : *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, arXiv :1702.05499 [cond-mat, physics :physics], août 2017, p. 1037-1046, DOI : 10.1145/3097983.3098145.

[76] Ingo Scholtes, Nicolas Wider et Antonios Garas, « Higher-order aggregate networks in the analysis of temporal networks : path structures and centralities », in : *The European Physical Journal B* 89 (2016), p. 1-15, DOI : https://doi.org/10.1140/epjb/e2016-60663-0.

[77] Claude Elwood Shannon, « A Mathematical Theory of Communication », in : *The Bell system technical journal* 27.*3* (1948), p. 379-423.

[78] Philipp Singer et al., « Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order », in : *PLOS ONE* 9.*7* (juill. 2014), p. 1-21, DOI : 10.1371/journal.pone.0102070, URL : 10.1371/journal.pone.0102070.

[79] Matthias Studer et Gilbert Ritschard, « What Matters in Differences Between Life Trajectories : A Comparative Review of Sequence Dissimilarity Measures », in : *Journal of the Royal Statistical Society Series A : Statistics in Society* 179.*2* (juill. 2015), p. 481-511, ISSN : 0964-1998, DOI : 10.1111/rssa.12125.

[80] Alcides Viamontes Esquivel et Martin Rosvall, « Compression of Flow Can Reveal Overlapping-Module Organization in Networks », in : *Phys. Rev. X* 1 (2 déc. 2011), p. 021025, DOI : 10.1103/PhysRevX.1.021025.

[81] Markus Wallinger et al., « Edge-path bundling : A less ambiguous edge bundling approach », in : *IEEE Transactions on Visualization and Computer Graphics* 28.*1* (2021), p. 313-323.

[82] S. S. Wilks, « The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses », in : *The Annals of Mathematical Statistics* 9.*1* (1938), p. 60-62, DOI : 10.1214/aoms/1177732360, URL : https://doi.org/10.1214/aoms/1177732360.

[83] Jierui Xie, Stephen Kelley et Boleslaw K. Szymanski, « Overlapping community detection in networks : The state-of-the-art and comparative study », in : *ACM Computing Surveys* 45.*4* (août 2013), p. 1-35, ISSN : 1557-7341, DOI : 10.1145/2501654.2501657.

[84] Jian Xu, Thanuka L. Wickramarathne et Nitesh V. Chawla, « Representing higher-order dependencies in networks », en, in : *Science Advances* 2.*5* (mai 2016), Publisher : American Association for the Advancement of Science Section : Research Article, e1600028, ISSN : 2375-2548, DOI : 10.1126/sciadv.1600028.

[85] Dengyong Zhou, Jiayuan Huang et Bernhard Schölkopf, « Learning with Hypergraphs : Clustering, Classification, and Embedding », in : *Advances in Neural Information Processing Systems*, sous la dir. de B. Schölkopf, J. Platt et T. Hoffman, t. 19, MIT Press, 2006.

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

Nantes Université

Titre : Analyse de réseaux d'ordre supérieur construits à partir de séquences historio-géographiques

Mot clés : Analyse de Réseaux, Séquences discrètes, Ordre supérieur, Centralité, Clustering

Résumé : L'analyse de réseaux permet d'extraire de l'information sur un système à partir des relations existantes entre ses entités. Ces relations peuvent correspondre à des flux tirés de séquences d'états (*i.e.* des trajectoires d'agents entre ces entités). La représentation de ces données sous la forme de graphes suppose généralement que ces trajectoires respectent la propriété de Markov ; seul l'état courant est suffisant pour déterminer l'état futur d'un agent. De nombreux travaux ont remis en cause cette hypothèse et proposé d'autres modèles permettant de passer outre : les réseaux d'ordre supérieur ou HON. Dans les HON, un état peut être représenté par différents nœuds-mémoires encodant les états précédemment visités avant d'aboutir à l'état courant. Une première problématique traitée dans cette thèse est la construction des HON, qui est un problème de modélisation statistique : on cherche un bon compromis entre taille et qualité du modèle. Nous proposons un modèle de réseaux d'ordre variable plus parcimonieux que les modèles existants. Une deuxième problématique est l'analyse des HON. Un avantage souvent mis en avant est que les algorithmes classiques de graphes peuvent être utilisés avec peu de modifications. Nous montrons, dans le cas de mesures de centralité ou de clustering de graphes, que ce n'est pas le cas. Nous affirmons qu'il faut au contraire adapter les algorithmes aux modèles.

Title: Analysis of Higher-Order Networks built from historio-geographic sequences

Keywords: Network Analysis, Discrete Sequences, Higher-Order, Centrality, Clustering

Abstract: Network analysis extracts information about a system from the relationships between its actors. These relationships can correspond to flows drawn from sequences of states (i.e. agent trajectories between these entities). Representing these data as graphs generally assumes that these trajectories respect the Markov property; only the current state is sufficient to determine an agent's future state. Numerous studies have challenged this assumption and proposed other models that allow us to overcome it: higher-order networks or HON. In it, a state can be represented by different memory nodes encoding the states previously visited before arriving at the current state. A first problem addressed in this thesis is the construction of HON, which is a statistical modeling problem: finding a good compromise between model size and quality. We propose here a model of variable-order networks that is more parsimonious than existing models. A second problem is the analysis of HON. An advantage often put forward is that classical graph algorithms can be used with few modifications. We show, in the case of centrality measures or graph clustering, that this is not the case and argue that algorithms should be adapted to models.