Hyper-bag-graphs and their applications: Modeling, Analyzing and Visualizing Complex Networks of Co-occurrences

Ouvrard, Xavier Eric

UNIVERSITÉ de GENÈVE
Département d'Informatique
CERN
Département IPT

FACULTÉ DES SCIENCES
Professeur Stéphane Marchand-Maillet

Docteur Jean-Marie Le Goff

Hyper-bag-graphs and their applications

Modeling, Analyzing and Visualizing Complex Networks of Co-occurrences

THÈSE

présentée à la Faculté des Sciences de l'Université de Genève
pour obtenir le grade de Docteur ès Sciences, mention Informatique

par

Xavier OUVRARD

de

Chevry, France

Thèse N° 5449

GENÈVE
Centre d'impression de l'Université de Genève
2020

UNIVERSITÉ
DE GENÈVE
FACULTÉ DES SCIENCES

DOCTORAT ÈS SCIENCES, MENTION INFORMATIQUE

Thèse de Monsieur Xavier Eric OUVRARD

intitulée :

«Hyper-bag-graphs and their Applications

**Modeling, Analyzing and Visualizing
Complex Networks of Co-occurrences»**

La Faculté des sciences, sur le préavis de Monsieur S. MARCHAND-MAILLET, professeur associé et directeur de thèse (Département d'informatique), Monsieur J.-M. LE GOFF, docteur et codirecteur de thèse (Physics Department, CERN - European Organization for Nuclear Research, Geneva, Switzerland), Monsieur G. FALQUET, professeur associé (Faculté d'Economie et de Management, Information Science Institute), Monsieur A. TELCS, professeur (Department of Computational Sciences, Wigner Research Centre for Physics, Budapest, Hungary) et Monsieur Z. BURDA, professeur (Department of Applied Informatics and Computational Physics, Faculty of Physics and Applied Computer Science, AGH University of Science and Technology, Kraków, Poland), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 28 avril 2020

Thèse - 5449 -

Le Doyen

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

Abstract

Obtaining insights in the tremendous amount of data in which the Big Data era has brought us, requires to develop specific tools, that are not only summaries of data through classical charts and tables, but that allow full navigation and browsing of a dataset. The proper modeling of databases can enable such navigation and we propose in this Thesis a methodology to achieve the browsing of an information space, through its different facets. To achieve the modeling of such an information space, co-occurrences of data instances are built referring to a common reference type. Historically, the co-occurrences were seen as pairwise relationships and developed as such. The move to hypergraphs enables the possibility to take into account the multi-adicity of the relationships, and to have a representation through the incident graph that simplifies deeply its 2-section.

Nonetheless, representing large hypergraphs calls for a coarsening of the information by having insights on important vertices and hyperedges. One classical way to achieve it is to use a diffusion process over the network. Achieving it using an incident matrix is feasible but brings us to a pitfall, as it brings us back to a pairwise relationship. Making proper diffusion requires a tensor approach. This is well known for uniform hypergraphs, where all the hyperedges have same cardinality, but still very challenging for general hypergraphs. After redefining the concept of adjacency in general hypergraphs, we propose a first e-adjacency tensor, that involves a Hypergraph Uniformisation Process and a Polynomial Homogenization Process. This is achieved by uniformisation of the original hypergraph by decomposing it into layers—each of them containing a uniform hypergraph—and filling each layer with additional special vertices and merging them together. This process requires to have as many additional vertices as the number of layers.

In order to reduce the number of special vertices, we need to have the possibility of repeating a vertex when filling, which is not possible with hyperedges as they are sets. We need multisets. It, therefore, requires a new mathematical structure, that we have introduced and called hyper-bag-graph—hb-graph for short—, which is a family of multisets of a given universe.

Co-occurrences can also have repetitions or individual weighting of their vertices inside a given co-occurrence and hb-graphs fit to handle it. Hence, we introduce a hb-graph framework for co-occurrence networks. We then work on diffusion on such structures, using, in a first step, a matrix approach. Aggregating the ranking of vertices and hb-edges of this diffusion on each of the facet of the information space is achieved by using a multi-diffusion scheme. Since different facets might have different focus of interest, we introduce a biased diffusion that enables a tuning on the point of emphasis on the feature we are interested in.

Finally, coming back to e-adjacency tensor, we propose three e-adjacency tensors of hb-graphs, that are based on different ways of filling the hb-edges. The m-uniformisation

that is achieved is evaluated and compared to the ones achieved by hb-edge splitting, concluding that any m-uniformisation process has an influence on the exchange-based diffusion that we propose. Hence, we conclude that diffusion using the tensor approach must be done in an informed manner to account for this diffusion change. We finally discuss different possible of achieving it and present a new Laplacian that can help to achieve it.

Résumé

À l'ère du Big Data, l'énorme quantité et la variété des données imposent, pour générer un aperçu signifiant, de développer des outils spécifiques qui ne se réduisent pas simplement à des graphiques ou à des tableaux de synthèse classiques ; ils doivent aussi permettre une navigation complète dans le jeu de données.

La modélisation appropriée de ces données va favoriser une telle navigation. Cette Thèse propose précisément un cadre adapté à la navigation dans tout espace d'infor- mation multi-facettes. Cette modélisation nécessite la construction de cooccurrences d'instances de données en se référant à un (ou des) type(s) commun(s) servant de référence. Pendant très longtemps, les cooccurrences, vues comme collaborations, ont été perçues comme des relations par paires : leur représentation était alors développée en tant que telles. Le passage aux hypergraphes pour modéliser les réseaux de cooccurrences laisse la possibilité de prendre en compte la multiplicité des relations. Cela permet également d'avoir une représentation via le graphe d'incidence de l'hypergraphe, ce qui simplifie profondément la visualisation obtenue par 2-section de l'hypergraphe.

Néanmoins, quand la taille des hypergraphes augmente, et afin de pouvoir continuer à les visualiser convenablement, il faut un tri de l'information pour révéler au mieux les sommets et hyper-arêtes importants du réseau. En ce domaine, une des méthodes classiques consiste à utiliser un processus de diffusion. Une première approche réside dans l'utilisation de la matrice d'incidence de l'hypergraphe. Mais cela comporte un possible piège, car, en raison de sa nature matricielle, la matrice d'incidence induit des relations par paires, soit explicitement soit implicitement. Aussi une autre approche de la diffusion est indispensable : il faut tenir compte des relations d'ordre supérieures induites par les hyper-arêtes et donc adopter une approche tensorielle. La diffusion tensorielle, via le Laplacien, est bien connue pour les hypergraphes uniformes, où toutes les hyper-arêtes ont la même cardinalité ; mais elle reste encore un vaste champ d'études pour les hypergraphes généraux. Après avoir redéfini le concept d'adjacence dans les hypergraphes généraux, nous proposons donc un premier tenseur d'e-adjacence, qui implique un processus d'uniformisation de l'hypergraphe et un processus d'homogénéisation polynomiale. Aussi, afin de représenter les relations d'e-adjacence d'un hypergraphe de manière tensorielle, on uniformise l'hypergraphe d'origine en le décomposant en couches d'hypergraphes uniformes, puis en remplissant chaque couche avec des sommets n'appartenant pas à l'hypergraphe d'origine et enfin en fusionnant ces couches. Ce processus nécessite à une unité près d'avoir autant de sommets que la taille maximale d'une hyper-arête.

En outre, pour réduire le nombre de sommets ajoutés, tout en gardant l'interprétabilité en terme d'uniformisation, il faut avoir la possibilité de répéter un sommet lors du remplissage. Or, comme cela n'est pas envisageable avec les hypergraphes du fait de leur définition ensembliste, cela requiert les multi-ensembles (ou multisets). En résulte la nécessité d'une nouvelle structure mathématique qui est introduite dans cette

Thèse et appelée hyper-bag-graphes (ou hb-graphes de manière abrégée) : les hb-graphes sont définis comme étant des familles de multi-ensembles d'un univers donné, considéré comme l'ensemble des sommets.

C'est à la lueur de cette nouvelle structure que l'on peut dès lors proposer une modélisation raffinée de l'espace d'information multi-facettes. Cela permet de tenir compte du fait que chaque cooccurrence peut contenir des répétitions ou nécessiter une pondération individuelle de ses sommets. Chaque facette est alors modélisée par un hb-graphe. En découle l'importance d'une étude de la diffusion sur les hb-graphes utilisant en première instance une approche matricielle. L'agrégation du classement des hb-arêtes obtenu lors de la diffusion sur chacune des facettes de l'espace d'information est alors réalisée à l'aide d'un schéma de multi-diffusion. Et, comme différentes facettes amènent à se focaliser sur différents centres d'intérêt, nous avons introduit une diffusion biaisée : elle permet d'ajuster l'accent mis sur la caractéristique qui nous intéresse ainsi que sur le type de valeurs considérées comme importantes.

Enfin, pour en revenir au tenseur d'e-adjacence, trois tenseurs d'e-adjacence de hb-graphes sont proposés, utilisant différentes manières de remplir les hb-arêtes. La m-uniformisation utilisée est évaluée par comparaison à celle obtenue par la division d'hb-arêtes qui avait été proposée dans un tenseur d'e-adjacence étudié antérieurement par d'autres auteurs. La conclusion de cette évaluation est la suivante : tout processus de m-uniformisation a une influence sur la diffusion par échanges ; mais seule celle obtenue par addition de sommets spéciaux permet d'avoir une perturbation compréhensible sur le classement des sommets. Par ailleurs, cette perturbation peut être évitable, mais au prix d'un calcul moins direct des degrés des sommets. On peut alors en déduire que la diffusion utilisant l'approche tensorielle doit être faite de manière informée de sorte à tenir compte de ce changement de diffusion. Reste ensuite à étudier les différentes stratégies possibles pour réaliser cette diffusion et finalement proposer un nouveau Laplacien tensoriel.

Acknowledgments

I am deeply grateful and thankful to Pr. Stéphane Marchand-Maillet for the direction of this Thesis and to Dr. Jean-Marie Le Goff for his co-direction: for their continuous supervision, discussions and advices, and their infinite patience during this thesis, the co-supervision having given some fruitful exchanges and refinements.

To the dissertation reviewers and Defence Jury members, Pr. Zdzislaw Burda (AGH University of Science and Technology, Kraków), Pr. András Telcs (Budapest University of Technology and Economics) and Pr. Gilles Falquet (University of Geneva), to have accept to review this work, and for evaluating this work,

I am also grateful:

To CERN, for having funded a doctoral position that has allowed me to be dedicated full time on my PhD, and, more particularly, to the IPT department and his department head, Mr. Thierry Lagrange.

To the members of the Collaboration Spotting project at CERN, Dimitrios Dardanis (University of Geneva / CERN), Richard Forster (CERN), and last, but not least André Rattinger (University of Gratz / CERN), for the discussions and exchange we had during all the thesis,

To the University of Geneva and to the Swiss Doctoral Program, CUSO, which has been a precious aid for training and exchanges during the doctoral studies,

To CUI, University of Besançon, which has allowed me to do remote studies to reach the Master level in Computer Science,

To Edoardo Provenzi (University of Bordeaux), for his advice on the presentation of tensors,

To Tullio Basaglia (CERN), for his precious advices on the application of the DataHb-Edron to publication searches,

To Linn Tvelde (CERN), which has helped in improving the English in some parts of this Thesis,

To all the ones, met during conferences and seminars, that have given me some previous advice or went to fruitful discussions,

Finally, a big thanks:

To my wife, Cécile, for her support in those late studies, for her full support on the everyday life and her precious advices in the moment of choices and doubts.

Contents

Nomenclature

Hypergraphs

$\mathcal{H} = (V, E)$	Hypergraph of vertex set V and hyperedge set E
$V = \{v_i : i \in [\![n]\!]\}$	Vertex set of a hypergraph $\mathcal{H}$
$E = (e_j)_{j \in [\![p]\!]}$	Hyperedge family of a hypergraph
$H(v)$	Star of a vertex $v \in V$
$\mathcal{H}^*$	Dual of a hypergraph $\mathcal{H}$
$\Gamma(v)$	Neighbourhood of a vertex $v \in V$
$\mathcal{H}_w = (V, E, w)$	Weighted hypergraph of vertex set V, edge family E and weight function w
$o_{\mathcal{H}}$	Order of a hypergraph $\mathcal{H}$
$r_{\mathcal{H}}$	Rank of a hypergraph $\mathcal{H}$
$s_{\mathcal{H}}$	Anti-rank of a hypergraph $\mathcal{H}$
$[\mathcal{H}]_2$	2-section of the hypergraph $\mathcal{H}$
$[\mathcal{H}]_{\mathcal{I}}$	Interection graph of the hypergraph $\mathcal{H}$
$\mathcal{H}_1 + \mathcal{H}_2$	Sum of two hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$
$\mathcal{H}_1 \oplus \mathcal{H}_2$	Direct sum of two hypergraphs $\mathcal{H}_1$ and $\mathcal{H}_2$
H	Incidence matrix of a hypergraph
A	Adjacency matrix of a hypergraph
A_w	Adjacency matrix of a weighted hypergraph
$l(\mathscr{P})$	Length of a path $\mathscr{P}$
$d(u, v)$	Distance between two vertices u and v

Hb-graphs

$\mathfrak{H} = (V, \mathfrak{E})$	Hb-graph $\mathfrak{H}$ of vertex set V and hb-edge family $\mathfrak{E}$
$V = \{v_i : i \in [\![n]\!]\}$	Vertex set of a hb-graph $\mathfrak{H}$

$\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![p]\!]}$	Hb-edge family of a hb-graph $\mathfrak{H}$, where $\mathfrak{e}_j$ are multisets of universe V
$m_{\mathfrak{e}_j}$ or m_j	Multiplicity function of the hb-edge $\mathfrak{e}_j$
$m_{\mathfrak{e}_j}(v_i)$ or m_{ij}	Multiplicity value of the vertex v_i of the hb-edge $\mathfrak{e}_j$
$O(\mathfrak{H})$	Order of a hb-graph $\mathfrak{H}$
$\underline{\mathfrak{H}}$	Support hypergraph of a hb-graph $\mathfrak{H}$
$r_m(\mathfrak{H})$	m-rank of a hb-graph $\mathfrak{H}$
$r(\mathfrak{H})$	Range of a hb-graph $\mathfrak{H}$
$\mathrm{cr}_m(\mathfrak{H})$	m-co-rank of a hb-graph $\mathfrak{H}$
$\#_m\mathfrak{H}$	Global m-cardinality of a hb-graph $\mathfrak{H}$
$\mathrm{cr}(\mathfrak{H})$	Co-range of a hb-graph $\mathfrak{H}$
$\mathfrak{h}(v)$	Hb-star of a vertex $v \in V$
$\deg_m(v)$	m-degree of a vertex $v \in V$
Δ_m	Maximal m-degree of a hb-graph $\mathfrak{H}$
$\mathfrak{H}_0$	Numbered-copy-hypergraph of the hb-graph $\mathfrak{H}$
$\tilde{\mathfrak{H}}$	Dual of the hb-graph $\mathfrak{H}$
$l(\mathscr{P})$	Length of a path $\mathscr{P}$
$d(u,v)$	Distance between two vertices u and v

General

$\#_m\mathfrak{A}_m$	m-cardinality of a multiset $\mathfrak{A}_m$
$[\![n]\!]$	Integers from 1 to n
$\mathfrak{A}_m^*$	Support of the multiset $\mathfrak{A}_m$
$\mathfrak{A}_m = (A, m)$	Multiset of universe A and multiplicity function m
$\mathfrak{M}(A)$	Set of all multisets of universe A
$a \otimes b$	Segre outerproduct of two vectors a and b

List of Figures

List of Tables

List of Algorithms

Introduction

Big Data with its five Vs implies a lot of innovative visual analytics tools to have deeper insights into the DataSet. As the Volume of information is increasing and at the same time the Variety is gaining in heterogeneity, new tools need to consider all the Variety of the information space, serving it with Velocity. The Veracity of information is a big challenge and requires different strategies: it requires not only dataset consolidation and data linkage, but also to gain insight into datasets in order to retrieve meaningful and genuine information.

This quest for Veracity is linked to the Value of the Data: Big Data calls for improved mining, aggregation, linkage of data,... Different approaches exist to visualize and summarize information in an information space. Datasets composed only of numerical data appear as the easiest case: plethoric treatments exist to enhance proper visualisation and summarization; links can often be retrieved by different methods either using classical statistic methods or more modern approaches such as neural nets and deep learning. When coming to non-numerical data, the summarization is often left to very basic approaches such as histograms and the data interactivity is often let behind. To find inherent links, machine learning methods can help, but they need a strong framework to enhance the structure of the dataset. In multimedia datasets, data instances are a blend of text, audio, image and/or video content all linked by some physical references. Variety induces new ways to address datasets.

There are different approaches to model databases, such as the relational database model introduced by Codd in [Cod70], based on sets and relations. A relational database schema is shown as a hypergraph with convenient properties in [Fag83]. Graph databases use a graph as schema and have been introduced in the 1990s as it is mentioned in [AG08] to overcome the limitations of traditional databases for capturing the inherent graph structure of data in various situations such as in hypertext or geographic database systems. The Knowledge Graph[1] introduced by Google in 2012 is a good example of the inherent graph structure of data.

In [Ran45], Ranganathan exposes the concept of facets in an information space—he introduced these concepts already in 1924. Ranganathan considers that a document can be described by five facets, with a rule of thumbs called PMEST: Personality, Matter, Energy, Space and Time. Personality reveals the focal subject of the document, Matter reflects a substance or a property of the subject, Energy reflects the operation or action done on the subject, Space is considering the geographical location and Time the chronological and temporal situation of the subject.

The concept of multi-faceted information space can be extended to any document and enlarge to different views of the information space within which the Dataset lives. Each

[1] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

facet of an information space can be modeled as a network of co-occurrences built on common references as we will show in Chapter 6.

Networks of co-occurrences can be modeled with different approaches: either co-occurrences are seen as pairwise relationships like in [New01a, New01b] or seen as proper multi-adic relationships as it is done in [TCR10], using hypergraphs that were introduced by Berge in [Ber73].

Hypergraphs extend graphs by considering a family of subsets of a vertex set. Hypergraphs naturally support multi-adicity where graphs support only 2-adicity, i.e. pairwise relationships. If the concept of incidence of vertices to edges in a graph clearly extends to hyperedges in hypergraphs, it does not hold for adjacency, for which the concept has to be refined.

Different hypergraph representations exist as mentioned already in [Mäk90], some with the subset standard using Venn or Euler diagrams—that are not scaling up properly—and some with the edge standard, consisting in transforming the hypergraph into a graph. The edge-standard comprises two main representations: the clique representation and the extra-node representation. The clique representation, corresponds to a graph view of the hypergraph 2-section, developing each hyperedge of the original hypergraph in its combination of pairwise relationships. The extra-node representation corresponds to the hypergraph incident graph—also called Levi graph—, each hyperedge being assorted with an extra-node and vertices of the hyperedge being linked to this extra-node.

We have shown in [OLGMM17b], that visualizing hypergraphs through their incident graph allows to diminish the cognitive load of the 2-section of the hypergraph—the 2-section being no more than the representation of a pairwise relationship. Moreover, the 2-section of a hypergraph looses the group information: two hypergraphs can have the same 2-section—which is in general not desirable. The incident graph is unique for a given hypergraph. Incident graph of hypergraphs are bipartite graphs and can be viewed as a developed version of the hypergraph: they are a lesser evil than the subset standard representation or the 2-section graph when the hypergraph becomes larger.

When scaling up large hypergraph representations, the hypergraph visualisation generative model requires at a point a coarsening of the hypergraph, that can be achieved, among other techniques, by enhancing diffusion over the structure.

Hypergraphs as graphs cannot be reduced to their incident graphs even if they are linked in a one-to-one manner; the set relationship induced by the hyperedge is powerful and regroups the information in a single object. The incidence relationships between vertices and hyperedges in hypergraphs is modeled by a rectangular matrix. This rectangular matrix when multiplied conveniently by its transpose represents the pairwise relationship between vertices: it is taken as the basis for an adjacency matrix for hypergraphs; but, this adjacency matrix reflects in fact the adjacency in the 2-section graph of the hypergraph. Also different hypergraphs can have the same adjacency matrix which is not desirable. Hence, a proper modeling of adjacency in a hypergraph calls for tensors to enable the capture of the multi-adicity.

In Appendix B, we present the necessary mathematical requirements for this Thesis. It includes all the necessary elements on hypergraphs, on multisets and, on tensors and hypermatrices.

Adjacency is properly defined for uniform hypergraphs since Cooper in [CD12]. But, when tackling general hypergraphs—i.e. hypergraphs with hyperedges of various cardinalities—, the concept of adjacency needs refinements in order to distinguish the fact that vertices live together in a hyperedge and the number of vertices that are simultaneously adjacent. Therefore, we already introduced different concepts of adjacency in [OLGMM17a, OLGMM18a]—reproduced in Appendix E.1—and explain why so far these concepts were not needed with uniform hypergraphs. We then have proposed a tensor that extends the adjacency tensor of J. Cooper to general hypergraphs. We have emphasized on the interpretability of the construction to reach the proposed tensor: it relies on a hypergraph uniformisation process, by building a uniform hypergraph inserting different additional vertices to hyperedges in sufficient quantity.

As this number of additional vertices depends on the number of layers of uniform hypergraphs contained in a general hypergraph, the idea is to simplify the process, by allowing a vertex to occur many times to complement all the hyperedges to the same cardinality. But hypergraphs, by nature, are families of sets and do not allow repetitions. This can only be made possible using multisets: they are in their simplest formalisation, unordered list of elements, or in their more refined version, weighted list of elements. Moreover, in some complex co-occurrence networks, naturally arises the need to have vertex duplication, or, even, an abusively named "hyperedge-based" weighting of vertices. To achieve these different goals, in Chapter 1, we introduce the mathematical structure of hyper-bag-graphs (hb-graphs for short), that we define as families of multisets of same universe, thus generalizing hypergraphs.

The next four chapters aim at studying the importance of vertices and hb-edges throughout the network, studying the consequences of the paradigm change induced by the introduction of the hb-graphs on varying diffusion schemes.

In Chapter 2, we propose a diffusion process that enables not only to rank the vertices but also the hb-edges. This diffusion process is experimented on different datasets, including textual data and image data, and uses intensively the incident matrix of the hb-graph.

The matrix approach enhances valuable information on the co-occurrence networks modeled by hb-graphs. Nonetheless, the e-adjacency can only be captured by hypermatrices.

In Chapter 3, using hb-graphs, we introduce three new e-adjacency tensors interpretable in term of m-uniformisation of hb-graphs—extension of the uniformisation of hypergraphs presented in Appendix E.1. A first choice of tensor is then made on its desirable properties for spectral analysis, and we use it for proposing a new e-adjacency tensor for general hypergraphs.

We then draw a conclusion in Chapter 4 on the construction of these tensors by considering the different m-uniformisation processes—either found in the literature or in our work—through the exchange-based diffusion prism and show that only vertex filling m-uniformisation processes lead to explainable rankings of the vertices, but, anyhow, perturb the diffusion process and, therefore, call for an informed diffusion process.

Ultimately, in chapter 5, we propose some hints to solve the diffusion on general hb-graphs, that takes into account the necessity of having either an informed diffusion or a new approach that takes into account the variety of multi-adic relationships induced by the hb-edges in general hb-graphs and the redundancy and/or weight they induce.

The following chapter—Chapter 6—pushes the thought further on co-occurrence networks for modeling an information space. Co-occurrences can be seen as multisets; hence hb-graphs can model co-occurrence networks, and refine the hypergraph framework approach used to model an information space as we have succinctly presented in [OLGMM18d] by a framework using a hb-graph approach.

We then propose, in Chapter 7, a multi-diffusion process in an information space using simultaneously the hb-graph framework presented in Chapter 6 and the exchange-based diffusion proposed in Chapter 2, in order to aggregate the rankings obtained for the hb-edge attached references of the different facet visualisation hb-graphs.

As the importance put on the different features can differ from one facet to the other, we give as a complement in Appendix D.1 some biased diffusion that generalize the one proposed already in Chapter 2: this allows to emphasize some values during the diffusion process, and refine the facet ranking depending on its specificity.

We finish by concluding on what we realized during this Thesis and give a list of possible research questions for future work.

Most of the chapters are based on preprints or publications that have been accepted or in submission during the three years of our PhD. We show in Table 0.1 the list of publications and their corresponding chapter. In Appendix A, we give a list of conferences we participated with the kind of contribution realized. In Table 0.2, we present the list of our contributions per chapter.

In Figure 0.1, we propose a reading help guide: Chapter 1 is the entry point and the directed edges indicates an order of reading: $A \to B$ indicates that Chapter A should be read before Chapter B. In this case, the reader can refer to the prerequisites placed at the head of each chapter to know if only specific sections of Chapter A are needed.

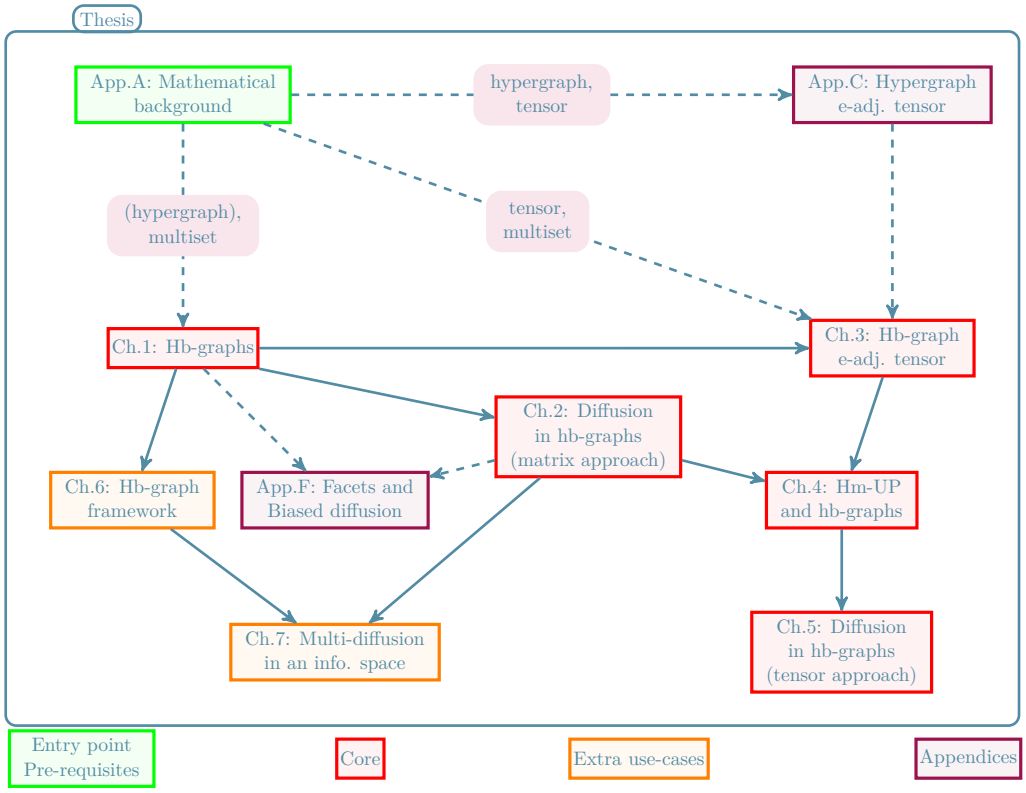


Figure 0.1.: A reading help guide.

Reference	Title	Chapter Appendix
	Journal	
[OLGMM18d]	Hypergraph Modeling and Visualisation of Complex Co-occurrence Networks. Electronic Notes in Discrete Mathematics.	Chap. 6
[OLGMM18a]	On Adjacency and e-Adjacency in General Hypergraphs: Towards a New e-Adjacency Tensor. Electronic Notes in Discrete Mathematics.	App. E.1
[OLGMM19c]	On Hb-graphs and their Application to General Hypergraph e-adjacency Tensor. MCCCC32 proceedings Journal of Combinatorial Mathematics and Combinatorial Computing.	App. B Chap. 1 Chap. 3
	Conference proceedings	
[OLGMM18c]	Exchange-Based Diffusion in Hb-Graphs: Highlighting Complex Relationships. IEEE CBMI 2018.	Chap. 2
[OLGMM20]	The HyperBagGraph DataEdron: An Enriched Browsing Experience of Multimedia Datasets. Springer Lecture Notes in Computer Science. Springer LNCS SOFSEM 2020.	App. D.1
	Pre-prints	
[OLGMM17a]	Adjacency and Tensor Representation in General Hypergraphs Part 1: e-adjacency Tensor Uniformisation Using Homogeneous Polynomials.	App. E.1
[OLGMM18b]	Adjacency and Tensor Representation in General Hypergraphs. Part 2: Multisets, Hb-graphs and Related e-adjacency Tensors	Chap. 1 Chap. 3
[OLGMM17b]	Networks of Collaborations: Hypergraph Modeling and Visualisation	App. B
[Ouv20]	Hypergraphs: an Introduction and Review	App. B Chap. 3
[OGMM20]	Tuning Ranking in Co-occurrence Networks with General Biased Exchange-based Diffusion on Hyper-bag-graphs	App. B
	In submission to a Journal	
[OLGMM19a]	Exchange-Based Diffusion in Hb-Graphs: Highlighting Complex Relationships. Extended version. Submitted to MTAP.	Chap. 2

Table 0.1.: List of publications related to this Thesis—as of 20.03.2020.

Corpus	
Chapter 1	• Introduction of hb-graphs: all the concepts in the chapter but the related work.
Chapter 2	• Exchange-based diffusion in co-occurrence networks
Chapter 3	• e-adjacency tensors of general hb-graphs.
Chapter 4	• Impact of the m-uniformisation process on diffusion.
Chapter 5	• A layered Laplacian for general hb-graphs
Chapter 6	• Hb-graph framework for co-occurrence networks modeling.
Chapter 7	• Multi-diffusion principle in an information space; • WT-MC4 for aggregation ranking.
Appendix	
Appendix B	• Review on hypergraphs; • Visualisation of large hypergraphs.
Appendix C.2	• Homomorphisms of hb-graphs
Appendix E.1	• e-adjacency tensor for general hypergraphs
Appendix D.1	• Facets and biased diffusion.

Table 0.2.: List of personal contributions related to this Thesis.

1. An introduction to hb-graphs

Highlights

This chapter is based on [OLGMM18b, OLGMM18c, OLGMM19a, OLGMM19c].

In this Chapter and in Appendix C.2, we extend the work on hyper-bag-graphs (hb-graphs for short) introduced in [OLGMM18c], and that has been extensively presented in [OLGMM18b, OLGMM19c]. Hb-graphs extend hypergraphs by substituting families of subsets of a vertex set by families of multisets of same universe, playing the role of the vertex set. After their introduction, we extend to hb-graphs some of the results found in [Bre13] for hypergraphs. The reader, not familiar with the mathematical notions at stake in this Chapter, can refer to Appendix B: more particularly the sections concerning hypergraphs—Section B.1—and multisets—Section B.2.

Prerequisites: Section B.1 and B.2 of Appendix B.

1.1. Motivation

There are different motivations that converge for the introduction of hb-graphs.

A very first motivation is that multisets are appropriate to enhance repetition of elements on a given universe. In Appendix B.2.5, multisets are shown to be intensively used in different fields as we already mentioned.

Moreover, in the case of complex co-occurrences networks, a second motivation is that the co-occurrences themselves can require this kind of repetitions, by the way the co-occurrences are built—such an example is given in Section C.1 of Appendix C—, or individual weighting, naturally enhancing families of multisets on a given universe.

A third motivation for introducing hb-graphs is related to the representation of large hypergraphs as it is presented in Section B.1.8 of Appendix B; large hypergraph representations call for a coarsening in their generative model for enhancing a representation

where the vertices and hyperedges are spread sufficiently over the surface of representation. One of the approach that can be taken is the coarsening by diffusion: the diffusion can be achieved using the incident matrix, but in order to capture higher order relationships a tensor approach is required that captures the information of adjacency. If for uniform structures, the tensors are well defined, in general hypergraphs it is still an ongoing work. In [OLGMM17a]—reproduced in Section E.1 of Appendix E—in order to build a tensor for general hypergraphs, we introduce a Hypergraph Uniformisation Process (HUP) for merging successively the different consecutive layers—possibly empty—of uniform hypergraphs ordered with increasing cardinality of their hyperedge, adding at each step one (or several if some layers are empty) different special vertex to reach the level of uniformity of the next non-empty layer. This process has been coupled to an iterative Polynomial Homogenization Process. In fact, the PHP could be achieved in only one step, by adding only one variable. But, to keep an interpretation similar to the Hypergraph Uniformisation Process, we have to get a structure that accepts the repetition of vertices. Hence, the following researching question:

Research question 1.1. *How to extend hypergraphs with family of multisets?*

We start by reviewing the related work that gives a collection of elements that are first steps toward hb-graphs, and then expose the concept of hb-graphs as a family of multisets on a given universe; we focus on families of natural multisets on a given universe, where elements are seen as repetition of a given element. In [GJ12], the authors express this need in real datasets, where two physical objects can be seen "as the same or equal, if they are indistinguishable, but possibly separate, and identical if they physically coincide". We revisit systematically the common definitions and properties of hypergraphs to extend them to hb-graphs. We present in this Chapter some first applications. The remainder of this Thesis addresses the analysis of such structures, some applications and use-cases.

1.2. Related work

Handling structures similar to hypergraphs but having a "hyperedge-based" weighting of vertices occur quite a few times in the literature. For instance in [BAD13], the authors show that retrieving information from a textual dataset is improved when using a modified random walk taking into account a "hyperedge-based" weighting of vertices compared to the binary approach. Implicitly, multisets are behind this approach.

In [GJ12], the authors define a multiset topology by considering a collection of multisets in the power set of a given multiset. The power set of a given multiset is defined as the support of the power mset of an m-set that corresponds to the mset of all submsets of that multiset (which implies redundancy). They then study the properties of these multiset topologies. It is a strong background for our work, but multiset topologies include all submset of a given collection in that collection. Multiset topologies have to be seen as a potential extension of simplicial complexes, which are a particular case of hypergraphs as exposed in [Ouv20].

In [PZ14], the authors introduce what they call *[PZ-]*[1]multi-hypergraphs[2] using multisets, allowing repetitions of vertices in the hyperedges, but where two hyperedges cannot be duplicates. *[PZ-]*Multi-hypergraphs are a particular case of hb-graphs as we define them in this Chapter: we call them natural hb-graphs with no repeated hb-edge. Waiting for the definitions given in Section 1.3, we call them temporarily PZ-multi-hypergraphs.

Definition 1.1. *A PZ-multi-hypergraph $\ddot{\ddot{\mathcal{H}}}$ is a pair (V, E), where E is a set of multisets of V. The elements of V are called the vertices and the elements of E are called the edges.*

In [PZ14], when the hyperedges are all of same cardinality k the PZ-multi-hypergraph is said k-uniform, and called k-multigraph[3]. (PZ-)Multi-hypergraphs are in fact a particular case of hb-graphs as we define in this Chapter. They define some additional notions such as a chain in a (PZ-)multi-hypergraph, the notion of connected (PZ-)multi-hypergraph and the notion of k-(PZ-)multi-hypergraphs.

Independently in [KPP$^+$19]—at the same time we were introducing hb-graphs in [OLGMM18b]—the authors consider a hypergraph where the hyperedges are multisets, transforming the initial definition of hypergraphs, to extend the Cheng-Lu model to hypergraphs, to achieve clustering via hypergraph modularity. They use a family of multisets and define the degree of a vertex as the sum of multiplicities and the size of a hyperedge as the sum of the multiplicities of its elements. They obtain good results with their proposed modularity getting a smaller number of hyperedges cut compared to the one achieved with the 2-section of the hypergraph.

In [CR19], published after [OLGMM18c], a hypergraph with hyperedge-dependent vertex weights is defined by considering a quadruple $\mathcal{H} = (V, E, \omega, \gamma)$ where ω is the edge weight vector, and γ is refined in a weight $\gamma_e(v)$ for every hyperedge $e \in E$. The authors are then using implicitly multisets, but without considering the related algebra. In a recent paper [PVAdST19], the authors introduce a continuous incident matrix for multimedia retrieval, which is no more than our hb-graph incident matrix.

1.3. Generalities

Let $V = \{v_i : i \in [\![n]\!]\}$ be a nonempty finite set.

A **hyper-bag-graph or hb-graph** for short over V is defined in [OLGMM18c] as a family of msets with universe V and support a subset of V. The msets are called the **hb-edges** and the elements of V the **vertices**.

We consider for the remainder of the Thesis a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$, with $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{E} = (\mathfrak{e}_j)_{j \in [\![p]\!]}$ the family of its hb-edges.

Each hb-edge $\mathfrak{e}_j \in \mathfrak{E}$ is of universe V and has a multiplicity function associated to it: $m_{\mathfrak{e}_j} : V \to \mathbb{W}$ where $\mathbb{W} \subseteq \mathbb{R}^+$. When the context is clear the notation m_j is used for $m_{\mathfrak{e}_j}$ and m_{ij} for $m_{\mathfrak{e}_j}(v_i)$.

[1] *[...]* is used in italic as abbreviation for [Thesis Author's Note: ...]

[2] Multi-hypergraph is in fact polysemic: multi-hypergraphs originally represent hypergraphs $\mathcal{H} = (V, E)$ where the repetition of hyperedges is authorized in E i.e. E is considered as a family [Bre13] or a multiset [CF88], which is the direct extension of multi-graph.

[3] It is also a polysemy: in [Maj87] k-multigraphs are multi-graphs—where multiple edges between a couple of vertices can occur—which are also k-graphs—i.e. graphs that are k-regular.

A hb-graph is said to be with **no repeated hb-edge** if:

$$\forall j_1 \in [\![p]\!], \forall j_2 \in [\![p]\!] : \mathfrak{e}_{j_1} = \mathfrak{e}_{j_2} \Rightarrow j_1 = j_2.$$

A hb-graph where each hb-edge is a natural mset is called a **natural hb-graph**.

The **empty hb-graph** is the hb-graph with an empty set of vertices. A **trivial hb-graph** is a hb-graph with a non empty set of vertices and an empty family of hb-edges.

For a general hb-graph, every hb-edge has to be seen as a weighted system of vertices, where the weights of each vertex are hb-edge dependent. In a natural hb-graph, the multiplicity function can be viewed as a duplication of the vertices.

In [PZ14], the authors have introduced what they call abusively multi-hypergraphs where the hyperedge set is composed of multisets of vertices. It corresponds to natural hb-graphs with no repeated hb-edges, name that we keep, as multi-hypergraphs are hypergraphs where the subsets of vertices the hyperedges correspond to can be repeated, i.e. constitute either a multiset of subsets of vertices [CF88] or a family of subsets of vertices [Bre13].

The **order** of a hb-graph $\mathfrak{H}$—written $O(\mathfrak{H})$—is given by: $O(\mathfrak{H}) \triangleq \sum_{i \in [\![n]\!]} \max_{\mathfrak{e} \in \mathfrak{E}} (m_{\mathfrak{e}}(v_i))$.

The **size** of a hb-graph corresponds to the number of its hb-edges.

If $\bigcup_{j \in [\![p]\!]} \mathfrak{e}_j^\star = V$, then the hb-graph is said with no isolated vertices. Otherwise, the elements of $V \setminus \bigcup_{j \in [\![p]\!]} \mathfrak{e}_j^\star$ are called the **isolated vertices**. They correspond to elements of hb-edges with zero-multiplicity for all hb-edges.

A **hypergraph** is a natural hb-graph where the vertices of the hb-edges have multiplicity one for any vertex of their support and zero otherwise.

The **support hypergraph** of a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ is the hypergraph whose vertices are the ones of the hb-graph and whose hyperedges are the support of the hb-edges. We write $\underline{\mathfrak{H}} \triangleq (V, E)$ with $E \triangleq (\mathfrak{e}^\star)_{\mathfrak{e} \in \mathfrak{E}}$, the support hypergraph of $\mathfrak{H}$.

We note that given a hypergraph $\mathcal{H}$, an infinite set $\mathcal{C}_\mathcal{H}$ of hb-graphs can be generated that all have this hypergraph as support. A hb-edge family is attached to each of the hb-graphs in $\mathcal{C}_\mathcal{H}$: each hb-edge support corresponds at least to one hyperedge in $\mathcal{H}$ and, reciprocally, each hyperedge of $\mathcal{H}$ is at least the support of one hb-edge per hb-graph of $\mathcal{C}_\mathcal{H}$. The unicity of the correspondence is ensured only for hypergraphs and hb-graphs without repeated hyperedges.

In this case, an equivalence relation $\mathcal{R}$ can be defined that puts in relation two hb-graphs if they have same hypergraph as hb-graph support. Hence, considering the quotient of the set of all hb-graphs with no-repeated hb-edges by $\mathcal{R}$, it is isomorph to the set of hypergraphs with no-repeated hyperedge.

The **m-rank** of a hb-graph—written $r_m(\mathfrak{H})$—is by definition[4]: $r_m(\mathfrak{H}) \triangleq \max_{\mathfrak{e} \in \mathfrak{E}} \#_m \mathfrak{e}$.

The **rank** of a hb-graph $\mathfrak{H}$—written $r(\mathfrak{H})$—is the rank of its support hypergraph $\underline{\mathfrak{H}}$.

The **m-co-rank** of a hb-graph—written $\mathrm{cr}_m(\mathfrak{H})$—is by definition:$\mathrm{cr}_m(\mathfrak{H}) \triangleq \min_{\mathfrak{e} \in \mathfrak{E}} \#_m \mathfrak{e}$.

[4]see Section B.2.1 for the definition of the notation $\#_m \mathfrak{e}$.

The **co-rank** of a hb-graph $\mathfrak{H}$—written $\mathrm{cr}(\mathfrak{H})$—is the co-rank of its support hypergraph $\underline{\mathfrak{H}}$.

The **global m-cardinality** of a hb-graph $\mathfrak{H}$—written $\#_m\mathfrak{H}$—is the sum of the m-cardinality of its hb-edges.

A hb-graph is said **k-m-uniform** if all its hb-edges have the same m-cardinality k.

A hb-graph is said **k-uniform** if its support hypergraph is k-uniform.

Proposition 1.1. *A hb-graph $\mathfrak{H}$ is k-m-uniform, if and only if: $r_m(\mathfrak{H}) = cr_m(\mathfrak{H}) = k$.*

We can still refer for vertices of a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ to the **degree of** $v \in V$ written $\deg(v) = d(v)$: it corresponds to the degree of the same vertex in the support hypergraph $\underline{\mathfrak{H}}$. The **maximal degree** of a hb-graph $\mathfrak{H}$ is written Δ and corresponds to the maximal degree of the support hypergraph $\underline{\mathfrak{H}}$.

Nonetheless, in hb-graphs, due to the multiplicity function attached to each hb-edge, we can consider another kind of degree. To complete its proper definition, we define the **hb-star** $\mathfrak{h}(v)$ of a vertex $v \in V$ as the multiset: $\mathfrak{h}(v) \triangleq \left\{\mathfrak{e}^{m_{\mathfrak{e}}(v)} : \mathfrak{e} \in \mathfrak{E} \wedge v \in \mathfrak{e}^\star\right\}$. The support of $\mathfrak{h}(v)$ is exactly the star of this vertex in the support hypergraph $\underline{\mathfrak{H}}$ and the cardinality of $\mathfrak{h}^\star(v)$ is exactly the degree of v.

The **m-degree of a vertex** $v \in V$ of a hb-graph $\mathfrak{H}$ is then defined as the m-cardinality of the hb-star attached to this vertex:

$$\deg_m(v) \triangleq \#_m\mathfrak{h}(v).$$

We also consider the **maximal m-degree** of a hb-graph $\mathfrak{H}$; we write it:

$$\Delta_m \triangleq \max_{v\in V} \deg_m(v).$$

A hb-graph having all its hb-edges of the same m-degree k is said **m-regular** or **k-m-regular**. A hb-graph is said **regular** if its support hypergraph is regular.

It is immediate that:

Proposition 1.2. *For any vertex $v \in V$ of a natural hb-graph: $d(v) \leqslant d_m(v) \leqslant \Delta_m$.*

This property is not true for non-natural hb-graphs.

Proposition 1.3. *If a hb-graph with a non-negative multiplicity function range is such that: $d(v) = d_m(v)$ for all its vertices, then this hb-graph is a hypergraph.*

We define now the **dual** of a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ as the hb-graph $\tilde{\mathfrak{H}} \triangleq \left(\tilde{V}, \tilde{\mathfrak{E}}\right)$ such that its set of vertices $\tilde{V} \triangleq \{\tilde{v_j} : j \in [\![p]\!]\}$ is in bijection $f : \mathfrak{E} \to \tilde{V}$ with the family of hb-edges $\mathfrak{E}$ of $\mathfrak{H}$ such that:

$$\forall \tilde{v_j} \in \tilde{V}, \exists! \mathfrak{e}_j \in \mathfrak{E} : \tilde{v_j} = f(e_j).$$

And its family of hb-edges $\tilde{\mathfrak{E}} \triangleq (\tilde{e_i})_{i\in[\![n]\!]}$ is in bijection $g : V \to \tilde{\mathfrak{E}}$ such that $g(v_i) = \tilde{\mathfrak{e}_i}$ for all $i \in [\![n]\!]$, where:

$$\tilde{\mathfrak{e}_i} \triangleq \left\{ \tilde{v_j}^{m_{\mathfrak{e}_j}(v_i)} : j \in [\![p]\!] \wedge \tilde{v_j} = f(\mathfrak{e}_j) \wedge v_i \in \mathfrak{e}_j^\star \right\}.$$

We summarize some dualities between a hb-graph and its dual in Table 1.1.

	$\mathfrak{H}$	$\widetilde{\mathfrak{H}}$
Vertices	$v_i,\ i \in [\![n]\!]$	$\tilde{v_j} = f(\mathfrak{e}_j),\ j \in [\![p]\!]$
Hb-edges	$\mathfrak{e}_j,\ j \in [\![p]\!]$	$\tilde{\mathfrak{e}_i} = g(v_i),\ i \in [\![n]\!]$
Multiplicity	$v_i \in \mathfrak{e}_j$ with $m_{\mathfrak{e}_j}(v_i)$	$\tilde{v_j} \in \tilde{\mathfrak{e}_i}$ with $m_{\tilde{\mathfrak{e}_i}}(v_j)$
m-degrees vs m-cardinality	$d_m(v_i)$ $\#_m \mathfrak{e}_i$	$\#_m \tilde{\mathfrak{e}_i}$ $d_m(\tilde{v_j})$
m-uniformity vs m-regularity	k-m-uniform k-m-regular	k-m-regular k-m-uniform

Table 1.1.: Dualities between a hb-graph and its dual.

1.4. Additional concepts for natural hb-graphs

1.4.1. Numbered copy hypergraph of a natural hb-graph

In natural hb-graphs, hb-edge multiplicity functions have their range which is a subset of $\mathbb{N}$. The vertices in a hb-edge with multiplicity strictly greater than 1 can be seen as copies of the original vertex.

Indexing the copies of the original vertex makes them seen as "numbered" copies. We consider a vertex v_i belonging to two hb-edges $\mathfrak{e}_{j_1}$ and $\mathfrak{e}_{j_2}$ of the hb-graph $\mathfrak{H} = (V, \mathfrak{E})$, with a multiplicity $m_{\mathfrak{e}_{j_1}}$ in $\mathfrak{e}_{j_1}$ and $m_{\mathfrak{e}_{j_2}}$ in $\mathfrak{e}_{j_2}$. Then $\mathfrak{e}_{j_1} \cap \mathfrak{e}_{j_2}$ holds $\min\left(m_{\mathfrak{e}_{j_1}}(v_i), m_{\mathfrak{e}_{j_2}}(v_i)\right)$ copies: by convention, the ones "numbered" from 1 to $\min\left(m_{\mathfrak{e}_{j_1}}, m_{\mathfrak{e}_{j_2}}\right)$. The remaining copies will be either in $\mathfrak{e}_{j_1}$ xor $\mathfrak{e}_{j_2}$ depending on which multiset has the highest multiplicity for v_i.

More generally, we define the **numbered-copy set** of a natural multiset $\mathfrak{A}_m = \{x_i^{m_i} : i \in [\![n]\!]\}$ as the copy-set $\breve{\mathfrak{A}_m} \triangleq \left\{ [x_{i\,j}]_{m_i} : i \in [\![n]\!] \right\}$ where: $[x_{i\,j}]_{m_i}$ is a shortcut to indicate the numbered copies of the original element x_i: $x_{i\,1}$ to $x_{i\,m_i}$ and j is designated as the copy number of the element x_i.

We define the **maximum multiplicity function** of $\mathfrak{H}$ as the function $m : V \to \mathbb{N}$ such that for all $v \in V$: $m(v) \triangleq \max_{\mathfrak{e}\in\mathfrak{E}} m_{\mathfrak{e}}(v)$ and consider the numbered-copy-set $\breve{V} \triangleq \left\{ [v_{i\,j}]_{m(v_i)} : i \in [\![n]\!] \right\}$ of the multiset $\left\{ v_i^{m(v_i)} : i \in [\![n]\!] \right\}$.

Then each hb-edge $\mathfrak{e}_k = \left\{ v_{i_j}^{m_{k\,i_j}} : j \in [\![k]\!] \wedge i_j \in [\![n]\!] \right\}$ is associated to a copy-set / equivalency relation $< e_{k\,0}, \rho_k >$ which elements are in $\breve{V}$ with the smallest copy numbers possible for any vertex in $\mathfrak{e}_k$. The hypergraph $\mathfrak{H}_0 \triangleq \left(\breve{V}, E_0\right)$ where $E_0 \triangleq (e_{k\,0})_{k\in[\![p]\!]}$ is called the **numbered-copy-hypergraph** of $\mathfrak{H}$.

Proposition 1.4. *A numbered-copy-hypergraph is unique for a given hb-graph.*

Proof. It is immediate by the way the numbered-copy-hypergraph is built from the hb-graph. □

Allowing duplicates to be numbered prevents ambiguities; nonetheless, it has to be seen as a conceptual approach since duplicates are entities that are actually not discernible.

1.4.2. Paths, distance and connected components

Defining a path in a hb-graph is not straightforward due to vertex duplication. We define two concepts of **m-paths**. The first imposes to the duplicate of a vertex inside a path to occur strictly at the intersection of two consecutive hb-edges. The second relaxes this hypothesis by allowing the vertex to be any duplicate in the union of two consecutive hb-edges—with of course one of the duplicate in the intersection.

More precisely, a **strict m-path** $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ in a hb-graph $\mathfrak{H}$ from a vertex u to a vertex v is a vertex / hb-edge alternation with s hb-edges $\mathfrak{e}_{j_k}$ such that: $\forall k \in [\![s]\!], j_k \in [\![p]\!]$ and $s+1$ vertices v_{i_k} with $\forall k \in \{0\} \cup [\![s]\!], i_k \in [\![n]\!]$ and such that $v_{i_0} = u$, $v_{i_s} = v$, $u \in \mathfrak{e}_{j_1}$ and $v \in \mathfrak{e}_{j_s}$ and that for all $k \in [\![s-1]\!]$, $v_{i_k} \in \mathfrak{e}_{j_k} \cap \mathfrak{e}_{j_{k+1}}$.

A **large m-path** $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ in a hb-graph $\mathfrak{H}$ from a vertex u to a vertex v is a vertex / hb-edge alternation with s hb-edges $\mathfrak{e}_{j_k}$ such that: $\forall k \in [\![s]\!], j_k \in [\![p]\!]$ and $s+1$ vertices v_{i_k} with $\forall k \in \{0\} \cup [\![s]\!], i_k \in [\![n]\!]$ and such that $v_{i_0} = u$, $v_{i_s} = v$, $u \in \mathfrak{e}_{j_1}$ and $v \in \mathfrak{e}_{j_s}$ and that for all $k \in [\![s-1]\!]$, $v_{i_k} \in \mathfrak{e}_{j_k} \cup \mathfrak{e}_{j_{k+1}}$ and that there exists a copy of v_{i_k} in $\mathfrak{e}_{j_k} \cap \mathfrak{e}_{j_{k+1}}$.

The **length of a m-path** from u to v is the number of hb-edges it traverses; given a path $\mathscr{P}$, we write $l(\mathscr{P})$ its length. It holds that if $\mathscr{P} = v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$, we have whatever the path is—strict or large—: $l(\mathscr{P}) = s$.

In a path $\mathscr{P} = v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$, the vertices v_{i_k}, $k \in [\![s-1]\!]$ are called the **interior vertices** of the m-path and v_{i_0} and v_{i_s} are called the **extremities** of the m-path.

If the extremities are different copies of the same object, then the m-path is said to be an **almost cycle**. If the extremities designate exactly the same copy of one object, the m-path is said to be a **cycle**.

Proposition 1.5. *1. For a strict m-path, there are* $\prod_{k\in[\![s-1]\!]} m_{\mathfrak{e}_{j_k}\cap\mathfrak{e}_{j_{k+1}}}(v_{i_k})$ *possibilities of choosing the interior vertices along a given m-path* $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ *and* $m_{\mathfrak{e}_{j_1}}(v_{i_0})\left(\prod_{k\in[\![s-1]\!]} m_{\mathfrak{e}_{j_k}\cap\mathfrak{e}_{j_{k+1}}}(v_{i_k})\right) m_{\mathfrak{e}_{j_s}}(v_{i_s})$ *possible strict m-paths in between the extremities.*

2. For a large m-path, there are $\prod_{k\in[\![s-1]\!]} m_{\mathfrak{e}_{j_k}\cup\mathfrak{e}_{j_{k+1}}}(v_{i_k})$ *possibilities of choosing the interior vertices along a given m-path* $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ *and* $m_{\mathfrak{e}_{j_1}}(v_{i_0})\left(\prod_{k\in[\![s-1]\!]} m_{\mathfrak{e}_{j_k}\cup\mathfrak{e}_{j_{k+1}}}(v_{i_k})\right) m_{\mathfrak{e}_{j_s}}(v_{i_s})$ *possible large m-paths in between the extremities.*

3. *As large m-paths between two extremities for a given sequence of interior vertices and hb-edges include strict m-paths, we often refer as* ***m-paths*** *for large m-paths.*
4. *If an m-path exists from u to v, then an m-path also exists from v to u.*

Proof. All these results come directly of combinatorics over the multisets involved in the different paths. □

An m-path $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ in a hb-graph corresponds to a unique path in the hb-graph support hypergraph called the **support path**.

Proposition 1.6. *Every m-path $v_{i_0}\mathfrak{e}_{j_1}v_{i_1}\dots\mathfrak{e}_{j_s}v_{i_s}$ traversing the same hb-edges and having similar vertices as intermediate and extremity vertices share the same support path.*

Proof. The common support path is then $v_{i_0}\mathfrak{e}^\star_{j_1}v_{i_1}\dots\mathfrak{e}^\star_{j_s}v_{i_s}$. □

The notion of distance is similar to the one defined for hypergraphs.

The **hb-graph distance** $d(u,v)$ between two vertices u and v of a hb-graph is the length of the shortest m-path between u and v, if there exists, that can be found in the hb-graph. In the case where there is no path between the two vertices, they are said disconnected, and we set: $d(u,v)=+\infty$. A hb-graph is said **connected** if its support hypergraph is connected, disconnected otherwise.

A **connected component** of a hb-graph is the maximal mset of vertices for which there exists a m-path in between every pair of vertices of the mset in the hb-graph.

Proposition 1.7. *A connected component of a hb-graph is a connected component of one of its copy hypergraph.*

The **diameter** of a hb-graph $\mathfrak{H}$—written $\operatorname{diam}(\mathfrak{H})$—is defined as:

$$\operatorname{diam}(\mathfrak{H}) \triangleq \max_{u,v\in V} d(u,v).$$

1.4.3. Adjacency

In [OLGMM18a], we have introduced different concepts of adjacency for hypergraphs. We extend here those concepts to hb-graphs, with some refinements.

We consider a hb-graph $\mathfrak{H}=(V,\mathfrak{E})$, a positive integer k and k vertices not necessarily distinct belonging to V. We write $\mathfrak{V}_{k,m}$ the mset consisting of these k vertices with multiplicity function m.

The k vertices of $\mathfrak{V}_{k,m}$ are said ***k*-adjacent** in $\mathfrak{H}$ if there exists $\mathfrak{e}\in\mathfrak{E}$ such that $V_{k,m}\subseteq\mathfrak{e}$.

Considering a hb-graph $\mathfrak{H}$ of m-rank $\overline{k}=r_m(\mathfrak{H})$, the hb-graph cannot handle any k-adjacency for k strictly greater than $r_m(\mathfrak{H})$. This maximal k-adjacency is called the **$\overline{k}$-adjacency** of $\mathfrak{H}$.

We consider now a hb-edge $\mathfrak{e}$ of a hb-graph $\mathfrak{H}$. The vertices in the support of $\mathfrak{e}$ are said **$e^\star$-adjacent**. For natural hb-graphs, we say that the vertices with nonzero multiplicity in a hb-edge are ***e*-adjacent**.

We can remark that $e^\star$-adjacency does not support redundancy of vertices while e-adjacency in natural hb-graphs allows it. e-adjacency in natural hb-graphs takes into account the multiplicity of the different vertices, which is not the case of $e^\star$-adjacency.

In a non-natural hb-graph, the vertices in the support of hb-edges with multiplicity different from 1 cannot be seen as copies of one another, hence only the $e^\star$-adjacency is valid for this kind of hb-graphs.

Two hb-edges are said **incident** if the intersection of their respective support is not empty.

1.4.4. Sum of two hb-graphs

Let $\mathfrak{H}_1 = (V_1, \mathfrak{E}_1)$ and $\mathfrak{H}_2 = (V_2, \mathfrak{E}_2)$ be two hb-graphs.

The **sum of two hb-graphs** $\mathfrak{H}_1$ and $\mathfrak{H}_2$ is the hb-graph, written $\mathfrak{H}_1 + \mathfrak{H}_2$, that has:

- $V_1 \cup V_2$ as vertex set;
- $\mathfrak{E}_1 + \mathfrak{E}_2$ as hb-edge family: hb-edges are obtained from the hb-edges of $\mathfrak{E}_1$ (respectively $\mathfrak{E}_2$) with the same multiplicity for vertices of V_1 (respectively V_2) but such that for each hb-edge in $\mathfrak{E}_1$ (respectively $\mathfrak{E}_2$) the universe is extended to $V_1 \cup V_2$ and the multiplicity function is extended such that $\forall v \in V_2 \backslash V_1 : m(v) = 0$ (respectively $\forall v \in V_1 \backslash V_2 : m(v) = 0$);
- $\mathfrak{H}_1 + \mathfrak{H}_2 \overset{\Delta}{=} (V_1 \cup V_2, \mathfrak{E}_1 + \mathfrak{E}_2)$.

This sum is said direct if $\mathfrak{E}_1 + \mathfrak{E}_2$ does not contain any new pair of repeated hb-edge other than the ones already existing in $\mathfrak{E}_1$ and in $\mathfrak{E}_2$, and that $\mathfrak{E}_1$ and $\mathfrak{E}_2$ do not have any common hb-edges. In this case, the sum is written $\mathfrak{H}_1 \oplus \mathfrak{H}_2$ and called the **direct sum of the two hb-graphs** $\mathfrak{H}_1$ and $\mathfrak{H}_2$.

1.4.5. An example

Example 1.4.1. *We consider* $\mathfrak{H} = (V, \mathfrak{E})$*, with* $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ *and,* $\mathfrak{E} = \{\mathfrak{e}_1, \mathfrak{e}_2, \mathfrak{e}_3, \mathfrak{e}_4\}$ *with:* $\mathfrak{e}_1 = \{v_1^2, v_4^2, v_5^1\}$, $\mathfrak{e}_2 = \{v_2^3, v_3^1\}$, $\mathfrak{e}_3 = \{v_3^1, v_5^2\}$ *and* $\mathfrak{e}_4 = \{v_6^1\}$.

It holds:

	$\mathfrak{e}_1$	$\mathfrak{e}_2$	$\mathfrak{e}_3$	$\mathfrak{e}_4$	$d_m(v_i)$	$\max\left\{m_{\mathfrak{e}_j}(v_i)\right\}$
v_1	2	0	0	0	2	2
v_2	0	3	0	0	3	3
v_3	0	1	1	0	2	1
v_4	2	0	0	0	2	2
v_5	1	0	2	0	3	2
v_6	0	0	0	1	1	1
v_7	0	0	0	0	0	0
$\#_m \mathfrak{e}_j$	5	4	3	1		

Therefore, the order of $\mathfrak{H}$ is $O(\mathfrak{H}) = 2+3+1+1+2+1+0 = 10$ and its size is $|\mathfrak{E}| = 4$.

v_7 is an isolated vertex.

$\mathfrak{e}_1$ and $\mathfrak{e}_3$ are incident as well as $\mathfrak{e}_3$ and $\mathfrak{e}_2$. $\mathfrak{e}_4$ is not incident to any hb-edge.

v_1, v_4 and v_5 are $e^\star$-adjacent as they are together in $\mathfrak{e}_1^\star$.

v_1^2, v_4^2 and v_5^1 are e-adjacent as they are together with the corresponding multiplicities in $\mathfrak{e}_1$.

The dual of $\mathfrak{H}$ is the hb-graph: $\widetilde{\mathfrak{H}} = \left(\widetilde{V}, \widetilde{\mathfrak{E}}\right)$ with:

- $\widetilde{V} = \{e_1, e_2, e_3, e_4\}$ with $f(e_i) = \mathfrak{e}_i$ for $i \in [\![4]\!]$;
- $\widetilde{\mathfrak{E}} = \{\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_3, \mathfrak{v}_4, \mathfrak{v}_5, \mathfrak{v}_6, \mathfrak{v}_7\}$ with: $\mathfrak{v}_1 = \mathfrak{v}_4 = \{e_1^2\}$; $\mathfrak{v}_2 = \{e_2^3\}$; $\mathfrak{v}_3 = \{e_2^1, e_3^1\}$; $\mathfrak{v}_5 = \{e_1^1, e_3^2\}$; $\mathfrak{v}_6 = \{e_4^1\}$; $\mathfrak{v}_7 = \emptyset$.

$\widetilde{\mathfrak{H}}$ has duplicated hb-edges and one empty hb-edge.

(a) Multipartite representation of a hb-graph.

(b) Multipartite representation of its dual.

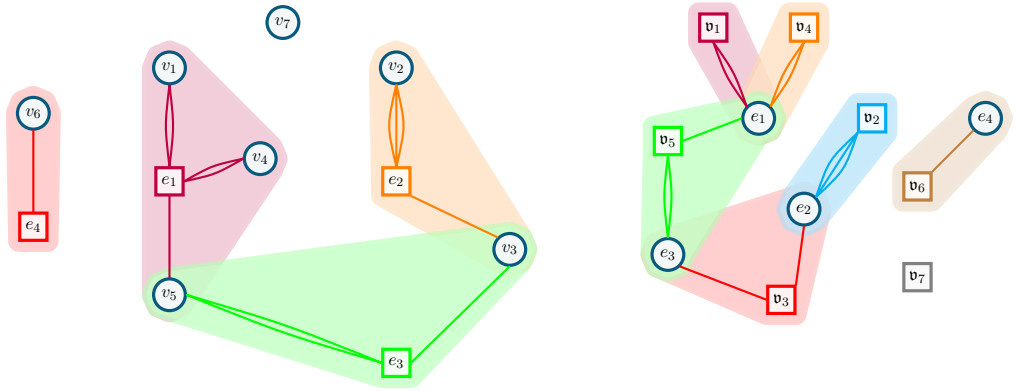


Figure 1.1.: Hb-graph (a) and its dual (b) given in Example 1.4.1 in their multipartite graph representation with in superposition the corresponding support hypergraph in the Euler representation.

1.4.6. Hb-graph representations

Representing hb-graphs can be thought along the two main standards found for hypergraphs as presented in Section B.1.8 of Appendix B and introduced in [Mäk90]: the subset standard and the edge standard. But both representations have to be adapted to support multisets.

1.4.6.1. Subset standard

In the subset standard for hypergraphs, a contour line is drawn to surround the vertices of a hyperedge. Each hyperedge is then represented using these contour lines. Depend-

ing on how their intersection is represented, we obtain a Venn diagram or an Euler diagram representation of the hypergraph. A Venn diagram systematically represents each possible intersection between hyperedges, while an Euler diagram addresses only the intersections that are needed for the representation. Hence, the Euler diagram is often preferred as this representation scales up a bit better than the Venn diagram does; but neither representations scale up to large hypergraphs.

When moving to multisets, a contour line is also drawn around the vertices that are now duplicated. In [Rad15], two Venn diagram representations of multisets are proposed for the representation of 2 and 3 multisets: a simplified representation where the parts are not disjoint and a complex representation where the parts are disjoint. Scaling up the number of multisets seems to be hard to achieve. Euler representation of multisets can be based on this work: a simplified and a complex representation can be drawn to depict only the parts needed to be represented. It simplifies the Venn representation and helps to scale up to somewhat larger hb-graphs.

1.4.6.2. Edge standard

For hypergraphs, there are two main representations in the edge standard: the clique representation and the extra-node representation, as presented in Section B.1.8. The clique representation transforms the hypergraph in its 2-section graph joining every pair of vertices of a hyperedge by an edge, while the extra-node representation corresponds to the incident graph—also called Levi graph—which is the graph obtained by representing each hyperedge as an extra-vertex and joining it to each vertex of the corresponding hyperedge.

For hb-graphs, the 2-section graph is always representable by considering each hb-edge support: but the quantity of information to display as well as its quality is not optimal in this representation.

Hence, we propose other alternatives based on the extra-node representation. Each hb-edge is represented by an extra-node. For natural hb-graph a first representation called the **extra-node multipartite representation** is achieved by joining each vertex of the hb-edge to the extra node with a number of edges that corresponds to the vertex multiplicity—which can be considered as a multipartite version of the incident graph of the hb-graph that takes into account the multiplicity of vertices through the number of edges linking the vertex to the extra-node representing the hb-edge. This representation does not fit for non-natural hb-graphs; moreover, it is hard to scale up when the values of multiplicities increase. We give an example of such a representation in Figure 1.1 using the hb-graph and its dual given in Example 1.4.1.

For hb-graphs with non-negative multiplicity ranges, we propose a second representation based on the extra-node representation of the support hypergraph, but where the thickness of the edges linking the vertices of the hb-edge with its extra-node are proportional either directly to the multiplicity (absolute version) or to the relative multiplicity of the vertex in the hb-edge (unnormalized version).

The **relative multiplicity** of a vertex v_i in a hb-edge $\mathfrak{e}_j$ is defined as:

$$m_{r\,\mathfrak{e}_j}(v_i) \triangleq \frac{m_{\mathfrak{e}_j}(v_i)}{\#_m \mathfrak{e}_j}.$$

We introduce the **normalised relative multiplicity** of a vertex $v_i \in \mathfrak{e}_j$ in the hb-graph as:

$$m_{nr\,\mathfrak{e}_j}(v_i) \triangleq \frac{m_{\mathfrak{e}_j}(v_i)}{\#_m \mathfrak{e}_j} \times \frac{\overline{m}(v_i)}{M}$$

where $\overline{m}(v_i) \triangleq \max_{\mathfrak{e}\in\mathfrak{E}}(m_{\mathfrak{e}}(v_i))$ and $M \triangleq \max_{v\in V}(\overline{m}(v))$. This provides an other extra-node representation where the edge thickness is proportional to the normalised relative multiplicity. This representation gives a direct view of the importance of each vertex contribution to a hb-edge compared to other hb-edges.

If the hb-graph has some multiplicity function with both negative and positive values, the former representations can be adapted by using different shapes for the edges linking the extra-node to the vertices of the hb-edges.

1.4.7. Incidence matrix of a hb-graph

A multiset is well defined by its universe and its multiplicity function. It can be represented by the vector representation of the multiset.

Hb-edges of a given hb-graph have the same universe. Let n and p be two positive integers and, $\mathfrak{H} = (V, \mathfrak{E})$ be a non-empty hb-graph with vertex set $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![p]\!]}$. We define the **incidence matrix** of the hb-graph $\mathfrak{H}$ as the matrix H :

$$H \triangleq [m_j(v_i)]_{\substack{i\in[\![n]\!]\\ j\in[\![p]\!]}}.$$

This incidence matrix is intensively used in [OLGMM18c] to formalize the diffusion by exchanges in hb-graphs.

Proposition 1.8. *Any non-negative matrix with real coefficients is the incident matrix of an hb-graph.*

Any non-negative matrix with integer coefficients is the incident matrix of a natural hb-graph.

1.5. Some (potential) applications

Potentially, all the applications that have been cited for hypergraphs in [Ouv20] can be refined using hb-graphs either general or natural; it includes applications involving collection of multisets, with multisets such as bags of words and bags of visual words or such as the ones cited in Appendix B.2.5. We will propose as application in Chapter 3 an e-adjacency tensor for general hb-graphs, deducing as a particular case the one of a general hypergraph. We will also develop a diffusion process on hb-graphs in Chapter 2, with applications in information and image retrieval. In Chapter D.1, we present an other application using hb-graphs, that refines the hypergraph framework given in Chapter 6. We present here other possible applications such as the prime decomposition representation, the problem of hb-graph transversals, and networks. These applications are not the only ones, as many other use multisets, or, for some of them, collection of multisets and, hence, are potential applications for hb-graphs.

1.5.1. Prime decomposition representation and elementary operations on hb-graphs

In [Cor04], a natural number network based on common divisors of two vertices is proposed to replace the search for real scale-free networks by the generation of a deterministic network that is also scale-free. Degrees of vertices are studied as well as the clustering coefficient and the average distance of vertices in the graph. Some topological properties of such networks are addressed in [ZWHC06]. In [FCS12], the authors study the page rank of an integer network built using a directed graph where the vertices are labeled by non-negative integers; an edge links two vertices m and n if m divides n, m being different of 1 and n, with a weight k corresponding to the maximal k such that m^k divides n, i.e. the valuation of m in n. All these approaches are built using graphs and pairwise relationships.

As already mentioned, multisets can be used for prime decomposition ([Bli88]). In particular, multisets are intensively used in [Tar14] to achieve primality decomposition of numbers and to achieve product, division, gcd and lcd of numbers. Using hb-graphs, we can revisit some of the results of [Tar14], and have a visual representation of simultaneous decomposition of numbers interpretable in term of elementary operations that transform a hb-graph representation into another one. It should also induce some refinement of results obtained with graphs since multisets handle not solely the multiadic relationships that could have been achieved using sets, but also the hb-edge based weighting of the divisors.

We focus on the prime decomposition of numbers. Considering the set $\mathbb{P}$ of prime numbers, any positive integer n greater or equal to 2 can be decomposed into a product of prime numbers: p_{i_1} with multiplicity $m_n(i_1)$ to p_{i_n} with multiplicity $m_n(i_n)$. This decomposition is then uniquely described by the multiset: $\mathfrak{e}_n = \left\{p_{i_1}^{m_{i_1}}, ..., p_{i_n}^{m_{i_n}}\right\}$. The prime decomposition hb-graph $\mathfrak{H}_{\mathbb{P}} \triangleq (\mathbb{P}, \mathfrak{E})$, is the hb-graph of universe $\mathbb{P}$ and of hb-edges $\mathfrak{E} \triangleq (\mathfrak{e}_n)_{n\in\mathbb{N}\setminus\{0,1\}}$ that correspond to the prime decomposition of the integers greater than 2. It contains all the possible natural multisets composed of elements of $\mathbb{P}$. To represent this hb-graph, each extra-node is labeled by the corresponding number n which decomposition in prime numbers constitutes the multiset $\mathfrak{e}_n = \left\{p_{i_1}^{m_{i_1}}, ..., p_{i_n}^{m_{i_n}}\right\}$.

We consider a subset A of the hb-edges of $\mathfrak{H}_{\mathbb{P}}$ and write $\mathfrak{H}_A \triangleq (\mathbb{P}, A)$ the sub-hb-graph of $\mathfrak{H}_{\mathbb{P}}$ associated to A. We also consider elementary operations on the multipartite extra-node representation of the natural hb-graphs constructed to switch between the decomposition involved by the two integers and the prime decomposition of the results. We observed that these elementary operations are similar to the elementary operations involved in the graph edit distance [GXTL10], albeit the fact that, with extra-nodes, some supplementary operations are possible: deletion of an edge, relabeling of an extra-node, deletion of an extra-node, merging of two extra-nodes.

The decomposition in primes of the product mn of two integers m and n is represented by a hb-edge $\mathfrak{e}_{mn}$ of $\mathfrak{H}_{\mathbb{P}}$ which is such that $\mathfrak{e}_{mn} = \mathfrak{e}_m \uplus \mathfrak{e}_n$. The multi-partite extra-node representation of $\mathfrak{H}_{\{\mathfrak{e}_{mn}\}}$ is obtained from the one of $\mathfrak{H}_{\{\mathfrak{e}_m,\mathfrak{e}_n\}}$ by merging the two extra-nodes n and m of the $\mathfrak{e}_n$ and $\mathfrak{e}_m$ representations, while keeping all the existing edges of their respective representation. This is illustrated on an example in Figure 1.2.

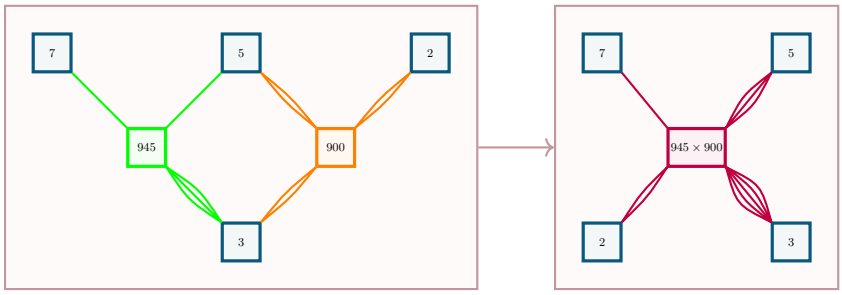


Figure 1.2.: Finding the prime decomposition of mn from the decomposition of m and n with $m = 900$ and $n = 945$.

If we suppose that m divides n, then the decomposition of $n \div m$ in primes is stored in a hb-edge $\mathfrak{e}_{n\div m}$ such that $\mathfrak{e}_{n\div m} = \mathfrak{e}_n \backslash \mathfrak{e}_m$. The multipartite extra-node representation of $\mathfrak{H}_{\{\mathfrak{e}_{n\div m}\}}$ is obtained from the one of $\mathfrak{H}_{\{\mathfrak{e}_m,\mathfrak{e}_n\}}$ by deleting all edges in the representation of $\mathfrak{e}_m$ and in the same quantities the corresponding edges in the representation of $\mathfrak{e}_n$ and relabeling the extra-node of $\mathfrak{e}_n$ to be the one of $\mathfrak{e}_{n\div m}$.

The decomposition in primes of the greater common divider of two integers m and n is stored in $\mathfrak{e}_{\gcd(m,n)} = \mathfrak{e}_m \cap \mathfrak{e}_n$. The representation of $\mathfrak{H}_{\{\mathfrak{e}_{\gcd(m,n)}\}}$ is obtained from the one of $\mathfrak{H}_{\{\mathfrak{e}_m,\mathfrak{e}_n\}}$ by deleting any edge from one vertex in $\mathfrak{e}_m^*$ to the extra-node representing $\mathfrak{e}_m$ that is greater in quantity than the one linking this vertex to the extra-node representing $\mathfrak{e}_n$ and reciprocally. The final representation is obtained by deleting one of the remaining extra-node vertex and its connected edges and relabeling the other extra-vertex with $\gcd(m,n)$. It is illustrated in Figure 1.3.

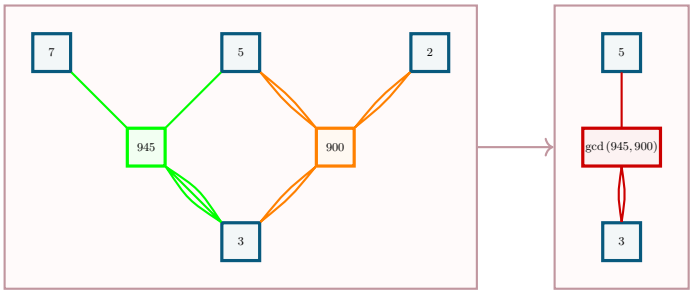


Figure 1.3.: Finding the prime decomposition of $\gcd(m,n)$ from the decomposition of m and n with $m = 900$ and $n = 945$.

The decomposition in primes of the least common multiple of two integers m and n is stored in $\mathfrak{e}_{\mathrm{lcm}(m,n)} = \mathfrak{e}_m \cup \mathfrak{e}_n$. The representation of $\mathfrak{H}_{\{\mathfrak{e}_{\mathrm{lcm}(m,n)}\}}$ is obtained from the one of $\mathfrak{H}_{\{\mathfrak{e}_m,\mathfrak{e}_n\}}$ by deleting any edge from one vertex in $\mathfrak{e}_m^*$ to the extra-node representing $\mathfrak{e}_m$ that is greater in quantity than the one linking this vertex to the extra-node representing $\mathfrak{e}_n$ and reciprocally. The final representation is obtained by deleting one of the remaining extra-node vertex and its connected edges and relabeling the other extra-vertex with $\mathrm{lcm}(m,n)$. It is illustrated in Figure 1.4.

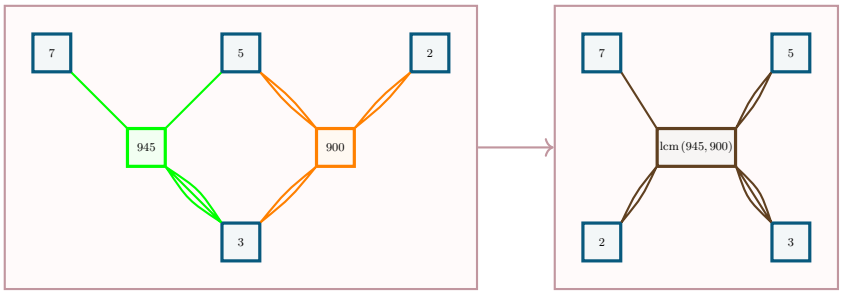


Figure 1.4.: Finding the prime decomposition of $\operatorname{lcm}(m,n)$ from the decomposition of m and n with $m = 900$ and $n = 945$.

We can then formulate the property:

$$mn = \gcd(m,n) \times \operatorname{lcm}(m,n)$$

by using the hb-graphs $\mathfrak{H}_{\{\mathfrak{e}_{mn}\}}$ and $\mathfrak{H}_{\{\mathfrak{e}_{\gcd(m,n)},\mathfrak{e}_{\operatorname{lcm}(m,n)}\}}$.

It holds: $\mathfrak{e}_{mn} = \mathfrak{e}_{\gcd(m,n)} \uplus \mathfrak{e}_{\operatorname{lcm}(m,n)}$, which can be written: $\mathfrak{e}_{mn} = (\mathfrak{e}_m \cap \mathfrak{e}_n) \uplus (\mathfrak{e}_m \cup \mathfrak{e}_n)$, which can be easily observed on the results shown in Figure 1.2 and Figure 1.5.

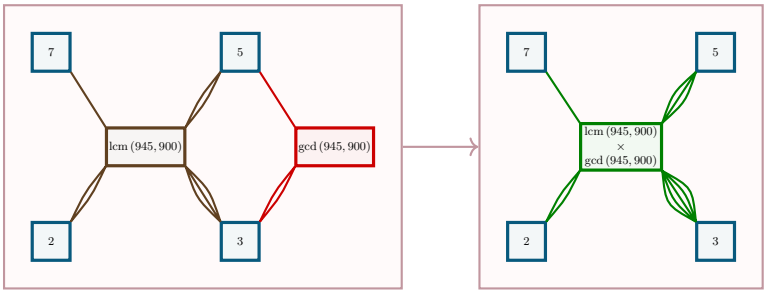


Figure 1.5.: Illustrating the property $\operatorname{lcm}(m,n) \times \gcd(m,n) = mn$ with $m = 900$ and $n = 945$.

It follows that any connected natural hb-graph can be attached to a number by labeling vertices with prime numbers and multiplying successively the hb-edges by this number: in some way it makes a summary of this hb-graph. Finding the prime labeling such that the number the natural hb-graph represents is minimal is a NP-hard problem.

Reciprocally, being given a number we can use its decomposition to create a collection of hb-graphs that have this number as overall representation.

1.5.2. Graph pebbling and families of multisets

In [BBCH03], the authors use families of multisets to enounce analogs of two theorems on graph pebbling for multisets and improve some bounds on the random pebbling threshold for sequence of paths in graphs. The authors consider a pebbling problem on graphs in the following variant: from an initial pebble distribution on graph vertices,

they consider a vertex with two pebbles and move one of the pebbles to an adjacent vertex, with the aim of repeating the operation until only one pebble is remaining on a predetermined root vertex. All the configurations are not always solvable depending on the number of pebbles distributed.

A classical variant of the pebble problem consists in moving labeled pebbles on a graph from a given configuration to an other one with a constraint on the maximal number of pebbles that can be found on a vertex at any time. Pebble motion problems have applications in multi-robot motion planning and network routing. Famous pebble motion problems are sliding puzzles like the 15-puzzle. An other variation of a pebble problem has been used in [ST09] to characterize the sparsity of hypergraphs. In [BS17], the authors use the former work to extract rigidity and flexibility information of bio-molecules using a phased pebble-game on a multi-graph that reflects the mechanical model of the atom-bond network.

One can, therefore, consider another variation of these problems solving some natural hb-graph pebbling problems, such that pebbles occupy vertices of natural hb-graphs with allocated resources, that depends on the hb-edges the pebbles are coming from. This game would take into account the constraints that are depending on the hb-edge itself to allocate resources on one of its vertex.

1.5.3. Chemical reaction networks and hb-graphs

As mentioned in [CTTV15], chemical reaction network (CRN for short) theory is at the confluence of computer science and the study of interactions in natural sciences and chemistry; it is also used as model for natural programming. CRN theory focuses on the kinetic of the various components involved in the reactions. The authors in [Gun03] use sets of multisets to model complexes of species; chemical reactions occur on complexes transforming a complex into another complex. Each chemical reaction is associated to a rate. The complexes of the chemical reaction network constitute a hb-graph of universe the entities and where the hb-edges represent the complexes. The reactions can be seen as morphisms on this hb-graph, that are pondered with a rate. In [CTTV15], the authors focus on the kinetic of such networks with semantics based on ordinary differential equations. In [LSP16], the states of the CRN are viewed as multisets of species. Hence the reactions reflect the possibility to go from one state to another. Succession of reactions are called traces and valid traces are the ones that are possible, i.e. with sufficient reactant at each step.

A reaction can be seen as a replacement rule in the current multiset that is possible only if all its reactants are present in sufficient quantity in the hb-edge. Using directed hb-graphs over a universe, with a source and a target in each hb-edge, the reactions are viewed as directed hb-edges such that source and target are not overlapping multisets. A trace is a transversal of the hb-graph of states by directed hb-edges such that each source of the reaction is a submset of the current state and the target state is a submset of the new state, where the new state is written using operation on multisets: $S_f = (S_i \backslash R) \uplus P$, where $R|P$ is the directed hb-edge of the reaction, S_f represents the final state and S_i the initial state.

1.5.4. Multiset cover problems and hb-graph transversals

Set cover problems consist in covering a given set U with a set of subsets which union corresponds to the initial set U, using a minimum amount of those subsets: it is an NP-complete problem listed in Karp's NP complete problems in [Kar72]. Different flavors of this problem exist: some of them impose a covering of maximal size, some give a cost to each subset or impose a maximal number of repetitions, imposing to each subset to have a maximal covering capacity that should not be exceeded.

In [RV93], the authors study the multiset multicover problem, similar to the set cover problem, replacing subsets by sub-multisets to cover a multiset of N elements. They see the multisets as copy-sets and adapt the parallel algorithm given in the case of sets to give an approximation of the solution of the problem using a greedy multiset cover algorithm. In [HYLW09], the authors describe a multiset multicover problem and give an exact algorithm to solve it in both cases i.e. without and with multiplicity constraints on the number of times the submset can be picked up. Multisets are viewed as copy-sets and transformed as such.

The problem of hypergraph vertex covering, or known also as the problem of hypergraph transversal is a well studied NP-hard problem—see for instance [EG95] that gives also applications in database and model-based diagnosis. It consists in finding a minimal set of vertices that intersects each hyperedge. This is equivalent to the set cover problem as it is shown in [Hal02] by considering the dual hypergraph: the set to cover is constituted in this case of the—non-repeated—hyperedges and the subsets that are at the basis of the covering corresponds to the dual hypergraph hyperedges, which contain the incident original hypergraph hyperedges of each vertex. In [SK12], the authors propose an approximation algorithm to handle hypergraph cover with hard capacities, which is seen as analogous to consider set systems with constrained belongings. Hypergraph minimal transversal is fruitful in data mining. A recent application occurs in the frequent item-set hiding problem in [SVK16] and security in [RKS17].

Hence, for hb-graphs the problem of vertex covering can be seen as an extension of the hypergraph transversal problem to a minimal multiset of vertices that intersect each hb-edge and is similarly equivalent to the multiset cover problem by considering the dual hb-graph: the multiset to cover is the one of the hb-edges and the family of multisets used for the covering corresponds to the dual hb-edges, attached to each vertex of the original incident hb-edges. This kind of problems can be linked to constrained optimization problems of fuzzy relational equations [HF11] and relates to set cover problems as it is mentioned in [CW02] and in [Mar05].

1.5.5. Networks

1.5.5.1. Computer networks

Byzantine-fault problems often consider some multisets for representing the information collected by one agent in the network as in [MHVG15]. In [Geo17], multisets are used to model beliefs of agents and also desirable believes.

Hb-graphs can definitively model this kind of interaction between agents, even giving a way to encode additional information that sets cannot capture.

1.5.5.2. Neural networks

In [XHLJ18], the authors use multisets in graph neural networks, to focus on the neighborhood. In graph neural networks, the aim is to learn the graph through the neural network by a recursive neighborhood aggregation, where the information of the neighborhood of the node contained in different vectors is aggregated to create its own feature vector. The authors propose to model this neighborhood using multisets combined with some convenient function.

Hypergraph neural networks that encode higher order relationships are proposed in [FYZ+19] and are shown to overtake classical methods.

Learning hb-graph through neural networks can, therefore, open up a wide range of applications, since it could also capture the multiplicity of elements.

1.6. Further comments on hb-graphs

Shifting to hb-graphs is not only a change in modeling for visualisation: it is also a means to refine any network in which hb-edge-based individual weighting of vertices is required. Natural hb-graphs constitute a new category, as it is shown in Appendix C.2.

We have already shown in [OLGMM18c] that diffusion by exchange in hb-graphs—as it will be presented in Chapter 2—provides a fine ranking not only of vertices but also of hb-edges. We think that extending the definitions of hb-graphs to support negative multiplicities or even complex values could open the door to yet a wider variety of applications.

We can foresee many applications of hb-graphs, as they give the best of several worlds: multisets, sets, graphs, and, also, as we will see in the Chapter 3, algebraic and polynomial approaches. Natural hb-graphs support the duplication of elements, while general hb-graphs allow the weighting of elements in a refined manner with respect to hypergraphs.

2. Diffusion in hb-graphs (matrix approach)

Highlights

This chapter is based on [OLGMM19a], which is an extension of [OLGMM18c].

Prerequisites: Section 1.3.

2.1. Motivation

Highlighting important information in networks is commonly achieved using random walks. In 1905, Pearson was using the idea of random walk to model a population of mosquitoes in a forest; previously, Rayleigh studied acoustic waves using random walks. The term itself was introduced by Polya during the 1920s. A random walk is a stochastic process related to a Markov chain. A random walk is a succession of steps, of eventually constant length, executed in a random manner regarding the direction and the length of the walk. Constrained to a network structure, the random walker is supposed to randomly move from a vertex to a neighborhood vertex.

Random walks, by the phenomena involved, are tied up to diffusion over the structure, and have been largely used to retrieve centrality of information in graphs. Random walks have had a revival of interest since the raise of Internet. They are at the origin of the PageRank algorithm developed by Google to rank websites [PBMW99].

Random walks on hypergraphs and diffusion have also been well studied since [ZHS07]; in [BAD13], the authors show that the weighting of vertices at the level of the hyperedges in a hypergraph provides better information retrieval. These two approaches—[ZHS07, BAD13]—mainly focus on vertices. But, more generally, as hb-edges of a hb-graph are linked to references that can be used as pivots in between the different facets as shown in Chapter 6, it is also interesting to highlight important hb-edges. Hence, the following research question:

Research question 2.1. *Can we find a diffusion process that not only ranks vertices but also ranks hb-edges in hb-graphs?*

We start by reviewing existing work, and propose a model of diffusion on the hb-graph, that is exchange-based: this process requires two-phase steps. We then evaluate this framework with generated data and apply it to two real data frameworks, one based on queries over Arxiv database.

2.2. Related work

Random walks are largely used to evaluate the importance of vertices in networks, either seen as graphs or as hypergraphs. In [Lo93], a survey is achieved on random walks on graphs; it includes connections with eigenvalues of the adjacency matrix of graphs as well as estimates on important features such as access time and cover time. Random walks are closely related to diffusion processes.

Diffusion kernels have been introduced in [KL02]: they are shown to be efficient on different kinds of graphs. In [BN03], the authors build a neighboring weighted graph using a truncated heat kernel to weight the proximity of two vertices, and achieve nonlinear dimensionality reduction preserving the local properties. In [MYLK08], the authors use a heat diffusion process for marketing candidates selection in social networks, and, in [MKL11], for mining web graphs for recommendations. In [TDKF17], the authors uses a sum of heat diffusion processes to learn about the hidden graph structure. The interested reader can refer to the extensive survey on random walks and diffusion over networks done in [MPL17].

In [ZHS07], a random walk on a hypergraph is defined by choosing a hyperedge e_j with a probability proportional to $w_e(e_j)$, and within that hyperedge a vertex is randomly chosen using a uniform law. The probability transition from a vertex v_{i_1} to a vertex v_{i_2} is:

$$p(v_{i_1}, v_{i_2}) = \sum_{j \in [\![p]\!]} w_e(e_j) \frac{h_{i_1 j}}{d_{i_1}} \times \frac{h_{i_2 j}}{\delta_j},$$

where $\delta_j = \deg(e_j)$, $j \in [\![p]\!]$ is the degree of a hyperedge defined in [ZHS07] as its cardinality. This random walk has a stationary state which is shown to be $\pi = (\pi_i)_{i \in [\![n]\!]}$ with $\pi_i = \dfrac{d_i}{\mathrm{vol}V}$ for $i \in [\![n]\!]$—[DB14]. This process differs from the one we propose: our diffusion process is done in successive steps on vertices and hyperedges from a random initial vertex.

In [BAD13], the authors define a random walk for weighted hypergraphs using weight functions both for hyperedges and vertices: for each vertex, a vector of weights is built making weights of vertices hyperedge-based; a random walk similar to the one above is then built that takes into account the vertex weight. The evaluation is performed on a hypergraph built from a public dataset of computer science conference proceedings; each document is seen as a hyperedge that contains keywords; hyperedges are weighted by citation score and vertices of a hyperedge are weighted with a tf-idf score. In [BAD13], the authors show that a random walk on the (double-) weighted hypergraph enables vertex ranking with higher precision than random walks using unweighted vertices.

This process differs also from our proposal: our process not only enables simultaneous alternative updates of vertices and hb-edges values but also provides hb-edge ranking. We also introduce a new theoretical framework to perform our diffusion process.

Random walks relate to diffusion processes. Diffusion processes are extensively used for information retrieval: [DB13] is an extensive survey on the subject. More particularly, random walks are used with hypergraphs for image matching—[LCL11]. Higher order random walks in hypergraph are built in [LP11], where a generalised Laplacian attached to the graphs generated from these random walks is constructed. In [GMM$^+$13], the authors propose a co-occurrence based diffusion for expert search on the web: the model uses the heat diffusion over a heterogeneous co-occurrence network—composed of co-occurrences of persons and keywords—that is reduced to its 2-section when enabling the diffusion, considering hyperedge-based weighting of vertices, i.e. a hb-graph structure. They achieve good performances on retrieval of the experts on ground truth datasets. In [MSCM16], the authors propose a multi-scale hypergraph framework for image annotation, that exploits the eigenvalues of the hypergraph Laplacian matrix in the definition of the heat kernel. This latter includes a parameter to tune the scale of propagation of the labels in order to solve the problem of class imbalance in the data.

2.3. Exchange-based diffusion in hb-graphs

Diffusion processes result in homogenising information over a structure; an initial stroke is done on a vertex that propagates over the network structure. This propagation is often modeled by a random walk on the network. Random walks in hypergraphs rank vertices by the number of times they are reached, and this ranking is related to the network structure itself. Several random walks with random choices for the starting vertex are needed to achieve a ranking by averaging. Moreover, to avoid loops, teleportation of vertices is needed.

We consider a weighted hb-graph $\mathfrak{H} = (V, \mathfrak{E}, w_e)$ with $|V| = n$ and $|\mathfrak{E}| = p$; we write H the incidence matrix of the hb-graph.

At time t, we set a distribution of values over the vertex set:

$$\alpha_t : \begin{cases} V \to \mathbb{R} \\ v_i \mapsto \alpha_t(v_i) \end{cases}.$$

and a distribution of values over the hb-edge set:

$$\epsilon_t : \begin{cases} \mathfrak{E} \to \mathbb{R} \\ \mathfrak{e}_j \mapsto \epsilon_t(\mathfrak{e}_j) \end{cases}.$$

We write $P_{V,t} = (\alpha_t(v_i))_{i \in [\![n]\!]}$ the row state vector of the vertices at time t and $P_{\mathfrak{E},t} = (\epsilon_t(\mathfrak{e}_j))_{j \in [\![p]\!]}$ the row state vector of the hb-edges.

The initialisation is done such that $\sum_{v_i \in V} \alpha_0(v_i) = 1$ and the information value is concentrated uniformly on the vertices at the beginning of the diffusion process and, consequently, each hb-edge has a zero value associated to it. Writing $\alpha_{\text{ref}} = \dfrac{1}{|V|}$, we set for all $v_i \in V : \alpha_0(v_i) = \alpha_{\text{ref}}$ and for all $\mathfrak{e}_j \in \mathfrak{E}$, $\epsilon(\mathfrak{e}_j) = 0$.

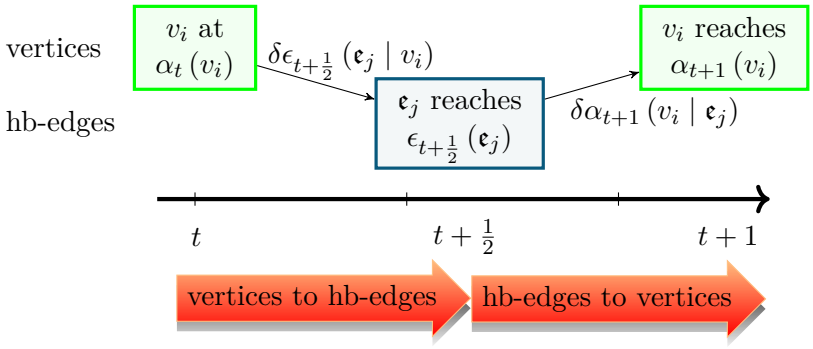


Figure 2.1.: Diffusion by exchange: principle, with:
$\delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \triangleq \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \alpha_t(v_i)$ and $\delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \triangleq \frac{m_j(v_i)}{\#_m \mathfrak{e}_j} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j)$.

We consider an iterative process with two-phase steps. At every step, the first phase starts at time t and ends at $t+\frac{1}{2}$, followed by the second phase between time $t+\frac{1}{2}$ and $t+1$. This iterative process is illustrated in Figure 2.1 that preserves the overall value held by the vertices and the hb-edges, meaning that we have at any $t \in \left\{\frac{1}{2}k : k \in \mathbb{N}\right\}$:

$$\sum_{v_i \in V} \alpha_t(v_i) + \sum_{\mathfrak{e}_j \in \mathfrak{E}} \epsilon_t(\mathfrak{e}_j) = 1.$$

During the first phase between time t and $t+\frac{1}{2}$, each vertex v_i of the hb-graph shares its value $\alpha_t(v_i)$ hold at time t with the hb-edges it is connected to.

In an unweighted hb-graph, the fraction of $\alpha_t(v_i)$ given by v_i of m-degree $d_{v_i} = \deg_m(v_i)$ to each hb-edge is $\frac{m_j(v_i)}{\deg_m(v_i)}$, which corresponds to the ratio of the multiplicity of the vertex v_i due to the hb-edge $\mathfrak{e}_j$ over the total m-degree of hb-edges containing v_i in their support.

In a weighted hb-graph, each hb-edge has a weight $w_e(\mathfrak{e}_j)$. The value $\alpha_t(v_i)$ of vertex v_i is shared by accounting not only the multiplicity of the vertices in the hb-edge but also the weight $w_e(\mathfrak{e}_j)$ of a hb-edge $\mathfrak{e}_j$.

The weights of the hb-edges are stored in a column vector:

$$w_{\mathfrak{E}} = (w_e(\mathfrak{e}_j))^{\top}_{j \in [\![p]\!]}.$$

We also consider the weight diagonal matrix:

$$W_{\mathfrak{E}} = \operatorname{diag}\left((w_e(\mathfrak{e}_j))_{j \in [\![p]\!]}\right).$$

We introduce the weighted m-degree matrix:

$$D_{w,V} = \operatorname{diag}\left((d_{w,v_i})_{i \in [\![n]\!]}\right) = \operatorname{diag}(H w_{\mathfrak{E}}).$$

where d_{w,v_i} is called the weighted m-degree of the vertex v_i. It is:

$$d_{w,v_i} = \deg_{w,m}(v_i) = \sum_{j \in [\![p]\!]} m_j(v_i)\, w_e(\mathfrak{e}_j).$$

The contribution of vertex v_i to the value $\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j)$ attached to hb-edge $\mathfrak{e}_j$ of weight $w_e(\mathfrak{e}_j)$ is:

$$\delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \triangleq \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \alpha_t(v_i).$$

It corresponds to the ratio of weighted multiplicity of vertex v_i in $\mathfrak{e}_j$ over the total weighted m-degree of the hb-edges where v_i is in the support.

We remark that if $v_i \notin \mathfrak{e}_j^\star$:

$$\delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) = 0.$$

And the value $\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j)$ is calculated by summing over the vertex set:

$$\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) = \sum_{i \in [\![n]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i).$$

Hence, we obtain:

$$P_{\mathfrak{E},t+\frac{1}{2}} = P_{V,t} D_{w,V}^{-1} H W_{\mathfrak{E}}. \tag{2.1}$$

The value given to the hb-edges is subtracted to the value of the corresponding vertex, hence for all $i \in [\![n]\!]$:

$$\alpha_{t+\frac{1}{2}}(v_i) = \alpha_t(v_i) - \sum_{j \in [\![p]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i).$$

Claim 2.1 (No information on vertices at $t+\frac{1}{2}$)**.** *It holds:*

$$\forall i \in [\![n]\!] : \alpha_{t+\frac{1}{2}}(v_i) = 0.$$

Proof. For all $i \in [\![n]\!]$:

$$\begin{aligned}
\alpha_{t+\frac{1}{2}}(v_i) &= \alpha_t(v_i) - \sum_{j \in [\![p]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \\
&= \alpha_t(v_i) - \sum_{j \in [\![p]\!]} \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \alpha_t(v_i) \\
&= \alpha_t(v_i) - \alpha_t(v_i) \frac{\sum_{j \in [\![p]\!]} m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \\
&= 0.
\end{aligned}$$

□

Claim 2.2 (Conservation of the information of the hb-graph at $t+\frac{1}{2}$)**.** *It holds:*

$$\sum_{v_i \in V} \alpha_{t+\frac{1}{2}}(v_i) + \sum_{\mathfrak{e} \in \mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}) = 1.$$

Proof. We have:

$$\begin{aligned}
\sum_{v_i \in V} \alpha_{t+\frac{1}{2}}(v_i) + \sum_{\mathfrak{e} \in \mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}) &= \sum_{\mathfrak{e}_j \in \mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \sum_{\mathfrak{e}_j \in \mathfrak{E}} \sum_{i \in [\![n]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \\
&= \sum_{\mathfrak{e}_j \in \mathfrak{E}} \sum_{i \in [\![n]\!]} \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \alpha_t(v_i) \\
&= \sum_{i \in [\![n]\!]} \alpha_t(v_i) \frac{\sum\limits_{\mathfrak{e}_j \in \mathfrak{E}} m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \\
&= \sum_{i \in [\![n]\!]} \alpha_t(v_i) \\
&= 1.
\end{aligned}$$

□

During the second phase which starts at time $t + \frac{1}{2}$, the hb-edges share their values across the vertices they hold taking into account the vertex multiplicities in the hb-edge.

The contribution to $\alpha_{t+1}(v_i)$ given by a hb-edge $\mathfrak{e}_j$ is proportional to $\epsilon_{t+\frac{1}{2}}$ in a factor corresponding to the ratio of the multiplicity $m_j(v_i)$ of the vertex v_i to the hb-edge m-cardinality:

$$\delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \triangleq \frac{m_j(v_i)}{\#_m \mathfrak{e}_j} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j).$$

The value $\alpha_{t+1}(v_i)$ is then obtained by summing on all values associated to the hb-edges that are incident to v_i :

$$\alpha_{t+1}(v_i) = \sum_{j \in [\![p]\!]} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j).$$

Writing $D_{\mathfrak{E}} = \operatorname{diag}\left((\#_m \mathfrak{e}_j)_{j \in [\![p]\!]}\right)$ the diagonal matrix of size $p \times p$, it comes:

$$P_{\mathfrak{E},t+\frac{1}{2}} D_{\mathfrak{E}}^{-1} H^{\top} = P_{V,t+1}. \tag{2.2}$$

The values given to the vertices are subtracted to the value associated to the corresponding hb-edge. Hence, for all $j \in [\![p]\!]$:

$$\epsilon_{t+1}(\mathfrak{e}_j) = \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i \in [\![n]\!]} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j).$$

Claim 2.3 (The hb-edges have 0 value at $t+1$)**.** *It holds:*

$$\epsilon_{t+1}(\mathfrak{e}_j) = 0.$$

Proof. For all $i \in [\![p]\!]$:

$$\begin{aligned}
\epsilon_{t+1}(\mathfrak{e}_j) &= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i\in[\![n]\!]} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \\
&= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i\in[\![n]\!]} \frac{m_j(v_i)}{\#_m \mathfrak{e}_j} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \left(1 - \frac{\sum\limits_{i\in[\![n]\!]} m_j(v_i)}{\#_m \mathfrak{e}_j}\right) \\
&= 0.
\end{aligned}$$

□

Claim 2.4 (Conservation of the information of the hb-graph at $t+1$)**.** *It holds:*

$$\sum_{v_i\in V} \alpha_{t+1}(v_i) + \sum_{\mathfrak{e}_j\in\mathfrak{E}} \epsilon_{t+1}(\mathfrak{e}_j) = 1.$$

Proof.

$$\begin{aligned}
\sum_{v_i\in V} \alpha_{t+1}(v_i) + \sum_{\mathfrak{e}\in\mathfrak{E}} \epsilon_{t+1}(\mathfrak{e}) &= \sum_{v_i\in V} \alpha_{t+1}(v_i) \\
&= \sum_{v_i\in V} \sum_{j\in[\![p]\!]} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \\
&= \sum_{v_i\in V} \sum_{j\in[\![p]\!]} \frac{m_j(v_i)}{\#_m \mathfrak{e}_j} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \sum_{j\in[\![p]\!]} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \frac{\sum\limits_{v_i\in V} m_j(v_i)}{\#_m \mathfrak{e}_j} \\
&= \sum_{j\in[\![p]\!]} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= 1.
\end{aligned}$$

□

Regrouping (2.1) and (2.2):

$$P_{V,t+1} = P_{V,t} D_{w,V}^{-1} H W_{\mathfrak{E}} D_{\mathfrak{E}}^{-1} H^\top. \tag{2.3}$$

It is valuable to keep a trace of the intermediate state $P_{\mathfrak{E},t+\frac{1}{2}} = P_{V,t} D_{w,V}^{-1} H W_{\mathfrak{E}}$ as it records the importance of the hb-edges.

Writing $T = D_{w,V}^{-1} H W_{\mathfrak{E}} D_{\mathfrak{E}}^{-1} H^\top$, it follows from 2.3

$$P_{V,t+1} = P_{V,t} T. \tag{2.4}$$

Claim 2.5 (Stochastic transition matrix)**.** *T is a square row stochastic matrix of dimension n.*

Proof. Let consider: $A = (a_{ij})_{\substack{i\in[\![n]\!]\\ j\in[\![p]\!]}} = D_{w,V}^{-1} H W_{\mathfrak{E}} \in M_{n,p}$ and $B = (b_{jk})_{\substack{j\in[\![p]\!]\\ k\in[\![n]\!]}} = D_{\mathfrak{E}}^{-1} H^{\top} \in M_{p,n}$.

A and B are non-negative rectangular matrices. Moreover:

$$a_{ij} = \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}}$$

and, it holds:

$$\sum_{j\in[\![p]\!]} a_{ij} = \frac{\sum\limits_{j\in[\![p]\!]} m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} = 1.$$

$b_{jk} = \dfrac{m_j(v_k)}{\#_m(\mathfrak{e}_j)}$ and it holds:

$$\sum_{k\in[\![n]\!]} b_{jk} = \frac{\sum\limits_{k\in[\![n]\!]} m_j(v_k)}{\#_m \mathfrak{e}_j} = 1.$$

We have $P_{V,t+1} = P_{V,t} AB$ where $AB = \left(\sum\limits_{j\in[\![p]\!]} a_{ij} b_{jk} \right)_{\substack{i\in[\![n]\!]\\ k\in[\![n]\!]}}$.

It yields:

$$\begin{aligned} \sum_{k\in[\![n]\!]} \sum_{j\in[\![p]\!]} a_{ij} b_{jk} &= \sum_{j\in[\![p]\!]} a_{ij} \sum_{k\in[\![n]\!]} b_{jk} \\ &= \sum_{j\in[\![p]\!]} a_{ij} \\ &= 1. \end{aligned}$$

Hence AB is a non-negative square matrix with its row sums all equal to 1: it is a row stochastic matrix. □

Claim 2.6 (Properties of T)**.** *Supposing that the hb-graph is connected, the exchange-based diffusion matrix T is aperiodic and irreducible.*

Proof. This stochastic matrix is aperiodic, due to the fact that any vertex of the hb-graph retrieves a part of the value it has given to the hb-edge, hence $t_{ii} > 0$ for all $i \in [\![n]\!]$.

Moreover, as the hb-graph is connected, the matrix is irreducible as every state can be joined from any other state. □

Claim 2.7. *The sequence $(P_{V,t})_{t\in\mathbb{N}}$, with $P_{V,t} = (\alpha_t(v_i))_{i\in[\![n]\!]}$, in a connected hb-graph converges to the state vector π_V such that:*

$$\pi_V = \left(\frac{d_{w,v_i}}{\sum\limits_{k\in[\![n]\!]} d_{w,v_k}} \right)_{i\in[\![n]\!]}.$$

Proof. We denote by π an eigenvector of $T = (c_{ik})_{\substack{i\in[\![n]\!]\\k\in[\![n]\!]}}$ associated to the eigenvalue 1. We have $\pi T = \pi$.

Let consider $u = (d_{w,v_i})_{i\in[\![n]\!]}$.

We have:

$$\begin{aligned}
(uT)_k &= \sum_{i\in[\![n]\!]} d_{w,v_i} c_{ik} \\
&= \sum_{i\in[\![n]\!]} d_{w,v_i} \sum_{j\in[\![p]\!]} \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,v_i}} \times \frac{m_j(v_k)}{\#_m(\mathfrak{e}_j)} \\
&= \sum_{j\in[\![p]\!]} \sum_{i\in[\![n]\!]} m_j(v_i)\, w_e(\mathfrak{e}_j) \times \frac{m_j(v_k)}{\#_m(\mathfrak{e}_j)} \\
&= \sum_{j\in[\![p]\!]} w_e(\mathfrak{e}_j)\, m_j(v_k)\, \frac{\sum\limits_{i\in[\![n]\!]} m_j(v_i)}{\#_m(\mathfrak{e}_j)} \\
&= \sum_{j\in[\![p]\!]} w_e(\mathfrak{e}_j)\, m_j(v_k) \\
&= d_{w,v_k} = u_k.
\end{aligned}$$

Hence, u is a non-negative eigenvector of T associated to the eigenvalue 1.

For a connected hb-graph, when we iterate over the stochastic matrix T which is aperiodic and irreducible, we are then ensured to converge to a stationary state: this stationary state is the probability vector associated to the eigenvalue 1. It is unique and is equal to αu such that $\sum\limits_{k\in[\![n]\!]} \alpha u_k = 1$.

We have $\alpha = \dfrac{1}{\sum\limits_{k\in[\![n]\!]} d_{w,v_k}}$ and hence the result.

□

Claim 2.8. *The sequence* $\left(P_{\mathfrak{E},t+\frac{1}{2}}\right)_{t\in\mathbb{N}}$, *with* $P_{\mathfrak{E},t+\frac{1}{2}} = \left(\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j)\right)_{j\in[\![p]\!]}$, *in a connected hb-graph converges to the state vector* $\pi_{\mathfrak{E}}$ *such that:*

$$\left(\frac{w_e(\mathfrak{e}_j)\times\#_m(\mathfrak{e}_j)}{\sum\limits_{k\in[\![n]\!]} d_{w,v_k}}\right)_{j\in[\![p]\!]}.$$

Proof. As $P_{\mathfrak{E},t+\frac{1}{2}} = P_{V,t} D_{w,V}^{-1} H W_{\mathfrak{E}}$ and that $\lim\limits_{t\to+\infty} P_{V,t} = \pi_V$, the sequence $\left(P_{\mathfrak{E},t+\frac{1}{2}}\right)_{t\in\mathbb{N}}$ converges towards a state vector $\pi_{\mathfrak{E}}$ such that: $\pi_{\mathfrak{E}} = \pi_V D_{w,V}^{-1} H W_{\mathfrak{E}}$.

We have:

$$\begin{aligned}
\pi_{\mathfrak{E}} &= \left(\sum_{i \in [\![n]\!]} \frac{d_{w,v_i}}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \times \frac{m_j(v_i) \times w_e(\mathfrak{e}_j)}{d_{w,v_i}} \right)_{j \in [\![p]\!]} \\
&= \left(\sum_{i \in [\![n]\!]} \frac{m_j(v_i) \times w_e(\mathfrak{e}_j)}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \right)_{j \in [\![p]\!]} \\
&= \left(\frac{w_e(\mathfrak{e}_j) \times \sum\limits_{i \in [\![n]\!]} m_j(v_i)}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \right)_{j \in [\![p]\!]} \\
&= \left(\frac{w_e(\mathfrak{e}_j) \times \#_m(\mathfrak{e}_j)}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \right)_{j \in [\![p]\!]}.
\end{aligned}$$

All components are non-negative and we check that the components of this vector sum to one:

$$\begin{aligned}
\sum_{j \in [\![p]\!]} \pi_{\mathfrak{E},j} &= \frac{\sum\limits_{j \in [\![p]\!]} w_e(\mathfrak{e}_j) \times \sum\limits_{i \in [\![n]\!]} m_j(v_i)}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \\
&= \frac{\sum\limits_{i \in [\![n]\!]} \sum\limits_{j \in [\![p]\!]} w_e(\mathfrak{e}_j) \times m_j(v_i)}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \\
&= \frac{\sum\limits_{i \in [\![n]\!]} d_{w,v_i}}{\sum\limits_{k \in [\![n]\!]} d_{w,v_k}} \\
&= 1.
\end{aligned}$$

□

These two claims show that this exchange-based process ranks vertices by their weighted m-degree and hb-edges by their weighted m-cardinality.

We have gathered the two-phase steps of the exchange-based diffusion process in Algorithm 2.1. The time complexity of this algorithm is $O(T(d_{\mathfrak{H}} n + r_{\mathfrak{H}} p))$ where $d_{\mathfrak{H}} = \max\limits_{v_i \in V}(d_i)$ is the maximal degree of vertices in the hb-graph and $r_{\mathfrak{H}} = \max\limits_{\mathfrak{e}_j \in \mathfrak{E}} \left| \mathfrak{e}_j^\star \right|$ is the maximal cardinality of the support of a hb-graph. Usually, $d_{\mathfrak{H}}$ and $r_{\mathfrak{H}}$ are small compared to n and p. Algorithm 2.1 can be refined to determine automatically the number of iterations needed, fixing an accepted error to ensure convergence on the values of the vertices and storing the previous state.

Algorithm 2.1 Exchange-based diffusion algorithm.

Given:

A hb-graph $\mathfrak{H} = (V, \mathfrak{E}, w_e)$ with $|V| = n$ and $|\mathfrak{E}| = p$

Number of iterations: T

Initialisation:

For all $v_i \in V : \alpha_i := \dfrac{1}{n}$

For all $\mathfrak{e}_j \in \mathfrak{E} : \epsilon_j := 0$

DiffuseFromVerticesToHbEdges():

For $j := 1$ to p:

$\epsilon_j := 0$

For $v_i \in \mathfrak{e}_j^\star$:

$$\epsilon_j := \epsilon_j + \frac{m_j(v_i)\, w_e(\mathfrak{e}_j)}{d_{w,m}(v_i)} \alpha_i$$

DiffuseFromHbEdgesToVertices():

For $i := 1$ to n:

$\alpha_i := 0$

For $\mathfrak{e}_j$ such that $v_i \in \mathfrak{e}_j^\star$:

$$\alpha_i := \alpha_i + \frac{m_j(v_i)}{\#_m \mathfrak{e}_j} \epsilon_j$$

Main():

Calculate for all $i : d_{w,m}(v_i)$ and for all $j : \#_m \mathfrak{e}_j$

For $t = 1$ to T:

DiffuseFromVerticesToHbEdges()

DiffuseFromHbEdgesToVertices()

2.4. Results and evaluation

We validate the approach taken on random hb-graphs and then we apply it to process the results of Arxiv querying and on an image dataset.

2.4.1. Validation on random hb-graphs

This diffusion by exchange process has been validated on two experiments: the first experiment generates a random hb-graph to validate our approach and the second compares the results to a classical random walk on the hb-graph.

We built a random unweighted hb-graph generator. The generator makes it possible to construct a hb-graph with inter-connected sub-hb-graphs; those sub-hb-graphs can be potentially disconnected leading to multiple connected components. We restricted ourselves in the experiments to connected hb-graphs. A single connected component is built by choosing the number of intermediate vertices that link the different sub-hb-graphs together. As it is shown in Figure 2.2, we generate $N_{\max}$ vertices. We start by building each sub-hb-graph, called group, individually and then interconnect them. Let k be the number of groups. A first set V_0 of interconnected vertices is built by choosing

N_0 vertices out of the $N_{\max}$. The remaining $N_{\max} - N_0$ vertices are then separated into k subsets $(V_j)_{j \in [\![k]\!]}$. In each of these k groups V_j, we generate two subsets of vertices: a first set $V_{j,1}$ of $N_{j,1}$ vertices and, a second set $V_{j,2}$ of $N_{j,2}$ vertices with $N_{j,1} \ll N_{j,2}$, $j \in [\![k]\!]$. The number of hb-edges to be built is adjustable: their number is shared between the different groups. The m-cardinality $\#_m(\mathfrak{e})$ of a hb-edge is chosen randomly below a maximum tunable threshold. The $V_{j,1}$ vertices are considered as important vertices and must be present in a certain number of hb-edges per group; the number of important vertices in a hb-edge is randomly fixed below a maximum number. The completion of the hb-edge is done by choosing vertices randomly in the $V_{j,2}$ set. The random choice made into these two groups is tuned to follow a power law distribution. It implies that some vertices occur more often than others. Interconnection between the k components is achieved by choosing vertices in V_0, and inserting them randomly into the hb-edges built.

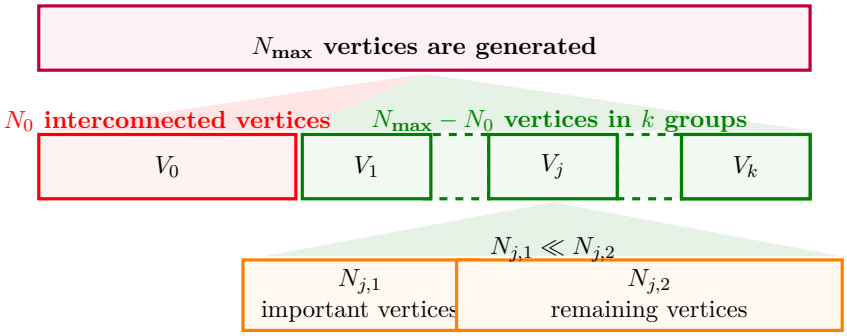


Figure 2.2.: Random hb-graph generation principle.

We apply the exchange-based diffusion process on this generated hb-graph: after a few iterations, we visualize the hb-graph to observe the evolution of the vertex values using a gradient coloring scale. We also take advantage of the first phase of each step to highlight hb-edges in the background, and show hb-edge importance using another gradient coloring scale.

To get proper evaluation and show that vertices with the highest α-values correspond to the important vertices of the network—in the sense of being central for the connectivity—, we compute the eccentricity of vertices from a subset S of the vertex set V to the remaining $V \backslash S$ of the vertices. The eccentricity of a vertex in a graph is the length of a maximal shortest path between this vertex and the other vertices of this graph: extending this definition to hb-graphs is straightforward. If the graph is disconnected, then each vertex has infinite eccentricity.

For the purpose of evaluation, in this Thesis, we define a **relative eccentricity** as the length of a maximal shortest path starting from a given vertex in S and ending with any vertices of $V \backslash S$. The relative eccentricity is calculated for each vertex of S provided that it is connected to vertices of $V \backslash S$; otherwise, it is set to $-\infty$. The concept of relative eccentricity is illustrated in Figure 2.3.

For the vertex set V, the subset used for relative eccentricity is built by using a threshold value s_V : vertices with α value above this threshold are gathered into a subset $A_V(s_V)$ of V. We consider $B_V(s_V) \triangleq V \backslash A_V(s_V)$, the set of vertices with α values below this

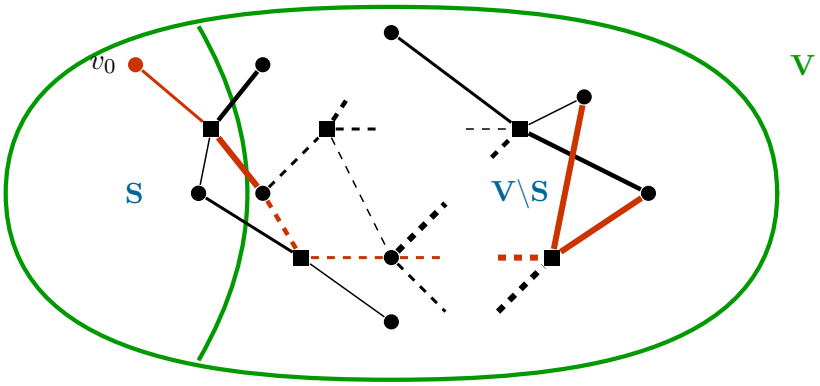


Figure 2.3.: Relative eccentricity: finding the length of a maximal shortest path in the hb-graph starting from a given vertex v_0 of S and finishing with any vertex in $V \backslash S$.

threshold. We evaluate the relative eccentricity of each vertex of $A_V(s_V)$ to vertices of $B_V(s_V)$ in the support hypergraph of the corresponding hb-graph.

Assuming that we stop iterating at time T, we let s_V vary from 0 to the value $\alpha_{T,\max} \triangleq \max_{v \in V}(\alpha_T(v))$—obtained by iterating the algorithm on the hb-graph—in incremental steps and while the eccentricity is kept above 0. In order to have a ratio, we calculate:

$$r_V \triangleq \frac{s_V}{\alpha_{\text{ref}}}$$

where α_{ref} is the reference normalised value used for the initialisation of the α value of the vertices of the hb-graph $\mathfrak{H}$. This ratio has values increasing by steps from 0 to $\frac{\alpha_{T,\max}}{\alpha_{\text{ref}}}$.

We show the results obtained in Figure 2.4 on two plots. The first plot corresponds to the maximal length of the path between vertices of $A_V(s_V)$ and vertices of $B_V(s_V)$ that are connected according to the ratio $r_V = \frac{s_V}{\alpha_{\text{ref}}}$: this path length corresponds to half of the length of the path observed in the extra-vertex graph representation of the hb-graph support hypergraph as in between two vertices of V there is an extra-vertex that represents the hb-edge (or the support hyperedge). The second curve plots the percentage of vertices of V that are in $A_V(s_V)$ in function of r_V. When r_V increases, the number of elements in $A_V(s_V)$ naturally decreases while they get closer to the elements of $B_V(s_V)$, marking the fact that they are central.

Figure 2.5 and Figure 2.6 show that high values of $\alpha_T(v)$ correspond to vertices that are highly connected either by degree or by m-degree.

A similar approach is taken for the hb-edges. Assuming that the diffusion process stops at time T, we use the $\epsilon_{T-\frac{1}{2}}$ function to partition the set of hb-edges into two subsets for a given threshold $s_{\mathfrak{E}}$: the subset $A_{\mathfrak{E}}(s_{\mathfrak{E}})$ of the hb-edges that have ϵ values above the threshold, and $B_{\mathfrak{E}}(s_{\mathfrak{E}})$ the one gathering hb-edges that have ϵ values below $s_{\mathfrak{E}}$.

$s_{\mathfrak{E}}$ varies from 0 to $\epsilon_{T-\frac{1}{2},\max} = \max_{\mathfrak{e} \in \mathfrak{E}}\left(\epsilon_{T-\frac{1}{2}}(\mathfrak{e})\right)$ by incremental steps while keeping the eccentricity above 0, first of the two conditions achieved. In the hb-graph representation,

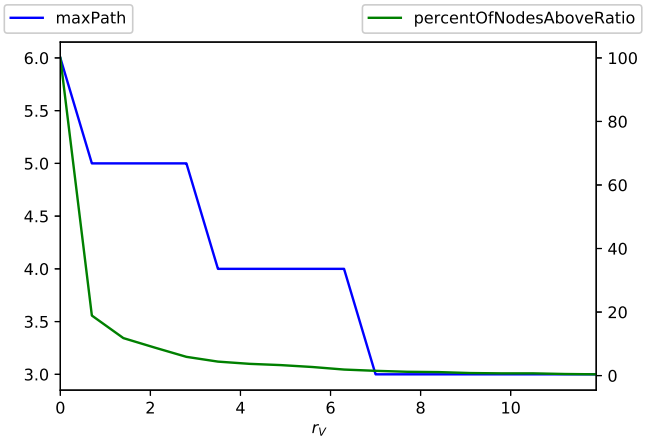


Figure 2.4.: Maximum path length and percentage of vertices in $A_V(s)$ over vertices in V vs ratio r_V.

each hb-edge corresponds to an extra-vertex. Each time, we evaluate the length of the maximal shortest path linking one vertex of $A_{\mathfrak{E}}(s_{\mathfrak{E}})$ to one vertex of $B_{\mathfrak{E}}(s_{\mathfrak{E}})$ for the connected vertices in the hb-graph support hypergraph incident graph: the length of the path corresponds to half of the one obtained from the graph for the same reason than before.

We define the ratio:

$$r_{\mathfrak{E}} \triangleq \frac{s_{\mathfrak{E}}}{\beta_{\text{ref}}}$$

where $\beta_{\text{ref}} \triangleq \frac{1}{|\mathfrak{E}|}$ that corresponds to the normalised value that would be used in the dual hb-graph to initialize the diffusion process. In Figure 2.7, we observe for the hb-edges the same trend than the one observed for vertices: the length of the maximal shortest path between a hb-edge of $A_{\mathfrak{E}}(s_{\mathfrak{E}})$ and any hb-edge of $B_{\mathfrak{E}}(s_{\mathfrak{E}})$ decreases as the ratio $r_{\mathfrak{E}}$ increases, while the percentage of hb-edges in $A_{\mathfrak{E}}(s_{\mathfrak{E}})$ over $\mathfrak{E}$ decreases.

Figure 2.8 shows the high correlation between the value of $\epsilon(\mathfrak{e})$ and the cardinality of $\mathfrak{e}$; Figure 2.9 shows that the correlation between value of $\epsilon(\mathfrak{e})$ and the m-cardinality of $\mathfrak{e}$ is even stronger.

The number of iterations needed to have a significant convergence depends on the initial conditions; we tried different initializations, either uniform, or applying some strokes on a different number of nodes. We observed that the more uniform the information on the network is, the less number of iterations for convergence is needed. No matter the configuration, the most important vertices in term of connectivity are always the most highlighted. Figure 2.10 and in Figure 2.11 depict the convergence observed on a uniform initial distribution as described in the former section. In Figure 2.10, we can see how the α-values, as observed in Figure 2.5, reflect the m-degree of the vertex they are associated to: 200 iterations is far enough to rank the vertices by m-degree. In Figure 2.11, we can observe an analogous phenomena with the ϵ-value associated to hb-edges that reflect the m-cardinality of the hb-edges. Again 200 iterations are sufficient to converge in studied cases.

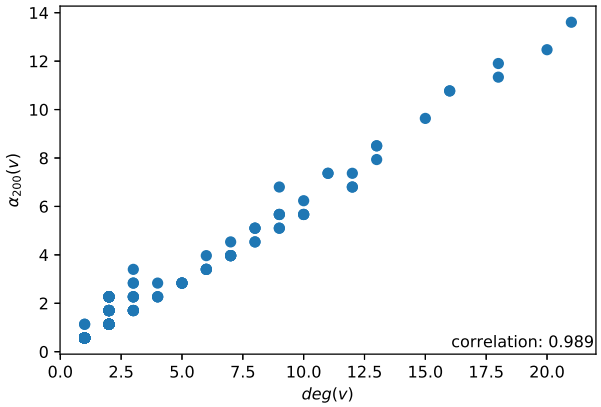


Figure 2.5.: Alpha value of vertices at step 200 and degree of vertices.

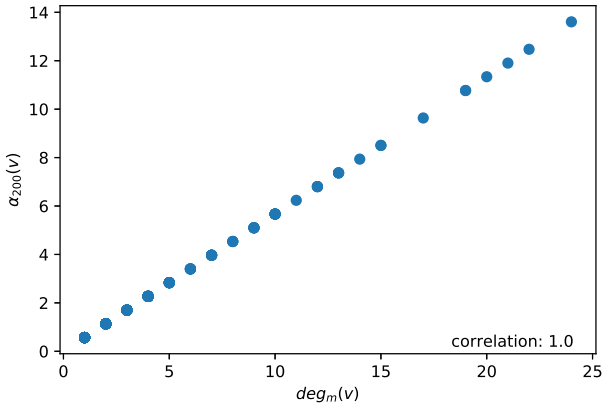


Figure 2.6.: Alpha value of vertices at step 200 and m-degree of vertices.

The number of iterations needed to converge depends on the structure of the network. In the transitory phase, the vertices need to exchange with the hb-edges; the process requires some iterations before converging and its behavior depends on the node connectivity and the hb-edge composition. It remains as an open question to investigate on this transitory phase to have more indications on the way the ϵ and the α-values vary.

We show an example of exchange-based diffusion on a lab-generated hb-graph in Figure 2.12 (a) and on its support hypergraph in Figure 2.12 (b). The vertices are colored depending on the value of the ratio:

$$c_\alpha(v) \triangleq \frac{\alpha_T(v)}{\alpha_{\text{ref}}}$$

using the scale of colors on the right. Vertices with near zero $c_\alpha(v)$ values—i.e. low $\alpha_T(v)$ values compared to α_{ref}—are dark bluish colored; on the opposite, with high $c_\alpha(v)$ values—i.e. with high $\alpha_T(v)$ values compared to α_{ref}—are yellowish colored; when

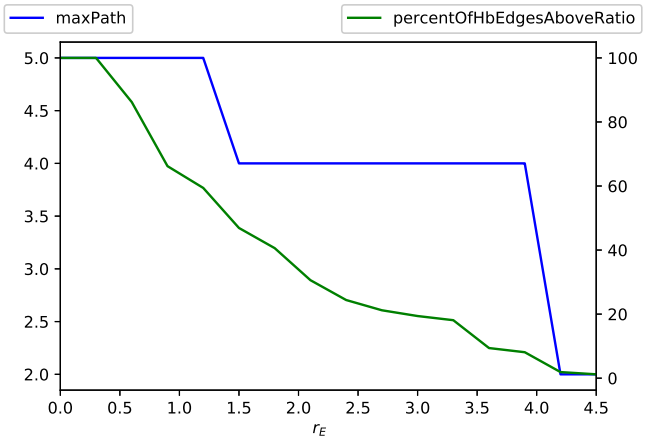


Figure 2.7.: Path maximum length and percentage of vertices in $A_{\mathfrak{E}}(s)$ vs ratio.

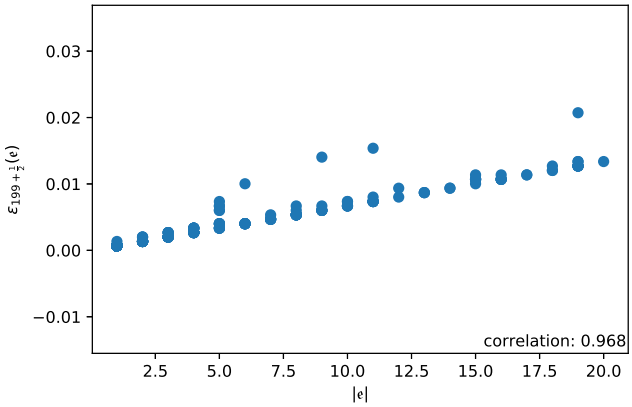


Figure 2.8.: Epsilon value of hb-edge at stage $199+\frac{1}{2}$ and cardinality of hb-edge.

$c_\alpha(v)$ is close to 1, the vertices are colored in a close turquoise. The hb-edges are colored with the left gradient color scale according to the value of the ratio:

$$c_\epsilon(v) \triangleq \frac{\epsilon_{T-\frac{1}{2}}(v)}{\epsilon_{\text{norm}}}$$

where $\epsilon_{\text{norm}}(\mathfrak{e}) \triangleq \sum_{v\in\mathfrak{e}^\star} \frac{m_{\mathfrak{e}}(v)}{\deg_m(v)}\alpha_{\text{ref}}$. $\epsilon_{\text{norm}}(\mathfrak{e})$ corresponds to the value the hb-edge $\mathfrak{e}$ should have in reference to the fraction of α_{ref} given by each vertex and depending on the fraction of its multiplicity versus its m-degree in the hb-edge. Hb-edges are colored using $c_\epsilon(\mathfrak{e})$: when this ratio is close to 0—i.e. when the hb-edges have low $\epsilon_{T-\frac{1}{2}}(v)$ compared to $\epsilon_{\text{norm}}(\mathfrak{e})$—hb-edges are colored in a blueish hue; when this ratio is high—i.e. when the hb-edges have high $\epsilon_{T-\frac{1}{2}}(v)$ compared to what was expected with $\epsilon_{\text{norm}}(\mathfrak{e})$—they are colored in a reddish hue. It is worth mentioning that diffusing only on the support hypergraph of a hb-graph highlights only nodes that are highly

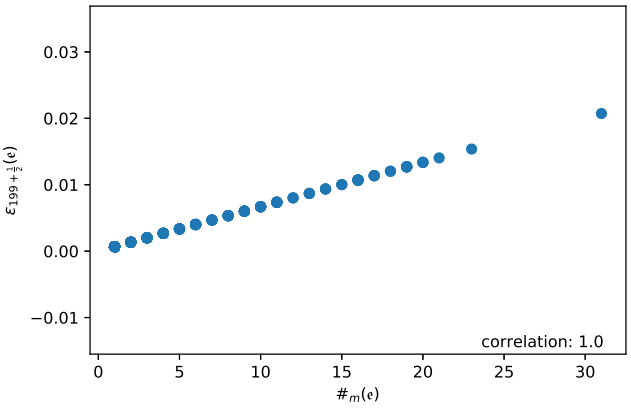


Figure 2.9.: Epsilon value of hb-edge at stage $199+\frac{1}{2}$ and (m-)cardinality of hb-edge.

connected inside a group, the ones being at the intersection of the different groups have less importance in this case. The diffusion on the hb-graph captures the centrality of these vertices that are peripheral to the groups and central for the connectivity of the hb-graph itself. Hence, taking the multiplicities into account brings valuable information on the network and on the centrality of some vertices.

To compare our exchange-based diffusion process to a baseline, we consider a classical random walk. In this classical random walk, the walker who is on a vertex v chooses randomly an incident hb-edge with a uniform probability law, and, when the walker is on a hb-edge $\mathfrak{e}$, he chooses a vertex inside the hb-edge randomly with a uniform probability law. We let the possibility of teleportation between vertices using a tunable parameter $\gamma : 1-\gamma$ represents the probability to be teletransported. We choose $\gamma = 0.85$, which is a classical value used with teleportation. We count the number of passages of the walker through each vertex and each hb-edge. We stop the random walk when the hb-graph is fully explored. We iterate N times the random walk, N varying.

To improve the results with the classical random walk, we propose a modified biased random walk on the hb-graphs—described in Algorithm 2.2—with random choice of the hb-edge when the walker is on a vertex v with a distribution of probability:

$$\left(\frac{w_e\left(\mathfrak{e}_i\right) m_i\left(v\right)}{\deg_{w,m}\left(v\right)}\right)_{i\in[\![p]\!]}$$

and, a random choice of the vertex when the walker is on a hb-edge $\mathfrak{e}$ with a distribution of probability:

$$\left(\frac{m_{\mathfrak{e}}\left(v_i\right)}{\#_m\left(\mathfrak{e}\right)}\right)_{i\in[\![n]\!]}.$$

We keep the possibility of teleportation as it is done in the classical random walk. Similarly to the classical random walk, we count the number of passages of the walker through each vertex and each hb-edge. We also stop the random walk when the hb-graph is fully explored. We iterate N times the random walk with various values of N. Assigning a multiplicity of 1 to every vertex and a weight of 1 to every hb-edge—with

Algorithm 2.2 Modified random walk in hb-graphs.

Given:

A hb-graph $\mathfrak{H} = (V, \mathfrak{E}, w_e)$ with $|V| = n$ and $|\mathfrak{E}| = p$
Number of Random walks: T_{RW}
A teleportation threshold: γ_{th}

Initialisation:

$\forall v \in V : n_V(v) := 0$
$\forall \mathfrak{e} \in \mathfrak{E} : n_{\mathfrak{E}}(\mathfrak{e}) := 0$
$Q :=$ deep copy (V)
$v_0 :=$ random $(v \in Q)$
$n_V(v_0) := 1$
$Q := Q \backslash \{v_0\}$

OneRW():

While $Q \neq \emptyset$:
$\gamma_{\mathrm{rand}} :=$ random $([0;1]$, weight = uniform$)$
if $\gamma_{\mathrm{rand}} < \gamma_{\mathrm{th}}$:
Visit of incident edges
$$\mathfrak{e}_{\mathrm{c}} := \text{random}\left(\mathfrak{e} \in \mathfrak{E} : v_{\mathrm{c}} \in \mathfrak{e}^{\star}, \text{weight} = \left(\frac{w_e(\mathfrak{e}_j)\, m_{\mathfrak{e}_j}(v_0)}{\deg_{w_e,m}(v_0)}\right)_{\mathfrak{e}_j \in \mathfrak{E}}\right)$$
$n_V(\mathfrak{e}_{\mathrm{c}}) := n_V(\mathfrak{e}_{\mathrm{c}}) + 1$
Choice of the next vertex
$$v_0 := \text{random}\left(v \in V : v \in \mathfrak{e}_{\mathrm{c}}^{\star}, \text{weight} = \left(\frac{m_{\mathfrak{e}_{\mathrm{c}}}(v)}{\#_m(\mathfrak{e}_{\mathrm{c}})}\right)_{v \in V}\right)$$
If $v_0 \in Q$:
$Q := Q \backslash \{v_0\}$
$n_V(v_0) := n_V(v_0) + 1$
else:
# Case of teleportation
$$v_0 := \text{random}\left(v \in V : v \in \mathfrak{e}_{\mathrm{c}}^{\star}, \text{weight} = \left(\frac{m_{\mathfrak{e}_{\mathrm{c}}}(v)}{\#_m(\mathfrak{e}_{\mathrm{c}})}\right)_{v \in V}\right)$$
$Q := Q \backslash \{v_0\}$
$n_V(v_0) := n_V(v_0) + 1$

Main():

For $i := 0$ to T_{RW} :
OneRW()
$$\forall v \in V\text{: } \overline{n_V}(v) := \frac{n_V(v)}{T_{\mathrm{RW}}}$$
$$\forall \mathfrak{e} \in \mathfrak{E}\text{: } \overline{n_{\mathfrak{E}}}(\mathfrak{e}) := \frac{n_{\mathfrak{E}}(\mathfrak{e})}{T_{\mathrm{RW}}}$$

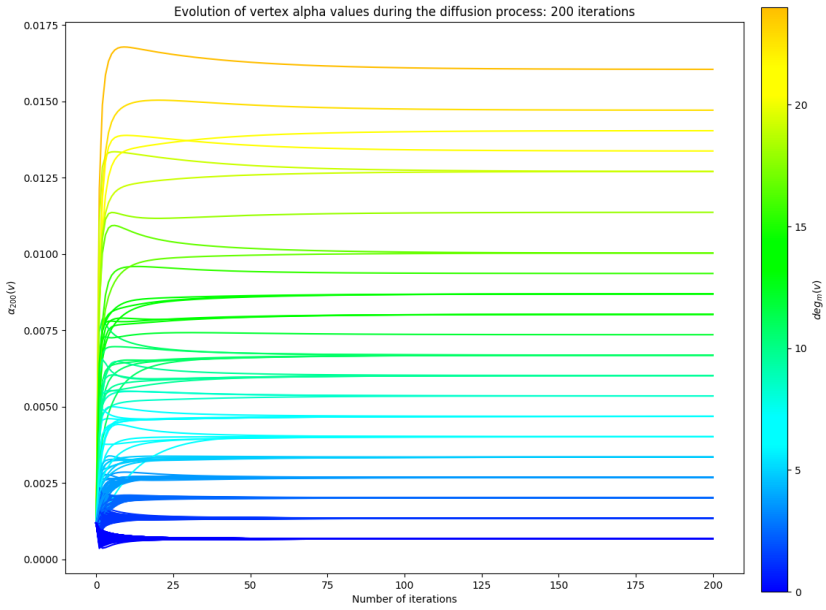


Figure 2.10.: Alpha value convergence of the vertices vs number of iterations. The plots are m-degree-based with gradient coloring.

the vertex degree and the hb-edge cardinality instead of the multiplicity—retrieves the classical random walk from the modified random walk.

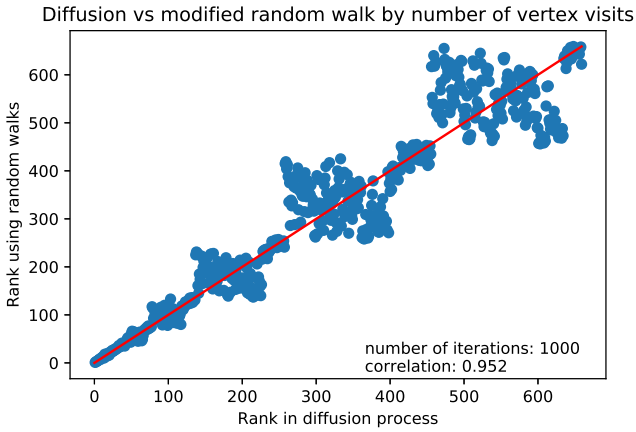


Figure 2.13.: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and the rank obtained with 200 iterations of the exchange-based diffusion process.

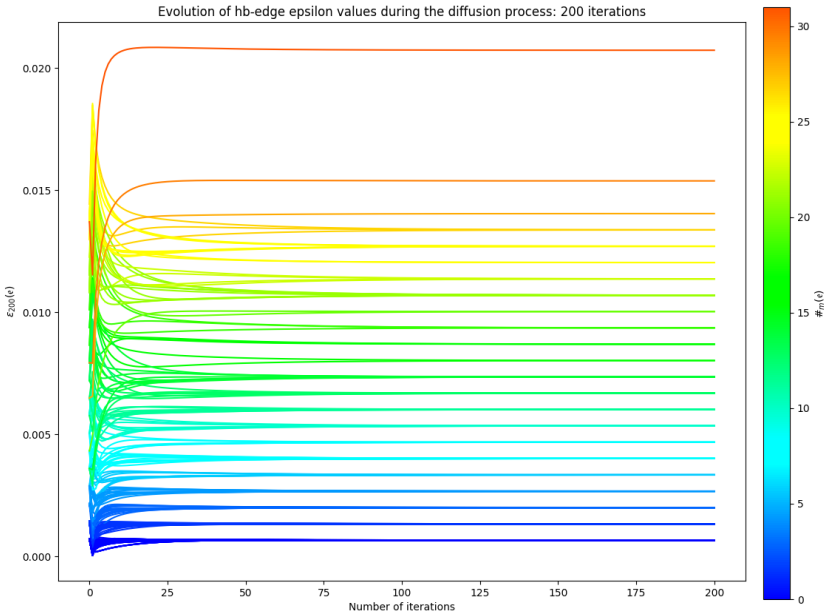


Figure 2.11.: Epsilon value convergence of hb-edges vs number of iterations. The plots are m-cardinality-based colored using gradient colors.

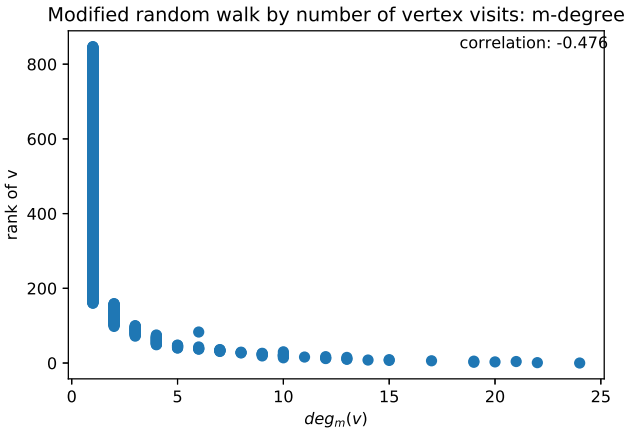


Figure 2.14.: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and m-degree of vertices.

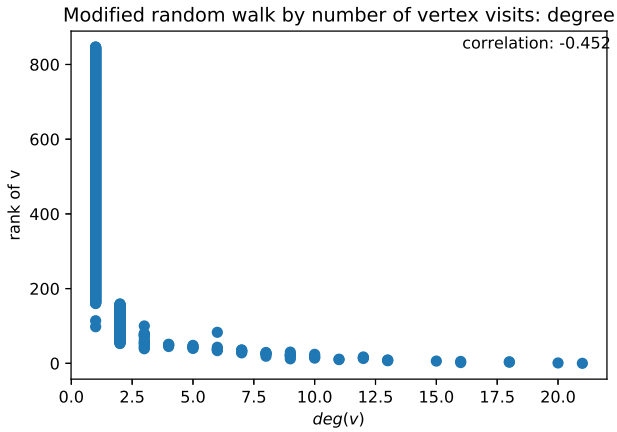


Figure 2.15.: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and degree of vertices.

Figure 2.13 shows that there is a good correlation between the rank obtained by a thousand modified random walks and two hundreds iterations of our diffusion process, especially for the first hundred vertices of the network, which is generally the ones that are targeted. The lack of correlation between the rank obtained by the random walk with the degree of the vertices and the m-degree of vertices as shown respectively in Figure 2.14 and Figure 2.15 is mainly due to the vertices with low m-degrees / degrees.

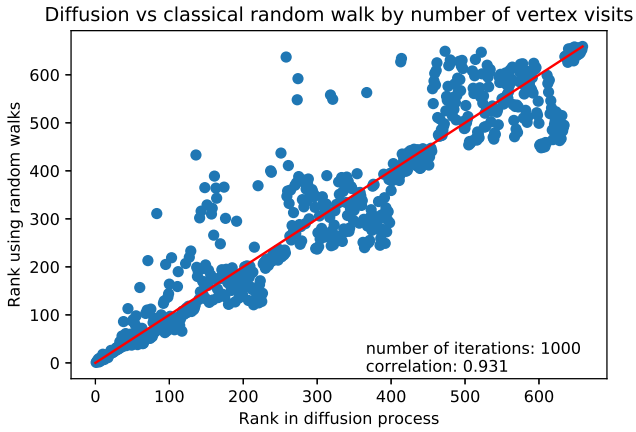


Figure 2.16.: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and rank obtained with 200 iterations of the exchange-based diffusion process.

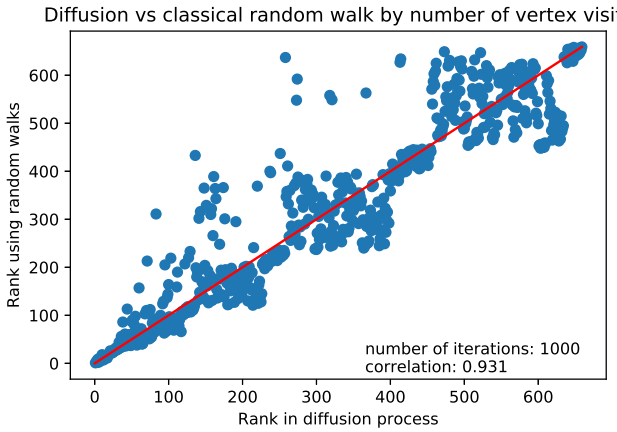


Figure 2.17.: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and m-degree of vertices.

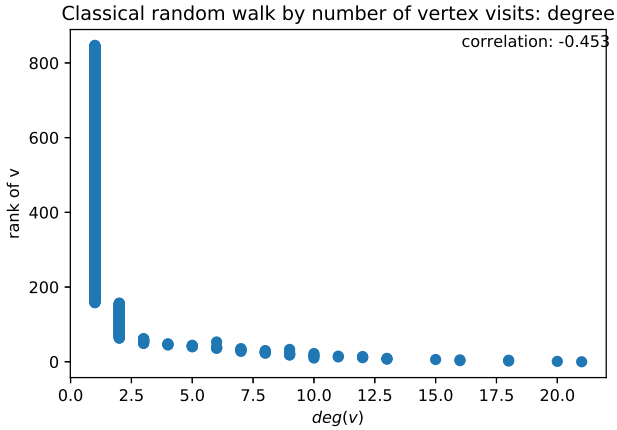


Figure 2.18.: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and degree of vertices.

We can remark in Figure 2.16 that the correlation is a bit lower with a thousand classical random walks due to the fact that there are more vertices that are seen as differently ranked in between the two approaches. In Figure 2.17, we can see that the ranks in the classical random walk rely more on the degree than on the m-degree as shown in Figure 2.18, especially for vertices with small (m-)degrees; but there is still a bad classification for lower (m-)degree vertices.

We have compared the three methods from a computational time perspective; the results are shown in Table 2.1. The diffusion process is clearly faster; the modified random walk, essentially due to the overhead due to the large number of divisions, takes longer than the classical random walk. A lot of optimization can be foreseen to make this modified random walk run faster. The random walks can be easily parallelized; it is also the case

$\lvert\mathfrak{E}\rvert$	$\lvert V\rvert$	k	N_1	N_0	Type of algorithm	100	200	500	1000
55	106	1	5	5	classical random walk	0.40 ± 0.05	0.78 ± 0.07	1.92 ± 0.10	3.82 ± 0.14
55	106	1	5	5	diffusion	0.05 ± 0.02	0.08 ± 0.02	0.20 ± 0.04	0.39 ± 0.06
55	306	1	5	5	modified random walk	0.71 ± 0.06	1.43 ± 0.09	3.56 ± 0.17	7.12 ± 0.23
55	132	3	5	5	classical random walk	0.49 ± 0.05	0.96 ± 0.06	2.36 ± 0.08	4.71 ± 0.12
55	132	3	5	5	diffusion	0.05 ± 0.02	0.09 ± 0.02	0.21 ± 0.04	0.42 ± 0.05
55	132	3	5	5	modified random walk	0.89 ± 0.06	1.77 ± 0.09	4.43 ± 0.13	8.85 ± 0.19
55	91	5	5	5	classical random walk	0.30 ± 0.04	0.59 ± 0.05	1.44 ± 0.06	2.85 ± 0.07
55	91	5	5	5	diffusion	0.04 ± 0.02	0.07 ± 0.02	0.16 ± 0.03	0.31 ± 0.04
55	91	5	5	5	modified random walk	0.55 ± 0.05	1.09 ± 0.06	2.71 ± 0.09	5.42 ± 0.14
305	534	1	5	5	classical random walk	4.05 ± 0.16	8.07 ± 0.26	20.10 ± 0.45	40.17 ± 0.85
305	534	1	5	5	diffusion	0.29 ± 0.06	0.57 ± 0.08	1.35 ± 0.09	2.64 ± 0.10
305	534	1	5	5	modified random walk	6.86 ± 0.28	13.71 ± 0.41	34.16 ± 0.75	68.28 ± 1.21
305	491	3	5	5	classical random walk	3.51 ± 0.13	6.98 ± 0.21	17.39 ± 0.38	34.77 ± 0.70
305	491	3	5	5	diffusion	0.27 ± 0.05	0.53 ± 0.09	1.25 ± 0.11	2.43 ± 0.11
305	491	3	5	5	modified random walk	6.02 ± 0.22	12.03 ± 0.41	30.10 ± 0.73	60.23 ± 1.34
305	499	5	5	5	classical random walk	3.31 ± 0.15	6.58 ± 0.20	16.38 ± 0.34	32.72 ± 0.51
305	499	5	5	5	diffusion	0.24 ± 0.04	0.47 ± 0.06	1.12 ± 0.06	2.18 ± 0.08
305	499	5	5	5	modified random walk	5.86 ± 0.26	11.70 ± 0.37	29.26 ± 0.58	58.51 ± 0.89

Table 2.1.: Time taken for doing 100, 200, 500 and 1000 iterations of the diffusion algorithm and 100, 200, 500 and 1000 classical and modified random walks on different hb-graphs.

for the diffusion process. The number of iterations in the diffusion process can also be optimized. These issues will be addressed in future work.

2.4.2. Two use cases

2.4.2.1. Application to Arxiv querying

We used the standard Arxiv API[1] to perform searches on Arxiv database. When performing a search, the query is transformed into a vector of words which is the basis for the retrieval of documents. The most relevant documents are retrieved based on a similarity measure between the query vector and the word vectors associated to individual documents. Arxiv relies on Lucene's built-in Vector Space Model of information retrieval and the Boolean model[2]. The Arxiv API returns the metadata associated to the document with highest scores for the query performed.

This metadata, filled by the authors during their submission of a preprint, contains different information such as authors, Arxiv categories and abstract.

We process these abstracts using TextBlob, a Python natural language processing library[3] and extract the nouns using the tagged text.

Nouns in the abstract of each document are scored with TF-IDF, the Term Frequency-Invert Document Frequency, defined as:

$$\text{TF-IDF}\,(x,d) \triangleq \text{TF}(x,d)\times\text{IDF}(x,d)$$

[1] https://arxiv.org/help/api/index
[2] https://lucene.apache.org/core/2_9_4/scoring.html
[3] https://textblob.readthedocs.io/en/dev/

with $\mathrm{TF}(x,d)$ the relative frequency of x in d and $\mathrm{IDF}\,(x,d)$ the invert document frequency.

Writing n_d the total number of terms in document d and n_x the number of occurrences of x :

$$\mathrm{TF}(x,d) \triangleq \frac{n_x}{n_d}$$

and, writing N the total number of documents and $n_{x\in d}$ the number of documents having an occurrence of x, we have:

$$\mathrm{IDF}\,(x,d) \triangleq \log_{10}\left(\frac{N}{n_{x\in d}}\right).$$

Scoring each noun in each abstract of the retrieved documents generates a hb-graph $\mathfrak{H}_Q$ of universe the nouns contained in the abstracts. Each hb-edge contains a set of nouns extracted from a given abstract with a multiplicity function that represents the TF-IDF score of each noun.

The exchange-based diffusion process is applied to the hb-graph $\mathfrak{H}_Q$. We show two typical examples on the same query of the first 50 results in Figure 2.19 (a) and of the first 100 results in Figure 2.19 (b). The number of iterations needed to have convergence is less than 10 in these two cases; with 500 results, around 10 iterations are needed for all hb-edges, but for one, where 30 iterations are needed.

As the hb-edges correspond to documents in Arxiv database, we compare the central documents obtained in the results of the queries: we observe that the ranking obtained based on the $\epsilon_{49+\frac{1}{2}}$ differs significantly from the ranking by pertinence given by Arxiv API. In the exchange-based diffusion, the ranking sorts documents depending on their word weights and their centrality as we have seen in the experimental part on random hb-graphs.

Moreover, we have observed that when the number of results retrieved increases the top 5 (respectively top 10) documents—out of the documents retrieved— can sometimes change drastically depending on the retrieval of new documents that are more central with relation to the words they contain. If the gap seems small with a few documents retrieved, this gap increases as the number of documents increases. The increasing number of results reveals the full theoretical hb-graph obtained from the whole dataset performing the querying, and hence, reveals central subjects in this dataset. Hence the diffusion process can allow to highlight the importance of documents by considering central subjects in the processing of the query results.

2.4.2.2. Application to an image database

We have applied the exchange-based diffusion to a database of images. We have used a hb-graph modeling of the objects detected on individual images to build a network of co-occurrences. Each image has been processed using a Retina neural network to label the objects it contains, and each object is then counted in its own category. The database used is the 2014 training set of the COCO dataset[4] [LMB+14]. The use of a pre-trained Retina net[5] allows to give bounding boxes corresponding to concepts, with a

[4] http://cocodataset.org/#home.
[5] https://github.com/fizyr/keras-retinanet

probability associated to them. A threshold of 0.5 is chosen—as proposed by the library developer—below which the bounding box is rejected. Hence, each image is associated to its concepts and their multiplicity.

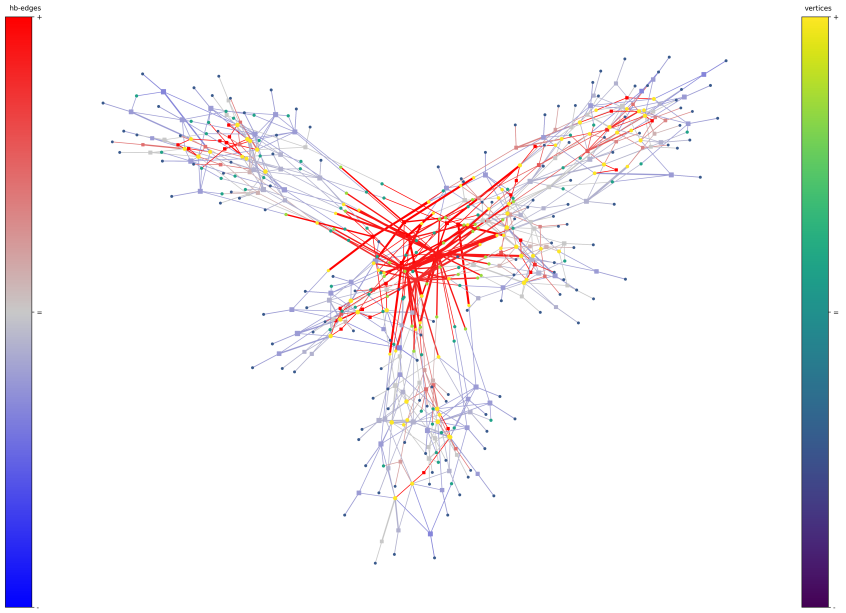
Out of this association, two hb-graphs can be build. First, a hb-graph of images $\mathfrak{H}_{\mathrm{Im}}$, where the vertex set is constituted of the different concepts—objects—that the image holds and where a hb-edge is related to an image, regrouping the different concepts with their respective multiplicity. The second hb-graph is the hb-graph of concepts $\mathfrak{H}_{\mathrm{Co}}$: the vertex set corresponds to the image set, and a hb-edge regroups the images holding the concept with a multiplicity that corresponds to the number of times the corresponding concept occurs in the image. These two hb-graphs are dual one of the other. We now focus on the hb-graph of images.

We randomly select 200 images of the COCO 2014 training dataset, building the original image hb-graph. To ensure connectivity, we focus on the first main component of the original image hb-graph. This component is designated as the hb-graph in the remainder. We then enhance the diffusion on this connected hb-graph. A typical result is presented in Figure 2.20: the concepts are the vertices, the images represent the extra-vertices corresponding to the hb-edges. The coloration of vertices—i.e. the nodes of the concepts— and of hb-edges—i.e. the extra-nodes representing images— is the same than the one used in Figure 2.12. The closer to red the images are, the more central to the sample drawn they are; hence, these images can potentially be used to make a summary of this sample.

2.5. Future work and Conclusion

The results obtained by using hb-graph highlight the possibility of using hb-edges for analyzing networks; they confirm that vertices are highlighted due to their connectivity. The highlighting of the hb-edges has been achieved by using the intermediate step of our diffusion process. Different applications can be thought in particular in the search of tagged multimedia documents for refining the results and scoring of documents retrieved. Using tagged documents ranking by this means could help in creating summary for visualisation. Our approach is seen as a strong basis to refine the approach of [XDYF16]. This approach can also be viewed as a means to make query expansion and disambiguation by using additional high scored words in the network and a way of making some recommendation based on the scoring of a document based on its main words. We have also proposed a generalization of this approach in Appendix Section D.1 using vertex and hb-edge abstract information functions and biased probability transition.

(a) Exchange-based diffusion on the hb-graph



(b) Exchange-based diffusion on the hb-graph support

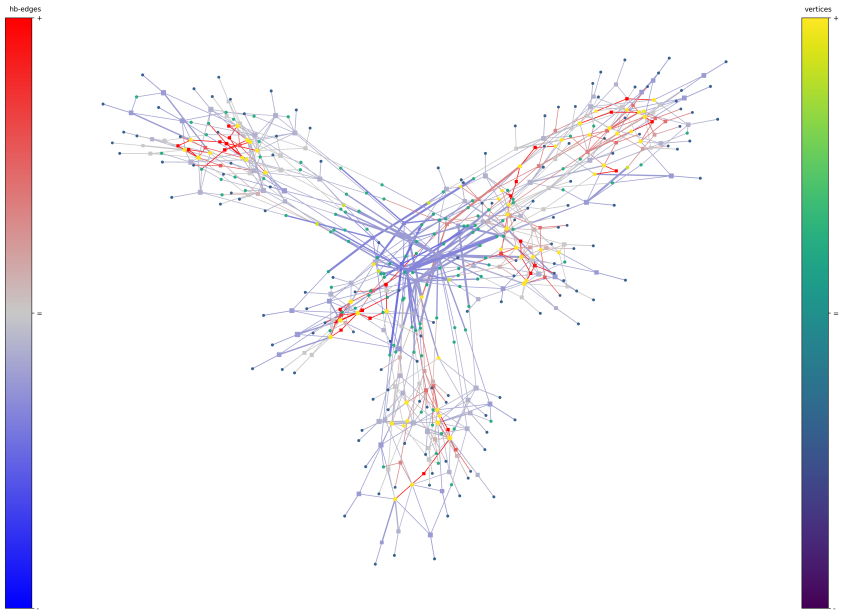
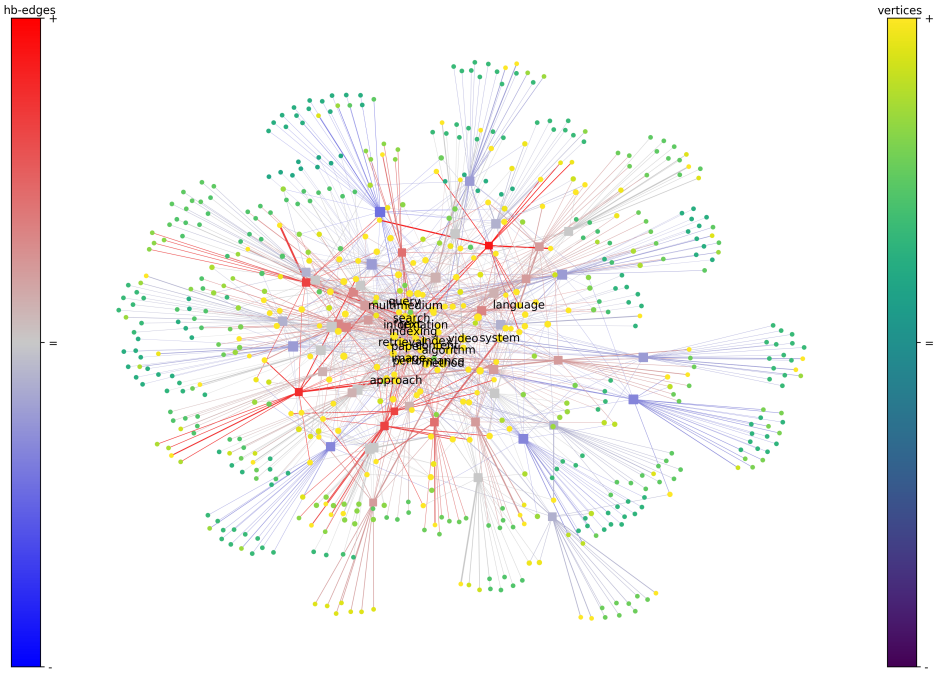
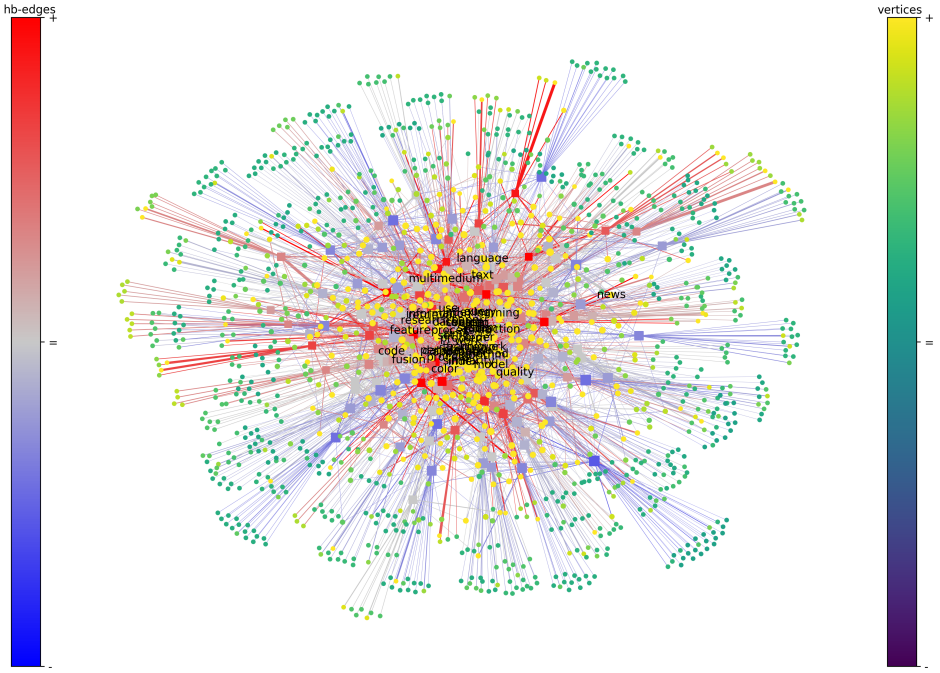


Figure 2.12.: Exchange-based diffusion in a hb-graph (a) and its support hypergraph (b) after 200 iterations of Algorithm 2.1: highlighting important hb-edges. Simulation with 848 vertices (chosen randomly out of 10 000) gathered in 5 groups of vertices (with 5, 9, 14, 16 and 9 important vertices and 2 important vertices per hb-edge), 310 hb-edges (with cardinality of support less or equal to 20), 10 vertices in between the 5 groups. Extra-vertices have square shape and are colored with the hb-edge color scale.



Sub-figure (a): 50 most relevant articles have been retrieved with 100 iterations.
Top 10: 1: multimedium; 2: video; 3: search; 4: retrieval; 5: image; 6: indexing; 7: paper; 8: index; 9: method; 10: system;



Sub-figure (b): 100 most relevant articles have been retrieved with 100 iterations.
Top 10: 1: paper; 2: index; 3: multimedium; 4: image; 5: method; 6: video; 7: retrieval; 8: performance; 9: indexing; 10: system;

Figure 2.19.: Querying Arxiv. The search performed is "content-based multimedia indexing" for which (a) 50 most—respectively (b) 100 most—relevant articles have been retrieved.

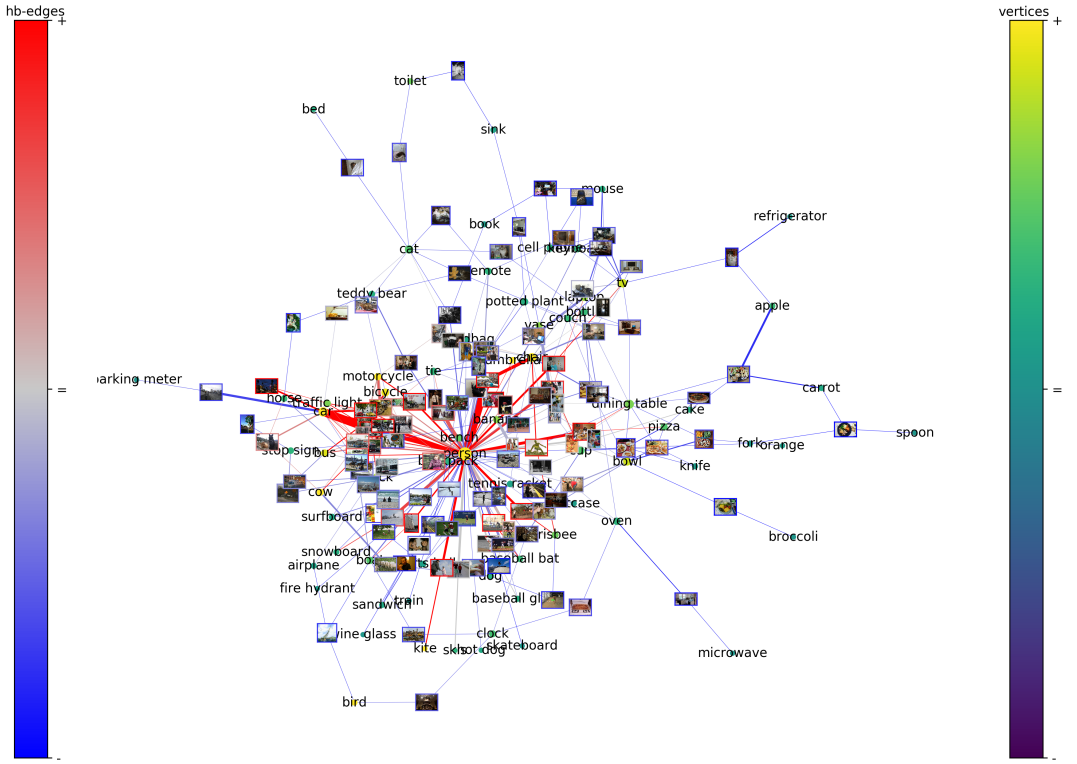


Figure 2.20.: Exchange-based diffusion on the sub-hb-graph first component with 177 images of a hb-graph of 200 images of the COCO 2014 training dataset.

3. e-adjacency tensor of natural hb-graphs

Highlights

This chapter is based on [OLGMM18b, OLGMM19a, OLGMM19c].

Prerequisites: Sections 1.3 and 1.4, Appendix B—particularly Section B.3—, and possibly Appendix Section E.1.4.

3.1. Motivation

In [OLGMM17a, OLGMM18a]—reproduced in Appendix Section E.1—we have presented a first e-adjacency tensor for general hypergraphs. Its construction has been achieved by involving a Hypergraph Uniformisation Process, coupled to a Polynomial Homogenization Process. The Uniformisation requires the addendum of several special vertices, one per layer of the hypergraph decomposition.

Nonetheless, in Chapter 1, one of the motivations for the introduction of hb-graph was that hb-graphs are a way of reducing the number of variables used in the Polynomial Homogenization Process, while keeping the Uniformisation of the Structure interpretable in term of m-uniformisation of hb-graphs.

Research question 3.1. *Which e-adjacency tensor(s) can we build for general hb-graphs?*

In this chapter, we mostly rely on the related work presented in [OLGMM17a]. We then present our expectations on the desirable properties the e-adjacency tensors of general hb-graphs should have and present three possible solutions. After discussing their properties, we justify the choice for the best candidate, and apply it to the particular case of general hypergraphs. This choice will be refined in Chapter 4.

3.2. Related work

In addition to the related work presented in Appendix Section E.1 for general hypergraphs, we found, after having developed the whole theory on hb-graphs, [PZ15]. Nonetheless, as it is a strong anteriority, we mention it here.

A ***[k-]*adjacency tensor** for k-PZ-multigraphs—i.e. k-m-uniform natural hb-graph with no repeated hb-edge—is defined as follow:

Definition 3.1. *Let $\dddot{\mathcal{H}}_k = (V, E)$ be a k-PZ-multigraph on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a set of edges $E = (e_j)_{j\in[\![p]\!]}$.*

*The **[k-]adjacency tensor** of a k-PZ-multigraph is the symmetric tensor $\mathcal{A}_{\dddot{\mathcal{H}}_m} \in \mathcal{T}_{k,n}$ of CHR $\boldsymbol{A}_{\dddot{\mathcal{H}}_k} = (a_{i_1,\dots,i_k})_{i_1,\dots,i_k\in[\![n]\!]}$ such that:*

$$a_{i_1,\dots,i_k} = \begin{cases} \dfrac{m_{j_1}!\dots m_{j_s}!}{(m-1)!}, & \text{if } \{\{v_{i_1},\dots,v_{i_m}\}\} = \left\{v_{j_1}^{m_{j_1}},\dots,v_{j_s}^{m_{j_s}}\right\} \in E; \\ 0, & \text{otherwise.} \end{cases}$$

The authors then study some spectral properties of k-PZ-multigraph. However, the subject of general natural hb-graphs remains to be tackled.

3.3. e-adjacency tensor of a natural hb-graph

Postulate 3.1. *In this section, we consider only natural hb-graphs with no repeated hb-edge.*

To build the e-adjacency tensor $\mathcal{A}(\mathfrak{H})$ of a natural hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ without repeated hb-edge—of vertex set $V = \{v_i : i \in [\![n]\!]\}$ and hb-edge set $\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![p]\!]}$—, we take an approach similar to Appendix Section E.1 using the strong link between cubical symmetric tensors and homogeneous polynomials.

3.3.1. Expectations for the e-adjacency tensor

We formulate the expected properties for the e-adjacency tensor of a natural hb-graph we want to construct. For general hypergraphs, we have insisted on the interpretation of the construction using a Hypergraph Uniformisation Process: that has required the filling of the hyperedges with additional and two-by-two different vertices since hyperedges cannot have duplicated elements.

As natural hb-graphs "naturally" allow vertices to be duplicated, we can think of different ways of filling the hb-edges with additional vertices. We have chosen three of them: the straightforward approach, the silo approach and the layered approach—as it was already done for the layered approach for general hypergraphs in Appendix Section E.1.

We first give some expected properties of such a tensor, some of them being more qualitative than quantifiable.

Expectation 3.1. *The e-adjacency tensor should be non-negative, symmetric and its generation should be as simple as possible.*

Since non-negative symmetric hypermatrices can be described with a small number of values, a hb-edge is representable with only one tuple of indices and their corresponding coefficient, the other coefficients of the tensor being obtained by permuting the indices of the first one while the same value is kept. Moreover, in the spectral theory, symmetric non-negative nonzero tensors ensure interesting properties as their spectral radius is positive [QL17]; furthermore, there is at most one H-eigenvalue that corresponds to the spectral radius with a positive Perron H-eigenvector.

Expectation 3.2. *The tensor should be globally invariant to vertex permutation in the original hb-graph.*

By globally invariant, we mean that a permutation of rows on each face of the hypermatrix follows the same permutation than the one involved in the vertex permutation. We do not expect the special vertices added for the filling of the hb-edges to follow the same rule.

Expectation 3.3. *The e-adjacency tensor should induce a unique reconstruction of the hb-graph it is originated from.*

The e-adjacency tensor should describe the hb-graph in a unique way up to a permutation of indices, so that no two hb-graphs have the same e-adjacency tensor unless they are isomorphic. This is a strong requirement as it enforces the addition of special vertices even for k-m-uniform hb-graphs, where $\overline{k}$-adjacency corresponds to k-adjacency. Hence, the special vertices will be systematically generated and added to the final tensor in order to meet this expectation.

Expectation 3.4. *Given the choice of two representations the one that can be described with the least possible number of elements should be chosen. Then the sparsest e-adjacency tensor should be chosen.*

It forces the hypermatrix to be easily describable before ensuring the lowest sparsity possible. The fact that the hypermatrix is symmetric will help.

Expectation 3.5. *The e-adjacency tensor should allow direct retrieval of the vertex degrees.*

This is required for all k-adjacency tensors of uniform hypergraphs. It is also the case for the e-adjacency tensors of [SZB19, BCM17] and for the first e-adjacency tensor we built for general hypergraphs.

3.3.2. Elementary hb-graph

A hb-graph that has only one non repeated hb-edge in its hb-edge family is called an **elementary hb-graph.**

Claim 3.1. *Let $\mathfrak{H} = (V, \mathfrak{E})$ be a hb-graph with no repeated hb-edge.*

Then:

$$\mathfrak{H} = \bigoplus_{\mathfrak{e}\in\mathfrak{E}} \mathfrak{H}_{\mathfrak{e}}$$

where $\mathfrak{H}_{\mathfrak{e}} = (V, (\mathfrak{e}))$ is the elementary hb-graph associated to the hb-edge $\mathfrak{e}$.

Proof. Let $\mathfrak{e}_1 \in \mathfrak{E}$ and $\mathfrak{e}_2 \in \mathfrak{E}$. As $\mathfrak{H}$ is with no repeated hb-edge, $\mathfrak{e}_1 + \mathfrak{e}_2$ does not contain any new pairs of repeated elements. Thus $\mathfrak{H}_{\mathfrak{e}_1} + \mathfrak{H}_{\mathfrak{e}_2}$ is a direct sum and can be written $\mathfrak{H}_{\mathfrak{e}_1} \oplus \mathfrak{H}_{\mathfrak{e}_2}$.

A straightforward iteration over elements of $\mathfrak{e} \in \mathfrak{E}$ leads to the result.

□

We need first to define the $\overline{k}$-adjacency hypermatrices for an elementary hb-graph and for a m-uniform hb-graph.

3.3.3. Normalised $\overline{k}$-adjacency tensor of an elementary hb-graph

We consider an elementary hb-graph $\mathfrak{H}_{\mathfrak{e}} = (V, (\mathfrak{e}))$ where $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{e}$ a multiset of universe V and multiplicity function m. The support of $\mathfrak{e}$ is $\mathfrak{e}^\star = \{v_{j_1}, \ldots, v_{j_k}\}$ by considering, without any loss of generality: $1 \leqslant j_1 < \ldots < j_k \leqslant n$.

$\mathfrak{e}$ is the multiset: $\mathfrak{e} = \left\{v_{j_1}^{m_{j_1}}, \ldots, v_{j_k}^{m_{j_k}}\right\}$ where $m_j = m(v_j)$.

The normalised hypermatrix representation of $\mathfrak{e}$, written $\boldsymbol{Q}_{\mathfrak{e}}$, describes uniquely the mset $\mathfrak{e}$. Thus the elementary hb-graph $\mathfrak{H}_{\mathfrak{e}}$ is also uniquely described by $\boldsymbol{Q}_{\mathfrak{e}}$ as $\mathfrak{e}$ is its unique hb-edge. $\boldsymbol{Q}_{\mathfrak{e}}$ is of rank $r = \#_m\mathfrak{e} = \sum_{j=1}^{k} m_j$ and dimension n.

Hence, the definition:

Definition 3.2. *Let $\mathfrak{H} = (V, (\mathfrak{e}))$ be an elementary hb-graph with $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{e}$ the multiset $\left\{v_{j_1}^{m_{j_1}}, \ldots, v_{j_k}^{m_{j_k}}\right\}$ of m-rank $r = \#_m\mathfrak{e}$, universe V and multiplicity function m.*

*The **normalised $\overline{k}$-adjacency hypermatrix of an elementary hb-graph** $\mathfrak{H}_{\mathfrak{e}}$ is the normalised representation of the multiset $\mathfrak{e}$, i.e. the symmetric hypermatrix $\boldsymbol{Q}_{\mathfrak{e}} \triangleq (q_{i_1\ldots i_r})_{i_1,\ldots,i_r\in[\![n]\!]}$ of rank r and dimension n where the only nonzero elements are:*

$$q_{\sigma\left(j_1^{m_i j_1},\ldots,j_k^{m_i j_k}\right)} = \frac{m_{ij_1}!\ldots m_{ij_k}!}{(r-1)!}$$

where $\sigma \in \mathcal{S}_{[\![r]\!]}$[1].

In an elementary hb-graph, $\overline{k}$-adjacency corresponds to $\#_m\mathfrak{e}$-adjacency. This hypermatrix encodes the $\overline{k}$-adjacency of an elementary hb-graph.

[1] $\mathcal{S}_{[\![r]\!]}$ designates the set of permutations on $[\![r]\!]$.

3.3.4. Hb-graph polynomial

Homogeneous polynomial associated to a hypermatrix:

With an approach similar to Appendix Section E.1 where full details are given, let $\boldsymbol{e_1}, \ldots, \boldsymbol{e_n}$ be the canonical basis of $\mathbb{R}^n$.

$(\boldsymbol{e_{i_1}} \otimes \ldots \otimes \boldsymbol{e_{i_k}})_{i_1,\ldots,i_k \in [\![n]\!]}$ is a basis of $\mathcal{L}_k^0(\mathbb{K}^n)$, where $\otimes$ is the Segre outerproduct.

A tensor $\mathcal{Q} \in \mathcal{L}_k^0(\mathbb{K}^n)$ is associated to a hypermatrix $\boldsymbol{Q} = (q_{i_1\ldots i_r})_{i_1,\ldots,i_r \in [\![n]\!]}$ by writing $\mathcal{Q}$ as:

$$\mathcal{Q} = \sum_{i_1,\ldots,i_r \in [\![n]\!]} q_{i_1\ldots i_r} \boldsymbol{e_{i_1}} \otimes \ldots \otimes \boldsymbol{e_{i_r}}.$$

$\boldsymbol{Q}$ is called the **canonical hypermatrix representation** (CHR for short) of $\mathcal{Q}$.

Considering n variables z_i attached to the n vertices v_i and $z = \sum_{i \in [\![n]\!]} z_i \boldsymbol{e_i}$, the multilinear matrix product $(z, \ldots, z) \cdot \boldsymbol{Q} = (z)_{[r]} \cdot \boldsymbol{Q}$ is a polynomial $P(\boldsymbol{z_0})$[2]:

$$P(\boldsymbol{z_0}) = \sum_{i_1,\ldots,i_r \in [\![n]\!]} q_{i_1\ldots i_r} z_{i_1} \ldots z_{i_r}$$

of degree r.

Elementary hb-graph polynomial:

Let $\mathfrak{H}_{\mathfrak{e}} = (V, (\mathfrak{e}))$ be an elementary hb-graph with $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{e}$ the multiset $\left\{v_{j_1}^{m_{j_1}}, \ldots, v_{j_k}^{m_{j_k}}\right\}$ of m-rank $r = \#_m \mathfrak{e}$, universe V and multiplicity function m.

Using the normalised $\overline{k}$-adjacency hypermatrix $\boldsymbol{Q_{\mathfrak{e}}} = (q_{i_1\ldots i_r})_{i_1,\ldots,i_r \in [\![n]\!]}$, which is symmetric, we can write the reduced version of its attached homogeneous polynomial $P_{\mathfrak{e}}$:

$$\begin{aligned} P_{\mathfrak{e}}(\boldsymbol{z_0}) &= \frac{r!}{m_{j_1}! \ldots m_{j_k}!} q_{j_1^{m_{j_1}} \ldots j_k^{m_{j_k}}} z_{j_1}^{m_{j_1}} \ldots z_{j_k}^{m_{j_k}} \\ &= \#_m \mathfrak{e}\, z_{j_1}^{m_{j_1}} \ldots z_{j_k}^{m_{j_k}}. \end{aligned}$$

Hb-graph polynomial:

Considering a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ with no-repeated hb-edge, with $V = \{v_i : i \in [\![n]\!]\}$ and $\mathfrak{E} = (\mathfrak{e}_i)_{i \in [\![p]\!]}$.

This hb-graph can be summarized by a polynomial of degree $r_{\mathfrak{H}} = \max_{\mathfrak{e} \in \mathfrak{E}} \#_m(\mathfrak{e})$:

$$\begin{aligned} P(\boldsymbol{z_0}) &\triangleq \sum_{i \in [\![p]\!]} c_{\mathfrak{e}_i} P_{\mathfrak{e}_i}(\boldsymbol{z_0}) \\ &= \sum_{i \in [\![p]\!]} c_{\mathfrak{e}_i} \frac{r_i!}{m_{ij_1}! \ldots m_{ij_{k_i}}!} q_{j_1^{m_{ij_1}} \ldots j_{k_i}^{m_{ij_{k_i}}}} z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}} \\ &= \sum_{i \in [\![p]\!]} c_{\mathfrak{e}_i} \#_m \mathfrak{e}_i\, z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}} \end{aligned}$$

[2]Where: $\boldsymbol{z_0} = (z_1, \ldots, z_n)$

where $c_{\mathfrak{e}_i}$ is a technical coefficient. $P(\boldsymbol{z_0})$ is called the **hb-graph polynomial**. The choice of $c_{\mathfrak{e}_i}$ will be further made in Section 3.3.8 in order to retrieve the m-degree of the vertices from the e-adjacency tensor.

3.3.5. $\overline{k}$-adjacency hypermatrix of a m-uniform natural hb-graph

We now extend to m-uniform hb-graphs the $\overline{k}$-adjacency hypermatrix obtained in the case of an elementary hb-graph.

In the case of a r-m-uniform natural hb-graph with no repeated hb-edge, each hb-edge has the same m-cardinality r. Hence, the $\overline{k}$-adjacency of a r-m-uniform hb-graph corresponds to r-adjacency where r is the m-rank of the hb-graph. The $\overline{k}$-adjacency tensor of the hb-graph has rank r and dimension n. The elements of the $\overline{k}$-adjacency hypermatrix are: $a_{i_1\dots i_r}$ with $i_1,\dots,i_r \in [\![n]\!]$.

The associated hb-graph polynomial is homogeneous of degree r.

We obtain the definition of the $\overline{k}$-adjacency tensor of a r-m-uniform hb-graph by summing the $\overline{k}$-adjacency tensor attached to each hb-edge with a coefficient c_i equals to 1 for each hb-edge.

Definition 3.3. *Let $\mathfrak{H} = (V, \mathfrak{E})$ be a hb-graph with $V = \{v_i : i \in [\![n]\!]\}$.*

*The **$\overline{k}$-adjacency hypermatrix of a r-m-uniform hb-graph** $\mathfrak{H} = (V, \mathfrak{E})$ is the hypermatrix $\boldsymbol{A}_{\mathfrak{H}} = (a_{i_1\dots i_r})_{i_1,\dots,i_r \in [\![n]\!]}$ defined by:*

$$\boldsymbol{A}_{\mathfrak{H}} \triangleq \sum_{i \in [\![p]\!]} \boldsymbol{Q}_{\mathfrak{e}_i}$$

where $\boldsymbol{Q}_{\mathfrak{e}_i}$ is the $\overline{k}$-adjacency hypermatrix of the elementary hb-graph associated to the hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, \dots, v_{j_{k_i}}^{m_{ij_{k_i}}}\right\} \in \mathfrak{E}$.

The only non-zero elements of $\boldsymbol{Q}_{\mathfrak{e}_i}$ are the elements with indices obtained by permutation of the elements of the multiset $\left\{j_1^{m_{ij_1}}, \dots, j_{k_i}^{m_{ij_{k_i}}}\right\}$ and are all equal to:

$$\frac{m_{ij_1}! \dots m_{ij_{k_i}}!}{(r-1)!}.$$

This definition corresponds to the definition given by [PZ15], which is a symmetrized version of the one given in [PZ14].

We can remark that when a r-m-uniform hb-graph has 1 as vertex multiplicity for any vertices in each hb-edge support of all hb-edges, then this hb-graph is a r-uniform hypergraph: in this case, we retrieve the result of the degree-normalized tensor defined in [CD12].

Claim 3.2. *The m-degree of a vertex v_j in a r-m-uniform hb-graph $\mathfrak{H}$ of the $\overline{k}$-adjacency hypermatrix is:*

$$\deg_m(v_j) = \sum_{j_2,\dots,j_r \in [\![n]\!]} a_{jj_2\dots j_r}.$$

Proof. $\sum_{j_2,\dots,j_r \in [\![n]\!]} a_{jj_2\dots j_r}$ has non-zero terms only for the corresponding hb-edges $\mathfrak{e}_i$ containing v_j. Such a hb-edge is described by:

$$\mathfrak{e}_i = \left\{ v_j^{m_{i\,j}}, v_{l_2}^{m_{i\,l_2}}, \dots, v_{l_{k_i}}^{m_{i\,l_{k_i}}} \right\}.$$

It means that the multiset $\{\{j_2, \dots, j_r\}\}$ corresponds exactly to the multiset:

$$\left\{ j^{m_{i\,j-1}}, l_2^{m_{i\,l_2}}, \dots, l_{k_i}^{m_{i\,l_{k_i}}} \right\}.$$

For each $\mathfrak{e}_i$ such that $v_j \in \mathfrak{e}_i$, there are $\dfrac{(r-1)!}{(m_{i\,j}-1)!m_{i\,l_2}!\dots m_{i\,l_{k_i}}!}$ possible permutations of the indices j_2 to j_l and:

$$a_{jj_2\dots j_r} = \frac{m_{i\,j}!m_{i\,l_2}!\dots m_{i\,l_{k_i}}!}{(r-1)!}.$$

Also:

$$\sum_{j_2,\dots,j_r \in [\![n]\!]} a_{jj_2\dots j_r} = \sum_{i \in [\![p]\!] \,:\, v_j \in \mathfrak{e}_i} m_{i\,j} = \deg_m(v_j).$$

□

3.3.6. Elementary operations on hb-graphs

In Appendix Section E.1, we describe two elementary operations that are used in the hypergraph uniformisation process. We describe here similar operations for hb-graphs.

Operation 3.3.1. *Let $\mathfrak{H} = (V, \mathfrak{E})$ be a hb-graph.*
Let w_1 be a constant weighted function on hb-edges with constant value 1.
*The weighted hb-graph $\mathfrak{H}_1 \overset{\Delta}{=} (V, \mathfrak{E}, w_1)$ is called the **canonical weighted hb-graph** of $\mathfrak{H}$.*
*The application $\phi_{cw} : \mathfrak{H} \mapsto \mathfrak{H}_1$ is called the **canonical weighting operation**.*

Operation 3.3.2. *Let $\mathfrak{H}_1 = (V, \mathfrak{E}, w_1)$ be a canonical weighted hb-graph.*
Let $c \in \mathbb{R}^{++}$. Let w_c be a constant weighted function on hb-edges with constant value c.
*The weighted hb-graph $\mathfrak{H}_c \overset{\Delta}{=} (V, \mathfrak{E}, w_c)$ is called the **c-dilated hb-graph** of $\mathfrak{H}$.*
*The application $\phi_{c\text{-}d} : \mathfrak{H}_1 \mapsto \mathfrak{H}_c$ is called the **c-dilatation operation**.*

Operation 3.3.3. *Let $\mathfrak{H}_w = (V, \mathfrak{E}, w)$ be a weighted hb-graph. Let y be a vertex that is not in V. The* ***y-complemented hb-graph*** *of $\mathfrak{H}_w$ is the hb-graph $\mathfrak{H}_{y^c} \triangleq (V_{y^c}, \mathfrak{E}_{y^c}, w_{y^c})$ where:*

- $V_{y^c} \triangleq V \cup \{y\}$;
- $\mathfrak{E}_{y^c} \triangleq (\xi(\mathfrak{e}))_{\mathfrak{e}\in\mathfrak{E}}$ *where the map* $\xi : \mathfrak{E} \to \mathcal{M}(V_{y^c})$ *is such that for all* $\mathfrak{e} \in \mathfrak{E}$, $\xi(\mathfrak{e}) \triangleq \left\{x^{m_{\xi(\mathfrak{e})}(x)} : x \in V_{y^c}\right\} \in \mathcal{M}(V_{y^c})$ *with* $m_{\xi(\mathfrak{e})}(x) \triangleq \begin{cases} m_{\mathfrak{e}}(x) & \text{if } x \in \mathfrak{e}^\star; \\ r_{\mathfrak{H}} - \#_m \mathfrak{e} & \text{if } x = y; \end{cases}$
- *the weight function w_{y^c} is such that $\forall \mathfrak{e} \in \mathfrak{E}$: $w_{y^c}(\xi(\mathfrak{e})) \triangleq w(\mathfrak{e})$.*

The application $\phi_{y\text{-}c} : \mathfrak{H}_w \mapsto \mathfrak{H}_{y^c}$ is called the ***y-complemented operation****.*

Operation 3.3.4. *Let $\mathfrak{H}_w = (V, \mathfrak{E}, w)$ be a weighted hb-graph. Let y be a vertex that is not in V. Let $\alpha \in \mathbb{R}^{++}$. The y^α* ***-vertex-increased hb-graph*** *of $\mathfrak{H}_w$ is the hb-graph $\mathfrak{H}^+_{w^+} \triangleq (V^+, \mathfrak{E}^+, w^+)$ where:*

- $V^+ \triangleq V \cup \{y\}$;
- $\mathfrak{E}^+ \triangleq (\phi(\mathfrak{e}))_{\mathfrak{e}\in\mathfrak{E}}$ *where the map* $\phi : \mathfrak{E} \to \mathcal{M}(V^+)$ *such that for all* $\mathfrak{e} \in \mathfrak{E}$, $\phi(\mathfrak{e}) \triangleq \left\{x^{m_{\phi(\mathfrak{e})}(x)} : x \in V^+\right\} \in \mathcal{M}(V^+)$ *with* $m_{\phi(\mathfrak{e})}(x) \triangleq \begin{cases} m_{\mathfrak{e}}(x) & \text{if } x \in \mathfrak{e}^\star; \\ \alpha & \text{if } x = y; \end{cases}$
- *the weight function is w^+ is such that $\forall \mathfrak{e} \in \mathfrak{E}$: $w^+(\phi(\mathfrak{e})) \triangleq w(\mathfrak{e})$.*

The application $\phi_{y^\alpha\text{-}v} : \mathfrak{H}_w \mapsto \mathfrak{H}^+_{w^+}$ is called the y^α ***-vertex-increasing operation****.*

Operation 3.3.5. *The* ***merged hb-graph*** $\widehat{\mathfrak{H}_{\widehat{w}}} \triangleq \left(\widehat{V}, \widehat{\mathfrak{E}}, \widehat{w}\right)$ *of a family* $(\mathfrak{H}_i)_{i\in I}$ *of weighted hb-graphs with* $\forall i \in I : \mathfrak{H}_i = (V_i, \mathfrak{E}_i, w_i)$ *is the weighted hb-graph where:*

- $\widehat{V} \triangleq \bigcup_{i\in I} V_i$;
- $\widehat{\mathfrak{E}} \triangleq \left(\psi\left(\mathfrak{e}\right)\right)_{\mathfrak{e}\in {}^{a}\sum_{i\in I} \mathfrak{E}_i}$ *where the map* $\psi : \sum_{i\in I} \mathfrak{E}_i \to \mathcal{M}\left(\widehat{V}\right)$ *such that for all* $\mathfrak{e} \in \sum_{i\in I} \mathfrak{E}_i$, $\psi\left(\mathfrak{e}\right) \in \mathcal{M}\left(\widehat{V}\right)$ *and is the multiset* $\left\{x^{m_{\psi(\mathfrak{e})}(x)} : x \in \widehat{V}\right\}$, *with* $m_{\psi(\mathfrak{e})}(x) \triangleq \begin{cases} m_{\mathfrak{e}}(x) & \text{if } x \in \mathfrak{e}^{\star}; \\ 0 & \text{otherwise}; \end{cases}$
- $\forall \mathfrak{e} \in \mathfrak{E}_i$, $\widehat{w}(\mathfrak{e}) \triangleq w_i(\mathfrak{e})$.

The application $\phi_m : (\mathfrak{H}_i)_{i\in I} \mapsto \widehat{\mathfrak{H}}$ *is called the* ***merging operation****.*

[a] $\sum_{i\in I} \mathfrak{E}_i$ is the family obtained with all elements of each family $\mathfrak{E}_i$.

Operation 3.3.6. *Decomposing a hb-graph* $\mathfrak{H} = (V, \mathfrak{E})$ *into a family of hb-graphs* $(\mathfrak{H}_i)_{i\in I}$, *where* $\mathfrak{H}_i = (V, \mathfrak{E}_i)$ *such that* $\mathfrak{H} = \bigoplus_{i\in I} \mathfrak{H}_i$ *is called a* ***decomposition operation*** $\phi_d : \mathfrak{H} \mapsto (\mathfrak{H}_i)_{i\in I}$.

The direct sum of two hb-graphs appears as a merging operation in one way and as a decomposition operation in the other. For a given hb-graph, different decomposition operations exist. Nonetheless, the decomposition in elementary hb-graph is unique as well as the decomposition in m-uniform hb-graphs representing the different levels of m-uniformity in that hb-graph.

We now focus on the preservation of e-adjacency through these different operations, as it is fundamental to ensure the soundness of the constructed hypermatrix.

Definition 3.4. *Let* $\mathfrak{H} = (V, \mathfrak{E})$ *and* $\mathfrak{H}' = (V', \mathfrak{E}')$ *be two hb-graphs.*

Let $\phi : \mathfrak{H} \mapsto \mathfrak{H}'$.

ϕ *is said* ***preserving e-adjacency*** *if vertices of* V' *that are e-adjacent in* $\mathfrak{H}'$ *and also in* V *are e-adjacent in* $\mathfrak{H}$.

ϕ *is said* ***preserving exactly e-adjacency*** *if vertices that are e-adjacent in* $\mathfrak{H}'$ *are e-adjacent in* $\mathfrak{H}$ *and reciprocally.*

We can extend these definitions to $\psi : (\mathfrak{H}_i)_{i\in I} \mapsto \mathfrak{H}'$.

Definition 3.5. *Let* $(\mathfrak{H}_i)_{i\in I}$ *be a family of hb-graphs with* $\forall i \in I$, $\mathfrak{H}_i = (V_i, \mathfrak{E}_i)$ *and* $\mathfrak{H}' = (V', \mathfrak{E}')$ *a hb-graph.*

Let consider $\psi : (\mathfrak{H}_i)_{i\in I} \mapsto \mathfrak{H}'$.

ψ *is said* ***preserving e-adjacency*** *if vertices that are e-adjacent in* $\mathfrak{H}'$, *restricted to the ones that belong to* $V = \bigcup_{i\in I} V_i$, *are e-adjacent vertices in exactly one of the* $\mathfrak{H}_i$, $i \in I$.

*ψ is said **preserving exactly e-adjacency** if vertices that are e-adjacent in $\mathfrak{H}'$ are e-adjacent in exactly one of the $\mathfrak{H}_i$, $i \in I$ and reciprocally.*

We can extend these definitions to $\nu : \mathfrak{H} \mapsto (\mathfrak{H}_i)_{i\in I}$.

Definition 3.6. *Let $(\mathfrak{H}_i)_{i\in I}$ be a family of hb-graphs with $\forall i \in I$, $\mathfrak{H}_i = (V_i, \mathfrak{E}_i)$ and $\mathfrak{H} = (V, \mathfrak{E})$ a hb-graph.*

Let consider $\nu : \mathfrak{H} \mapsto (\mathfrak{H}_i)_{i\in I}$.

*ν is said **preserving e-adjacency** if vertices that are e-adjacent in one of the $\mathfrak{H}_i, i \in I$, restricted to the ones that belong to V, are e-adjacent in $\mathfrak{H}$.*

*ν is said **preserving exactly e-adjacency** if vertices that are e-adjacent in one of the $\mathfrak{H}_i$, $i \in I$ are e-adjacent in $\mathfrak{H}$ and reciprocally.*

The two following claims have immediate proofs.

Claim 3.3. *Let $\mathfrak{H} = (V, \mathfrak{E})$ be a hb-graph.*

The canonical weighting operation, the c-dilatation operation, the merging operation and, the decomposition operation preserve exactly e-adjacency.

The y-complemented operation and the y^α-vertex-increasing operation preserve e-adjacency.

Claim 3.4. *The composition of two operations which preserves (respectively exactly) e-adjacency preserves (respectively exactly) e-adjacency.*

The composition of two operations where one preserves exactly e-adjacency and the other preserves e-adjacency preserves e-adjacency.

3.3.7. Processes involved for building the e-adjacency tensor

In a general natural hb-graph $\mathfrak{H}$, hb-edges do not have the same m-cardinality: the rank of the $\overline{k}$-adjacency tensor of the elementary hb-graph associated to each hb-edge depends on the m-cardinality of the hb-edge. As a consequence, the hb-graph polynomial is no more homogeneous. Nonetheless, techniques to homogenize such a polynomial are well known.

We introduce here the **hb-graph m-uniformisation process** (Hm-UP for short) which transforms a given hb-graph of m-rank $r_{\mathfrak{H}}$ into a $r_{\mathfrak{H}}$-m-uniform hb-graph written $\overline{\mathcal{H}}$: this uniformisation can be mapped to the homogenization of the attached polynomial of the original hb-graph, called the **polynomial homogenization process** (PHP).

The Hm-UP can be achieved by different means of filling the not-at-the-level hb-edges so they reach a m-rank of $r_{\mathfrak{H}}$:

- **straightforward m-uniformisation** levels directly all hb-edges by adding a Null vertex Y_1 with a multiplicity being the difference between the hb-graph m-rank and the hb-edge m-cardinality. It is achieved by considering the Y_1-complemented hb-graph of $\mathfrak{H}$.

- **silo m-uniformisation** processes each of the m-uniform sub-hb-graphs obtained by gathering all hb-edges of a given m-cardinality r in a single sub-hb-graph, which is then $Y_r^{r_{\mathfrak{H}}-r}$-vertex-increased. A single $r_{\mathfrak{H}}$-m-uniformized hb-graph is then obtained by merging them.
- **layered m-uniformisation** processes m-uniform sub-hb-graphs of increasing m-cardinality by successively adding a vertex and merging it to the sub-hb-graph of the above layer. The layered homogenization process applied to hypergraphs was explained with full details in Appendix Section E.1; it involves two-phase step iterations based on successive $\{Y_k^1\}$-vertex-increased hb-graphs and merging with the dilated weighted hb-graph of the next layer.

All the hb-graph m-uniformisation process and constructed tensors are illustrated on an example given in Section 3.3.12.

3.3.8. On the choice of the technical coefficient $c_{\mathfrak{e}_i}$

To comply to the expectations, the technical coefficient $c_{\mathfrak{e}_i}$ has to be chosen such that by using the elements of the e-adjacency hypermatrix $\boldsymbol{A} = (a_{i_1\dots i_r})_{i_1,\dots,i_r\in[\![n]\!]}$, one can retrieve:

1. the m-degree of the vertices: $\sum_{i_2,\dots,i_r\in[\![n]\!]} a_{ii_2\dots i_r} = \deg_m(v_i)$;
2. the number of hb-edges $|\mathfrak{E}|$.

Similarly to Appendix Section E.1, we consider a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ that we decompose in a family of r-m-uniform hb-graphs $(\mathfrak{H}_r)_{r\in[\![r_{\mathfrak{H}}]\!]}$.

We consider $\mathcal{R}$ the equivalency relation defined on the family of hb-edges $\mathfrak{E}$ of $\mathfrak{H}$:

$$\mathfrak{e}\mathcal{R}\mathfrak{e}' \Leftrightarrow \#_m\mathfrak{e} = \#_m\mathfrak{e}'.$$

$\mathfrak{E}/\mathcal{R}$ is the set of classes of hb-edges of same m-cardinality. The elements of $\mathfrak{E}/\mathcal{R}$ are the sets:

$$\mathfrak{E}_r = \{\mathfrak{e}\in\mathfrak{E} : \#_m\mathfrak{e} = r\}.$$

Considering $R = \{r : \mathfrak{E}_r \in \mathfrak{E}/\mathcal{R}\}$, we set $\mathfrak{E}_r = \emptyset$ for all $r \in [\![r_{\mathfrak{H}}]\!]\setminus R$.

For all $r \in [\![r_{\mathfrak{H}}]\!]$, $\mathfrak{H}_r = (V, \mathfrak{E}_r)$ is r-m-uniform.

It holds: $\mathfrak{E} = \bigcup_{r\in[\![r_{\mathfrak{H}}]\!]} \mathfrak{E}_r$ and $\mathfrak{E}_{r_1} \cap \mathfrak{E}_{r_2} = \emptyset$ for all $r_1 \neq r_2$, hence $(\mathfrak{E}_r)_{r\in[\![r_{\mathfrak{H}}]\!]}$ constitutes a partition of $\mathfrak{E}$ which is unique from the way it has been defined.

Hence:

$$\mathfrak{H} = \bigoplus_{r\in[\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_r.$$

Each of these r-m-uniform hb-graph $\mathfrak{H}_r$, where the $\overline{k}$-adjacency is achieved by r-adjacency, can be associated to a $\overline{k}$-adjacency tensor $\mathcal{A}_r$ viewed as a hypermatrix $\boldsymbol{A}_{\mathfrak{H}_r} = \left(a_{(r)i_1\dots i_r}\right)$ of order r, hyper-cubic and symmetric of dimension $|V| = n$.

We write: $\left(a_{i_1\dots i_{r_{\mathfrak{H}}}}\right)_{i_1,\dots,i_{r_{\mathfrak{H}}}\in[\![n_1]\!]}$ the e-adjacency hypermatrix associated to $\mathfrak{H}$ where $n_1 = n + n_A$, n_A corresponds to the number of different special vertices added to the hb-edges. n_A depends on the way the hypermatrix is built:

- $n_A = 1$ for the straightforward process;
- $n_A = r_{\mathfrak{H}} - 1$ for the silo process;
- $n_A = r_{\mathfrak{H}} - 1$ for the layered process.

For a given $r \in [\![r_{\mathfrak{H}}]\!]$, the number of hb-edges in $\mathfrak{H}_r$ is given by summing the elements of $\boldsymbol{A}_{\mathfrak{H}_r}$:

$$\begin{aligned}\sum_{i_1,\ldots,i_r\in[\![n]\!]} a_{(r)i_1\ldots i_r} &= \sum_{i=1}^{n}\sum_{i_2,\ldots,i_r\in[\![n]\!]} a_{(r)ii_2\ldots i_r}\\ &= \sum_{i=1}^{n}\deg_m(v_i)\\ &= r\,|\mathfrak{E}_r|\,.\end{aligned}$$

In the m-uniformized hb-graph of $\mathfrak{H}$, the number of hb-edges can also be calculated using:

$$\sum_{i_1,\ldots,i_{r_{\mathfrak{H}}}\in[\![n_1]\!]} a_{i_1\ldots i_{r_{\mathfrak{H}}}} = r_{\mathfrak{H}}\,|\mathfrak{E}|\,.$$

As:

$$\begin{aligned}|\mathfrak{E}| &= \sum_{r=1}^{r_{\mathfrak{H}}}|\mathfrak{E}_r|\\ &= \sum_{r=1}^{r_{\mathfrak{H}}}\frac{1}{r}\sum_{i_1,\ldots,i_r\in[\![n]\!]} a_{(r)i_1\ldots i_r},\end{aligned}$$

it follows:

$$\sum_{i_1,\ldots,i_{r_{\mathfrak{H}}}\in[\![n_1]\!]} a_{i_1\ldots i_{r_{\mathfrak{H}}}} = \sum_{r=1}^{r_{\mathfrak{H}}}\frac{r_{\mathfrak{H}}}{r}\sum_{i_1,\ldots,i_r\in[\![n]\!]} a_{(r)i_1\ldots i_r}.$$

Also, choosing for all $i \in [\![p]\!]$: $c_{\mathfrak{e}_i} = \dfrac{r_{\mathfrak{H}}}{r}$ where $r = \#_m\mathfrak{e}_i$, we write for all $r \in [\![r_{\mathfrak{H}}]\!]$:

$$c_r \triangleq \frac{r_{\mathfrak{H}}}{r}.$$

c_r is the technical coefficient for the corresponding layer of level r of the hb-graph $\mathfrak{H}$.

3.3.9. Straightforward approach

Straightforward m-uniformisation:

We first decompose $\mathfrak{H} = \bigoplus_{r\in[\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_r$ as seen in Section 3.3.8.

We then transform each $\mathfrak{H}_r$, $r \in [\![r_{\mathfrak{H}}]\!]$ into a canonical weighted hb-graph $\mathfrak{H}_{r,1}$ that is dilated with the help of the dilatation coefficient c_r to obtain the c_r-dilated hb-graph $\mathfrak{H}_{r,c_r}$.

This family $(\mathfrak{H}_{r,c_r})$ is then merged into the hb-graph:

$$\mathfrak{H}_{w,d} = \bigoplus_{r\in[\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_{r,c_r}.$$

To get a m-uniform hb-graph, we finally generate a vertex $Y_1 \notin V$ and apply to $\mathfrak{H}_{w,d}$ the Y_1-complemented operation to obtain $\mathfrak{H}_{Y_1^c,d} = \mathfrak{H}_{\text{str}}$ the Y_1-complemented hb-graph of $\mathfrak{H}_{w,d}$. $\mathfrak{H}_{\text{str}}$ is called the **straightforward m-uniformized hb-graph** of $\mathfrak{H}$.

The different steps are summarized in Figure 3.1.

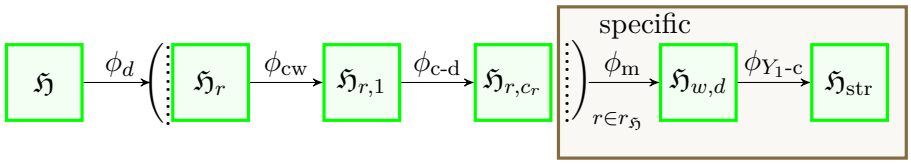


Figure 3.1.: Operations on the original hb-graph for its m-uniformization in the straightforward approach. Parenthesis with vertical dots indicate parallel operations.

Claim 3.5. *The transformation $\phi_s : \mathfrak{H} \mapsto \mathfrak{H}_{str}$ preserves e-adjacency.*

Proof. $\phi_{\text{s}} = \phi_{y_1\text{-c}} \circ \phi_{\text{m}} \circ \left(\vdots\ \phi_{\text{c-d}} \circ \phi_{\text{cw}}\ \vdots\right) \circ \phi_{\text{d}}$.[3]

All these operations either preserve e-adjacency or preserve exactly e-adjacency. Also, by composition, ϕ_{s} preserves e-adjacency.

□

Straightforward homogenization:

In order to homogenize the hb-graph polynomial, we introduce an additional variable y_1 that corresponds to the additional vertex Y_1 used during the Hm-UP.

The normalised $\overline{k}$-adjacency hypermatrix of the elementary hb-graph corresponding to the hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, \ldots, v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ is $\boldsymbol{Q}_{\mathfrak{e}_i}$ of rank $\rho_i = \#_m\mathfrak{e}_i$ and dimension n.

The corresponding reduced polynomial is:

$$P_{\mathfrak{e}_i}(\boldsymbol{z_0}) = \rho_i z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}}.$$

To transform this polynomial of degree ρ_i into a polynomial of degree $r_{\mathfrak{H}}$, we have to multiply the former by $y_1^{m_{i\,n+1}}$ where $m_{i\,n+1} = r_{\mathfrak{H}} - \rho_i$. This corresponds to adding the vertex Y_1 with multiplicity $m_{i\,n+1}$.

[3] $\left(\vdots\ \ \vdots\right)_{\ldots\ \in\ \ldots}$ indicates parallel operations on each member of the family as specified in index of the right parenthesis.

The term $P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ with the attached tensor $\mathcal{P}_{\mathfrak{e}_i}$ of rank ρ_i and dimension n is transformed in[4]:

$$R_{\mathfrak{e}_i}(\boldsymbol{z_1}) = P_{\mathfrak{e}_i}(\boldsymbol{z_0})\, y_1^{m_{i\,n+1}} = \rho_i z_{j_1}^{m_{ij_1}} \dots z_{j_{k_i}}^{m_{ij_{k_i}}} y_1^{m_{i\,n+1}}$$

with the attached tensor $\mathcal{R}_{\mathfrak{e}_i}$ of rank $r_{\mathfrak{H}}$ and dimension $n+1$.

The CHR of the tensor $\mathcal{R}_{\mathfrak{e}_i}$ is the hypermatrix $\boldsymbol{R_{\mathfrak{e}_i}} = \left(r_{i_1\dots i_{r_{\mathfrak{H}}}}\right)$. The elements that are non-zero in $\boldsymbol{R_{\mathfrak{e}_i}}$ have all the same value:

$$\rho_{\mathrm{str},\mathfrak{e}_i} = \rho_i \frac{m_{ij_1}! \dots m_{ij_{k_i}}! m_{i\,n+1}!}{r_{\mathfrak{H}}!}.$$

The indices of the non-zero elements of $\boldsymbol{R_{\mathfrak{e}_i}}$ are obtained by permutation of the elements of the multiset:

$$\left\{ j_1^{m_{ij_1}}, \dots, j_{k_i}^{m_{ij_{k_i}}}, (n+1)^{m_{i\,n+1}} \right\}.$$

The number of possible permutations is:

$$\frac{r_{\mathfrak{H}}!}{m_{ij_1}! \dots m_{ij_{k_i}}! m_{i\,n+1}!}.$$

The hb-graph polynomial $P(\boldsymbol{z_0}) = \sum\limits_{i\in[\![p]\!]} c_i P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ is transformed into a homogeneous polynomial:

$$R(\boldsymbol{z_1}) = \sum_{i\in[\![p]\!]} c_i R_{\mathfrak{e}_i}(\boldsymbol{z_1}) = \sum_{i\in[\![p]\!]} c_i z_{j_1}^{m_{ij_1}} \dots z_{j_{k_i}}^{m_{ij_{k_i}}} y_1^{m_{i\,n+1}}$$

representing the straightforward m-uniformized hb-graph $\mathfrak{H}_{\mathrm{str}}$ of $\mathfrak{H}$ with attached hypermatrix $\boldsymbol{R} = \sum\limits_{i=1}^{p} c_{\mathfrak{e}_i} \boldsymbol{R_{\mathfrak{e}_i}}$ where $c_{\mathfrak{e}_i} = \dfrac{r_{\mathfrak{H}}}{\rho_i} = \dfrac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_i}$. This provides a direct homogenization of the whole hb-graph polynomial.

Definition 3.7. *The **straightforward e-adjacency tensor** $\mathcal{A}_{str,\mathfrak{H}}$ **of a hb-graph** $\mathfrak{H} = (V, \mathfrak{E})$ is the tensor of CHR $\boldsymbol{A}_{str,\mathfrak{H}}$ defined by:*

$$\boldsymbol{A}_{str,\mathfrak{H}} \triangleq \sum_{i\in[\![p]\!]} c_{\mathfrak{e}_i} \boldsymbol{R_{\mathfrak{e}_i}}.$$

where for $\mathfrak{e}_i = \left\{ v_{j_1}^{m_{ij_1}}, \dots, v_{j_{k_i}}^{m_{ij_{k_i}}} \right\} \in \mathfrak{E}$:

$$c_{\mathfrak{e}_i} = \frac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_i}$$

is the dilatation coefficient and where $\boldsymbol{R_{\mathfrak{e}_i}} = \left(r_{i_1\dots i_{r_{\mathfrak{H}}}}\right)$ *is the hypermatrix whose elements have only two possible values, 0 and:*

$$\rho_{str,\mathfrak{e}_i} = \frac{m_{ij_1}! \dots m_{ij_{k_i}}! m_{i\,n+1}!}{r_{\mathfrak{H}}!} \#_m \mathfrak{e}_i$$

with $m_{i\,n+1} = r_{\mathfrak{H}} - \#_m \mathfrak{e}_i$. *The indices of the non-zero elements of* $\boldsymbol{R_{\mathfrak{e}_i}}$ *are obtained by permutation of the elements of the multiset:*

$$\left\{ j_1^{m_{ij_1}}, \dots, j_{k_i}^{m_{ij_{k_i}}}, (n+1)^{m_{i\,n+1}} \right\}.$$

[4] $\boldsymbol{z_k} = (z_1, ..., z_n, y_1, ..., y_k)$ for $k \in [\![r_{\mathfrak{H}} - 1]\!]$

Remark 3.3.1. *In practice, writing* $\boldsymbol{A}_{str,\mathfrak{H}} = \left(a_{l_1 \cdots l_{r_{\mathfrak{H}}}}\right)$, *the element of* $\boldsymbol{A}_{str,\mathfrak{H}}$ *of indices* $l_1, \cdots, l_{r_{\mathfrak{H}}}$ *such that:*

$$\{\{l_1, \cdots, l_{r_{\mathfrak{H}}}\}\} = \left\{j_1^{m_{ij_1}}, ..., j_{k_i}^{m_{ij_{k_i}}}, [n+1]^{m_{i\,n+1}}\right\},$$

corresponding to a hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, ..., v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ *of the original hb-graph* $\mathfrak{H}$, *is:*

$$a_{l_1 \ldots l_{r_{\mathfrak{H}}}} = \frac{m_{ij_1}! \ldots m_{ij_{k_i}}! m_{i\,n+1}!}{(r_{\mathfrak{H}} - 1)!}.$$

3.3.10. Silo approach

Silo m-uniformisation:

The first steps are similar to the straightforward approach.

The hb-graph $\mathfrak{H}$ is decomposed in layers $\mathfrak{H} = \bigoplus_{r \in [\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_r$ as described in Section 3.3.8. Each $\mathfrak{H}_r$, $r \in [\![r_{\mathfrak{H}}]\!]$ is canonically weighted and c_r-dilated to obtain $\mathfrak{H}_{r,c_r}$.

We generate $r_{\mathfrak{H}} - 1$ new vertices $Y_i \notin V$, $i \in [\![r_{\mathfrak{H}} - 1]\!]$.

We then apply to each $\mathfrak{H}_{r,c_r}$, $r \in [\![r_{\mathfrak{H}} - 1]\!]$ the $Y_r^{r_{\mathfrak{H}} - r}$-vertex-increasing operation to obtain $\mathfrak{H}^+_{r,c_r}$ the $Y_r^{r_{\mathfrak{H}} - r}$-complemented hb-graph for each $\mathfrak{H}_{r,c_r}$, $r \in [\![r_{\mathfrak{H}} - 1]\!]$. The family $\left(\mathfrak{H}^+_{r,c_r}\right)_{r \in r_{\mathfrak{H}}}$ is then merged using the merging operation to obtain the $r_{\mathfrak{H}}$-m-uniform hb-graph $\widehat{\mathfrak{H}_{\widehat{w}}} = \mathfrak{H}_{\text{sil}}$. $\mathfrak{H}_{\text{sil}}$ is called the **silo m-uniformized hb-graph** of $\mathfrak{H}$.

The different steps are summarized in Figure 3.2.

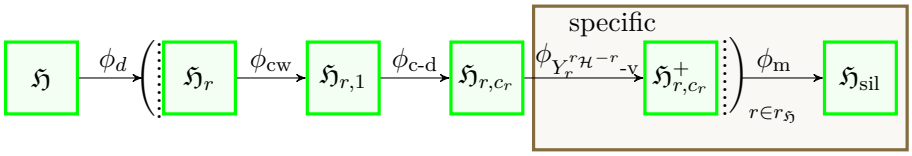


Figure 3.2.: Operations on the original hb-graph for its m-uniformisation in the silo approach. Parenthesis with vertical dots indicate parallel operations.

Claim 3.6. *The transformation* $\phi_s : \mathfrak{H} \mapsto \mathfrak{H}_{sil}$ *preserves e-adjacency.*

Proof. $\phi_{\text{s}} = \phi_{\text{m}} \circ \left(\vdots\ \phi_{y_r^{r_{\mathfrak{H}} - r}\text{-v}} \circ \phi_{\text{c-d}} \circ \phi_{\text{cw}}\ \vdots\right) \circ \phi_{\text{d}}$.

The operations involved in ϕ_s either preserve e-adjacency or preserve exactly e-adjacency: also, by composition, ϕ_{s} preserves e-adjacency.

□

Silo homogenization:

In this homogenization process, we suppose that the hb-edges are sorted by m-cardinality.

During the silo uniformisation, we added $r_{\mathfrak{H}}-1$ vertices Y_1 to $Y_{r_{\mathfrak{H}}-1}$ into the universe, i.e. the vertex set. These vertices correspond to $r_{\mathfrak{H}}-1$ additional variables respectively y_1 to $y_{r_{\mathfrak{H}}-1}$ that we introduced to homogenize the hb-graph polynomial.

The normalised $\overline{k}$-adjacency hypermatrix of the elementary hb-graph corresponding to the hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, \ldots, v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ is $\boldsymbol{Q}_{\mathfrak{e}_i}$ of rank $\rho_i = \#_m \mathfrak{e}_i$ and dimension n.

The corresponding reduced polynomial is:

$$P_{\mathfrak{e}_i}(\boldsymbol{z_0}) = \rho_i z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}}$$

attached to tensor $\mathcal{P}_{\mathfrak{e}_i}$ has for degree the m-cardinality of the hb-edge $\mathfrak{e}_i$, i.e. $\#_m \mathfrak{e}_i$. To transform it into a polynomial of degree $r_{\mathfrak{H}}$, we use the additional variable $y_{\#_m \mathfrak{e}_i}$ with multiplicity $m_{i\,\#_m \mathfrak{e}_i} = r_{\mathfrak{H}} - \#_m \mathfrak{e}_i$.

The term $P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ attached to tensor $\mathcal{P}_{\mathfrak{e}_i}$ of rank $\#_m \mathfrak{e}_i$ and dimension n is transformed in:

$$R_{\mathfrak{e}_i}(\boldsymbol{z}_{\boldsymbol{\#_m \mathfrak{e}_i}}) = P_{\mathfrak{e}_i}(\boldsymbol{z_0})\, y_{\#_m \mathfrak{e}_i}^{m_{i\,n+\#_m \mathfrak{e}_i}}$$

with attached tensor $\mathcal{R}_{\mathfrak{e}_i}$ of rank $r_{\mathfrak{H}}$ and dimension $n+1$.

The CHR of the tensor $\mathcal{R}_{\mathfrak{e}_i}$ is the hypermatrix $\boldsymbol{R}_{\mathfrak{e}_i} = \left(r_{i_1 \ldots i_{r_{\mathfrak{H}}}}\right)$. The elements of $\boldsymbol{R}_{\mathfrak{e}_i}$ that are non-zero elements have all the same value:

$$\rho_{\text{sil},\mathfrak{e}_i} = \rho_i \frac{m_{ij_1}! \ldots m_{ij_{k_i}}! m_{i\,n+\#_m \mathfrak{e}_i}!}{r_{\mathfrak{H}}!}$$

with $m_{i\,n+\#_m \mathfrak{e}_i} = r_{\mathfrak{H}} - \#_m \mathfrak{e}_i$. The indices of the non-zero elements of $\boldsymbol{R}_{\mathfrak{e}_i}$ are obtained by permutation of the multiset:

$$\left\{j_1^{m_{ij_1}}, \ldots, j_{k_i}^{m_{ij_{k_i}}}, [n+\#_m \mathfrak{e}_i]^{m_{i\,n+\#_m \mathfrak{e}_i}}\right\}.$$

P is transformed into a homogeneous polynomial:

$$R(\boldsymbol{z}_{\boldsymbol{r_{\mathfrak{H}}-1}}) = \sum_{i \in [\![p]\!]} c_i R_{\mathfrak{e}_i}(\boldsymbol{z}_{\boldsymbol{\#_m \mathfrak{e}_i}}) = \sum_{i \in [\![p]\!]} c_i z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}} y_{\#_m \mathfrak{e}_i}^{m_{i\,n+\#_m \mathfrak{e}_i}}$$

representing the silo m-uniformized hb-graph $\mathfrak{H}_{\text{sil}}$ of $\mathfrak{H}$ with attached hypermatrix $\boldsymbol{R} = \sum\limits_{i \in [\![p]\!]} c_{\mathfrak{e}_i} \boldsymbol{R}_{\mathfrak{e}_i}$ where:

$$c_{\mathfrak{e}_i} = \frac{r_{\mathfrak{H}}}{\rho_i} = \frac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_i}.$$

Definition 3.8. *The **silo e-adjacency tensor** $\mathcal{A}_{sil,\mathfrak{H}}$ **of a hb-graph** $\mathfrak{H} = (V, \mathfrak{E})$ is the tensor of CHR* $\boldsymbol{A}_{sil,\mathfrak{H}} \triangleq \left(a_{i_1 \ldots i_{r_{\mathfrak{H}}}}\right)_{i_1, \ldots, i_{r_{\mathfrak{H}}} \in [\![n]\!]}$ *defined by:*

$$\boldsymbol{A}_{sil,\mathfrak{H}} \triangleq \sum_{i \in [\![p]\!]} c_{\mathfrak{e}_i} \boldsymbol{R}_{\mathfrak{e}_i}$$

and where for $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, \ldots, v_{j_{k_i}}^{m_{ij_{k_i}}}\right\} \in \mathfrak{E}$:

$$c_{\mathfrak{e}_i} = \frac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_i}$$

is the dilatation coefficient and $\boldsymbol{R}_{\mathfrak{e}_i} = \left(r_{i_1 \ldots i_{r_{\mathfrak{H}}}}\right)$ *is the hypermatrix whose elements have only two possible values, 0 and:*

$$\rho_{sil,\mathfrak{e}_i} = \frac{m_{ij_1}! \ldots m_{ij_{k_i}}! m_{i\,n+\#_m \mathfrak{e}_i}!}{r_{\mathfrak{H}}!} \#_m \mathfrak{e}_i$$

with $m_{i\,n+\#_m \mathfrak{e}_i} = r_{\mathfrak{H}} - \#_m \mathfrak{e}_i$.

The indices of the non-zero elements of $\boldsymbol{R}_{\mathfrak{e}_i}$ *are obtained by permutation of the elements of the multiset:*

$$\left\{j_1^{m_{ij_1}}, \ldots j_{k_i}^{m_{ij_{k_i}}}, [n + \#_m \mathfrak{e}_i]^{m_{i\,n+\#_m \mathfrak{e}_i}}\right\}.$$

Remark 3.3.2. *In this case:*

$$\boldsymbol{A}_{sil,\mathfrak{H}} = \sum_{r \in [\![r_{\mathfrak{H}}]\!]} c_r \sum_{\mathfrak{e}_i \in \{\mathfrak{e}: \#_m \mathfrak{e} = r\}} \boldsymbol{R}_{\mathfrak{e}_i}$$

where $c_r = \dfrac{r_{\mathfrak{H}}}{r}$.

Remark 3.3.3. *In practice, writing* $\boldsymbol{A}_{sil,\mathfrak{H}} = \left(a_{l_1 \cdots l_{r_{\mathfrak{H}}}}\right)$, *the element of* $\boldsymbol{A}_{sil,\mathfrak{H}}$ *of indices* $l_1, \cdots, l_{r_{\mathfrak{H}}}$ *such that:*

$$\{\{l_1, ..., l_{r_{\mathfrak{H}}}\}\} = \left\{j_1^{m_{ij_1}}, ..., j_{k_i}^{m_{ij_{k_i}}}, [n + \#_m \mathfrak{e}_i]^{m_{i\,n+\#_m \mathfrak{e}_i}}\right\},$$

corresponding to a hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, ..., v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ *of the original hb-graph* $\mathfrak{H}$, *is:*

$$a_{l_1 \ldots l_{r_{\mathfrak{H}}}} = \frac{m_{ij_1}! \ldots m_{ij_{k_i}}! m_{i\,n+\#_m \mathfrak{e}_i}!}{(r_{\mathfrak{H}} - 1)!}.$$

3.3.11. Layered approach

Layered uniformisation:

The first steps are similar to the straightforward approach.

The hb-graph $\mathfrak{H}$ is decomposed into layers $\mathfrak{H} = \bigoplus_{r \in [\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_r$ as described in Section 3.3.8. Each $\mathfrak{H}_r$, $r \in [\![r_{\mathfrak{H}}]\!]$ is canonically weighted and c_r-dilated to obtain $\mathfrak{H}_{r,c_r}$.

We generate $r_{\mathfrak{H}} - 1$ additional different vertices $Y_i \notin V$, $i \in [\![r_{\mathfrak{H}} - 1]\!]$ and write $V_s = \{Y_i : i \in [\![r_{\mathfrak{H}} - 1]\!]\}$.

A two-phase steps iteration is considered: the inflation phase (IP) and the merging phase (MP). At step $k = 0$, $\mathcal{K}_0 = \mathfrak{H}_{1,c_1}$ and no further action is made but increasing k by 1. At step $k > 0$, the input is the k-m-uniform weighted hb-graph $\mathcal{K}_k$ obtained from

the previous iteration. In the IP, $\mathcal{K}_k$ is transformed into $\mathcal{K}_k^+$ the Y_k^1-vertex-increased hb-graph, which is $(k+1)$-m-uniform.

The MP merges the hypergraphs $\mathcal{K}_k^+$ and $\mathfrak{H}_{k+1,c_{k+1}}$ into a single $(k+1)$-m-uniform hb-graph $\widehat{\mathcal{K}_{\widehat{w}}}$.

We iterate by increasing k by 1, while $k < r_{\mathfrak{H}}$. When k reaches $r_{\mathfrak{H}}$, we stop iterating and the last $\widehat{\mathcal{K}_{\widehat{w}}}$ obtained, written $\mathfrak{H}_{\text{lay}}$ is called the layered m-uniformized hb-graph of $\mathfrak{H}$.

The different steps are summarized in Figure 3.3.

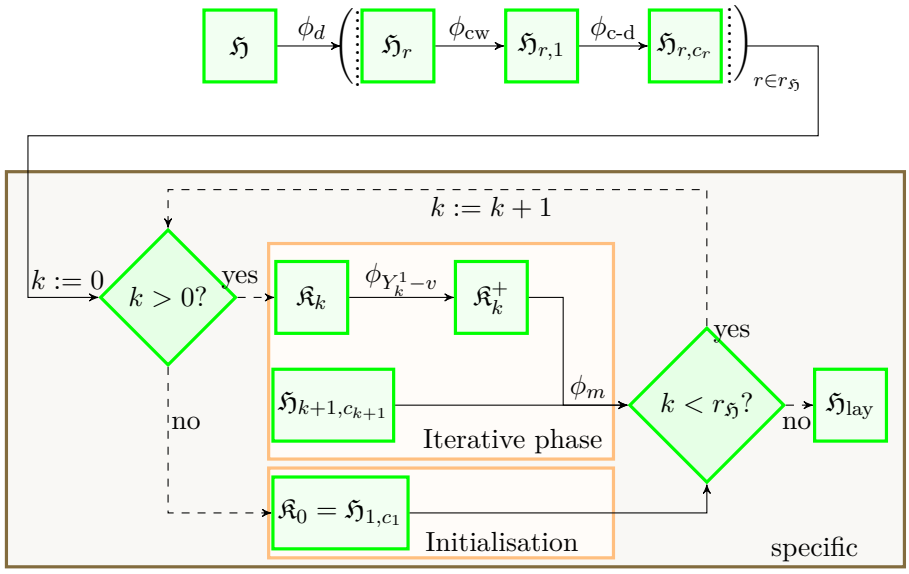


Figure 3.3.: Operations on the original hb-graph for its m-uniformisation in the layered approach. Parenthesis with vertical dots indicate parallel operations.

Claim 3.7. *The transformation $\phi_s : \mathfrak{H} \mapsto \mathfrak{H}_{lay}$ preserves e-adjacency.*

Proof. $\phi_{\text{s}} = \psi \circ \left(\vdots \phi_{\text{c-d}} \circ \phi_{\text{cw}} \vdots \right) \circ \phi_{\text{d}}$, where ψ is called the iterative layered operation that converts the family obtained by $\left(\vdots \phi_{\text{c-d}} \circ \phi_{\text{cw}} \vdots \right) \circ \phi_{\text{d}}$ and transforms it into the V_S-layered m-uniform hb-graph of $\mathfrak{H}$.

All the operations $\phi_{\text{c-d}}$, $\phi_{\text{c-w}}$ and ϕ_{d} either preserve e-adjacency or preserve exactly e-adjacency, and so forth by composition.

The iterative layered operation preserves e-adjacency as the operations involved are individually preserving e-adjacency and that the family of hb-graphs at the input contains hb-edge families that are totally distinct.

Also, by composition, ϕ_{s} preserves e-adjacency.

□

Layered homogenization:

The idea is to sort the hb-edges as in the silo homogenization and consider also $r_{\mathfrak{H}} - 1$ additional vertices Y_1 to $Y_{r_{\mathfrak{H}}-1}$ into the universe, corresponding to $r_{\mathfrak{H}} - 1$ additional variables respectively y_1 to $y_{r_{\mathfrak{H}}-1}$.

But in this case, these vertices are added successively to each hb-edge to fill them so they reach all the same m-cardinality $r_{\mathfrak{H}}$: a hb-edge of initial cardinality $\#_m\mathfrak{e}$ will be filled with elements $Y_{\#_m\mathfrak{e}}$ to $Y_{r_{\mathfrak{H}}-1}$. This corresponds to adding the k-m-uniform sub-hb-graph $\mathfrak{H}_k$ with the $k+1$-m-uniform sub-hb-graph $\mathfrak{H}_{k+1}$ by filling the hb-edge of $\mathfrak{H}_k$ with the additional vertex Y_k to get a homogenized $k+1$-m-uniform sub-hb-graph of the homogenized hb-graph $\overline{\mathfrak{H}}$.

The normalised $\overline{k}$-adjacency hypermatrix of the elementary hb-graph corresponding to the hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, \ldots, v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ is $\boldsymbol{Q}_{\mathfrak{e}_i}$ of rank $\rho_i = \#_m\mathfrak{e}_i$ and dimension n. The corresponding reduced polynomial is:

$$P_{\mathfrak{e}_i}(\boldsymbol{z_0}) = \rho_i z_{j_1}^{m_{ij_1}} \ldots z_{j_{k_i}}^{m_{ij_{k_i}}}$$

of degree $\#_m\mathfrak{e}_i$.

All the hb-edges of same m-cardinality belongs to the same layer. In order to transform all the hb-edges of m-cardinality $\#_m\mathfrak{e}_i$ into hb-edges of m-cardinality $\#_m\mathfrak{e}_i + 1$, we fill them with the corresponding element $Y_{\#_m\mathfrak{e}_i}$.

In this case, the polynomial $P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ is transformed into:

$$R_{(1)\mathfrak{e}_i}(\boldsymbol{z_{\#_m\mathfrak{e}_i}}) = P_{\mathfrak{e}_i}(\boldsymbol{z_0})\, y^1_{\#_m\mathfrak{e}_i}$$

of degree $\#_m\mathfrak{e}_i + 1$.

Iterating over the layers, the original hb-edge $\mathfrak{e}_i$ is successively filled with additional vertices Y_i, with $i \in [\![\#_m\mathfrak{e}_i; r_{\mathfrak{H}} - 1]\!]$, and the polynomial $P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ is transformed into the polynomial:

$$R_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}(\boldsymbol{z_{r_{\mathfrak{H}}-1}}) = P_{\mathfrak{e}_i}(\boldsymbol{z_0})\, y^1_{\#_m\mathfrak{e}_i} \ldots y^1_{r_{\mathfrak{H}}-1}$$

of degree $r_{\mathfrak{H}}$.

The polynomial $P_{\mathfrak{e}_i}(\boldsymbol{z_0})$ with attached tensor $\mathcal{P}_{\mathfrak{e}_i}$ of rank $\#_m\mathfrak{e}_i$ and dimension n is transformed in:

$$R_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}(\boldsymbol{z_{r_{\mathfrak{H}}-1}}) = R_{\mathfrak{e}_i}(\boldsymbol{z_0})\, y^1_{\#_m\mathfrak{e}_i} \ldots y^1_{r_{\mathfrak{H}}-1}$$

with attached tensor $\mathcal{R}_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}$ of rank $r_{\mathfrak{H}}$ and dimension $n + r_{\mathfrak{H}} - 1$.

The CHR of the tensor $\mathcal{R}_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}$ is the hypermatrix:

$$\boldsymbol{R_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}} = \left(r_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\, i_1 \ldots i_{r_{\mathfrak{H}}}}\right).$$

The elements of $\boldsymbol{R_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}}$ have only two possible values, 0 and:

$$\rho_{\text{lay},(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i} = \rho_i \frac{m_{ij_1}! \ldots m_{ij_{k_i}}!}{r_{\mathfrak{H}}!}.$$

The indices of the non-zero elements of $\boldsymbol{R}_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}$ are obtained by permutation of the elements of the multiset:

$$\left\{j_1^{m_{ij_1}},\ldots,j_{k_i}^{m_{ij_{k_i}}},[n+\#_m\mathfrak{e}_i]^1,\ldots,[n+r_{\mathfrak{H}}-1]^1\right\}.$$

And P is transformed into a homogeneous polynomial:

$$\begin{aligned} R\left(\boldsymbol{z}_{\boldsymbol{r_{\mathfrak{H}}-1}}\right) &= \sum_{i\in[\![p]\!]} c_i R_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}\left(\boldsymbol{z}_{\boldsymbol{r_{\mathfrak{H}}-1}}\right) \\ &= \sum_{i\in[\![p]\!]} c_i\rho_i z_{j_1}^{m_{ij_1}}\ldots z_{j_{k_i}}^{m_{ij_{k_i}}} y^1_{\#_m\mathfrak{e}_i}\ldots y^1_{r_{\mathfrak{H}}-1} \end{aligned}$$

representing the layered m-uniformized hb-graph $\mathfrak{H}_{\text{lay}}$ with attached hypermatrix:

$$\boldsymbol{R}=\sum_{i\in[\![p]\!]} c_{\mathfrak{e}_i}\boldsymbol{R}_{(\boldsymbol{r_{\mathfrak{H}}-\#_m\mathfrak{e}_i})\mathfrak{e}_i},$$

where:

$$c_{\mathfrak{e}_i}=\frac{r_{\mathfrak{H}}}{\rho_i}=\frac{r_{\mathfrak{H}}}{\#_m\mathfrak{e}_i}.$$

Definition 3.9. *The **layered e-adjacency tensor** $\mathcal{A}_{lay,\mathfrak{H}}$ **of a hb-graph** $\mathfrak{H}=(V,\mathfrak{E})$ is the tensor of CHR $\boldsymbol{A}_{lay,\mathfrak{H}}\triangleq\left(a_{i_1\ldots i_{r_{\mathfrak{H}}}}\right)_{i_1,\ldots,i_{r_{\mathfrak{H}}}\in[\![n]\!]}$ defined by:*

$$\boldsymbol{A}_{lay,\mathfrak{H}}\triangleq\sum_{i\in[\![p]\!]} c_{\mathfrak{e}_i}\boldsymbol{R}_{(\boldsymbol{r_{\mathfrak{H}}-\#_m\mathfrak{e}_i})\mathfrak{e}_i}$$

where for $\mathfrak{e}_i=\left\{v_{j_1}^{m_{ij_1}},\ldots,v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}\in\mathfrak{E}$:

$$c_{\mathfrak{e}_i}=\frac{r_{\mathfrak{H}}}{\#_m\mathfrak{e}_i}$$

is the dilatation coefficient and $\boldsymbol{R}_{(\boldsymbol{r_{\mathfrak{H}}-\#_m\mathfrak{e}_i})\mathfrak{e}_i}=\left(r_{(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)i_1\ldots i_{r_{\mathfrak{H}}}}\right)$ is the hypermatrix whose elements have only two possible values 0 and:

$$\rho_{lay,(r_{\mathfrak{H}}-\#_m\mathfrak{e}_i)\mathfrak{e}_i}=\frac{m_{ij_1}!\ldots m_{ij_{k_i}}!}{r_{\mathfrak{H}}!}\#_m\mathfrak{e}_i.$$

The indices of the non-zero elements of $\boldsymbol{R}_{(\boldsymbol{r_{\mathfrak{H}}-\#_m\mathfrak{e}_i})\mathfrak{e}_i}$ are obtained by permutation of the elements of the multiset:

$$\left\{j_1^{m_{ij_1}},\ldots,j_{k_i}^{m_{ij_{k_i}}},[n+\#_m\mathfrak{e}_i]^1,\ldots,[n+r_{\mathfrak{H}}-1]^1\right\}.$$

Remark 3.3.4. *$\boldsymbol{A}_{lay,\mathfrak{H}}$ can also be written:*

$$\boldsymbol{A}_{lay,\mathfrak{H}}=\sum_{r\in[\![r_{\mathfrak{H}}]\!]} c_r\sum_{\mathfrak{e}_i\in\{\mathfrak{e}:\#_m\mathfrak{e}=r\}}\boldsymbol{R}_{\boldsymbol{\mathfrak{e}_i}},$$

where $c_r=\dfrac{r_{\mathfrak{H}}}{r}$.

Remark 3.3.5. *In practice, writing* $\boldsymbol{A}_{lay,\mathfrak{H}} = \left(a_{l_1\dots l_{r_{\mathfrak{H}}}}\right)$, *the element of* $\boldsymbol{A}_{lay,\mathfrak{H}}$ *of indices* $l_1, \cdots, l_{r_{\mathfrak{H}}}$ *such that:*

$$\{\{l_1, \cdots, l_{r_{\mathfrak{H}}}\}\} = \left\{j_1^{m_{ij_1}}, ..., j_{k_i}^{m_{ij_{k_i}}}, [n + \#_m \mathfrak{e}_i]^1, \dots, [n + r_{\mathfrak{H}} - 1]^1\right\},$$

corresponding to a hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}}, ..., v_{j_{k_i}}^{m_{ij_{k_i}}}\right\}$ *of the original hb-graph* $\mathfrak{H}$, *is:*

$$a_{l_1\dots l_{r_{\mathfrak{H}}}} = \frac{m_{ij_1}! \dots m_{ij_{k_i}}!}{(r_{\mathfrak{H}} - 1)!}.$$

3.3.12. Examples related to e-adjacency hypermatrices for general hb-graphs

We illustrate the different construction of e-adjacency hypermatrices for general hb-graphs by a common example.

Example 3.3.1. *Given the following hb-graph:* $\mathfrak{H} = (V, \mathfrak{E})$ *where:* $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ *and* $\mathfrak{E} = \{\mathfrak{e}_1, \mathfrak{e}_2, \mathfrak{e}_3, \mathfrak{e}_4\}$ *with:* $\mathfrak{e}_1 = \{v_1^2, v_2^2, v_3\}$, $\mathfrak{e}_2 = \{v_1, v_4, v_5, v_6\}$, $\mathfrak{e}_3 = \{v_1, v_4^2\}$ *and* $\mathfrak{e}_4 = \{v_1\}$.

The hb-graph given in Example 3.3.1 is represented in Figure 1.1.

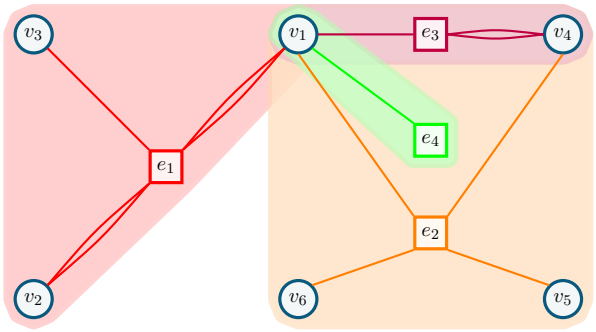


Figure 3.4.: Example of hb-graph for m-uniformisation principle

We show on this hb-graph how the filling is achieved depending on the filling option chosen. We have ordered the hb-edges in the order of increasing m-cardinality from bottom to top.

3.3.12.1. Layered filling option

In the layered filling approach, the different layers are filled with one or several vertices in order to reach the m-cardinality level of the m-uniform hb-graph next layer, in order to merge the filled layer and the reached level of m-uniformity layer. The special vertices are all different and written y_1 to y_4 as it is shown in Figure 3.5.

As the m-rank of the hb-graph is 5, the layered e-adjacency hypermatrix is of rank 5. There are 6 vertices and 4 additional special vertices, hence the dimension of the

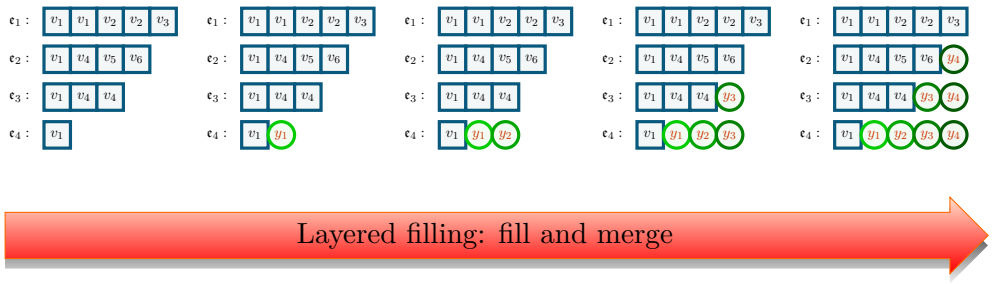


Figure 3.5.: Layered filling on the example of Figure 3.4

hypermatrix is 10, the place 1 to 6 storing the information for v_1 to v_6 in this order and the places 7 to 10 the information for y_1 to y_4. We write the layered e-adjacency hypermatrix $\boldsymbol{A}_{\text{lay}} = (a_{\text{lay}\, i_1 i_2 i_3 i_4 i_5})_{\forall j \in [\![5]\!] : i_j \in [\![10]\!]}$.

Considering all permutations of the indices $\sigma \in \mathcal{S}_5$, we have:

- $\mathfrak{e}_1$ stored in all the elements: $a_{\text{lay}\,\sigma(1\,1\,2\,2\,3)}$ of value $\dfrac{1}{6}$.
- $\mathfrak{e}_2$ stored in all the elements: $a_{\text{lay}\,\sigma(1\,4\,5\,6\,10)}$ of value $\dfrac{1}{24}$.
- $\mathfrak{e}_3$ stored in all the elements: $a_{\text{lay}\,\sigma(1\,4\,4\,9\,10)}$ of value $\dfrac{1}{12}$.
- $\mathfrak{e}_4$ stored in all the elements: $a_{\text{lay}\,\sigma(1\,7\,7\,7\,7)}$ of value 1.
- The other elements of $\boldsymbol{A}_{\text{lay}}$ are equal to zero.

3.3.12.2. Silo filling option

In the silo approach, the different layers of m-uniformity are filled up to the m-range of the hb-graph with different special vertices that depend only on the layer of m-uniformity, as it is shown in Figure 3.6.

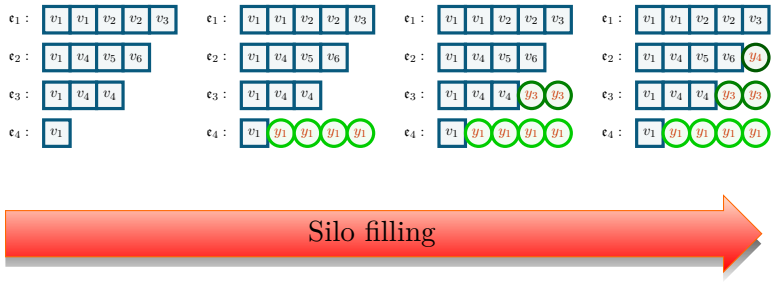


Figure 3.6.: Silo filling on the example of Figure 3.4

As the m-rank of the hb-graph is 5, the silo e-adjacency hypermatrix is of rank 5. There are 6 vertices and 4 additional special vertices, hence the dimension of the hypermatrix

is 10, the place 1 to 6 storing the information for v_1 to v_6 in this order and the places 7 to 10 the information for y_1 to y_4. We write the layered e-adjacency hypermatrix $\boldsymbol{A}_{\text{sil}} = (a_{\text{sil}\, i_1 i_2 i_3 i_4 i_5})_{\forall j \in [\![5]\!] : i_j \in [\![10]\!]}$.

Considering all permutations of the indices $\sigma \in \mathcal{S}_5$, we have:

- $\mathfrak{e}_1$ stored in all the elements: $a_{\text{sil}\,\sigma(1\,1\,2\,2\,3)}$ of value $\dfrac{1}{6}$.
- $\mathfrak{e}_2$ stored in all the elements: $a_{\text{sil}\,\sigma(1\,4\,5\,6\,10)}$ of value $\dfrac{1}{24}$.
- $\mathfrak{e}_3$ stored in all the elements: $a_{\text{sil}\,\sigma(1\,4\,4\,9\,9)}$ of value $\dfrac{1}{6}$.
- $\mathfrak{e}_4$ stored in all the elements: $a_{\text{sil}\,\sigma(1\,7\,7\,7\,7)}$ of value 1.
- The other elements of $\boldsymbol{A}_{\text{sil}}$ are equal to zero.

3.3.12.3. Straightforward filling option

In the straightforward approach, all the hb-edges are filled up to reaching a m-cardinality corresponding to the hb-graph m-range as it is shown in Figure 3.7.

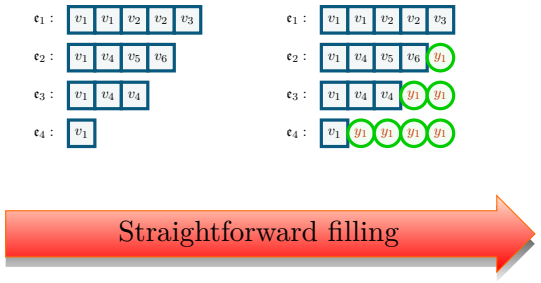


Figure 3.7.: Straightforward filling on the example of Figure 3.4

As the m-rank of the hb-graph is 5, the straightforward e-adjacency hypermatrix is of rank 5. There are 6 vertices and 1 additional special vertex, hence the dimension of the hypermatrix is 7, the place 1 to 6 storing the information for v_1 to v_6 in this order and the place 7 to y_1. We write the layered e-adjacency hypermatrix $\boldsymbol{A}_{\text{str}} = (a_{\text{str}\, i_1 i_2 i_3 i_4 i_5})_{\forall j \in [\![5]\!] : i_j \in [\![7]\!]}$.

Considering all permutations of the indices $\sigma \in \mathcal{S}_5$, we have:

- $\mathfrak{e}_1$ stored in all the elements: $a_{\text{str}\,\sigma(1\,1\,2\,2\,3)}$ of value $\dfrac{1}{6}$.
- $\mathfrak{e}_2$ stored in all the elements: $a_{\text{str}\,\sigma(1\,4\,5\,6\,7)}$ of value $\dfrac{1}{24}$.
- $\mathfrak{e}_3$ stored in all the elements: $a_{\text{str}\,\sigma(1\,4\,4\,7\,7)}$ of value $\dfrac{1}{6}$.
- $\mathfrak{e}_4$ stored in all the elements: $a_{\text{str}\,\sigma(1\,7\,7\,7\,7)}$ of value 1.
- The other elements of $\boldsymbol{A}_{\text{str}}$ are equal to zero.

3.4. Results on the constructed tensors

We recall that each of the tensors obtained is of rank $r_{\mathfrak{H}}$ and of dimension $n+n_{\mathcal{A}}$ where $n_{\mathcal{A}}$ is:

- in the straightforward approach: $n_{\mathcal{A}}=1$;
- in the silo approach: $n_{\mathcal{A}}=r_{\mathfrak{H}}-1$;
- and, in the layered approach: $n_A=r_{\mathfrak{H}}-1$.

3.4.1. Fulfillment of the expectations

We revisit the expectations formulated in Section 3.3.1 and prove that they are all met.

Guarantee 3.1. *The e-adjacency tensor should be non-negative, symmetric and its generation should be as simple as possible.*

Proof. The tensors that have been built are all non-negative and symmetric. Their generation mostly depends on the content of the hb-edges. Only the layered approach has a more complex generation process; however, it is retained, as it is the only possible approach that keeps hypergraph interpretability via Hm-UP without needing hb-graphs for general hypergraphs. □

Guarantee 3.2. *The tensor should be globally invariant to vertex permutation in the original hb-graph.*

Proof. Let $\mathfrak{H}=(V,\mathfrak{E})$ be a hb-graph with vertex set $V=\{v_i : i\in[\![n]\!]\}$ and $\mathfrak{E}=(\mathfrak{e}_j)_{j\in[\![p]\!]}$. We do the proof only for the straightforward tensor, since the other proofs are similar.

Let consider a permutation $\pi\in\mathcal{S}_n$, that corresponds to a relabeling of the vertices from the universe of the original hb-graph. Applying this relabeling to the content of the hb-edge:

$$\mathfrak{e}_j=\left\{v_{i_1}^{m_{j\,i_1}},...,v_{i_j}^{m_{j\,i_j}}\right\}$$

transforms it into:

$$\mathfrak{e}_j=\left\{v_{\pi(i_1)}^{m_{j\,i_1}},...,v_{\pi(i_j)}^{m_{j\,i_j}}\right\}.$$

The original hb-edge was stored in the elements of the form:

$$a_{i_1^{m_{j\,i_1}}...i_j^{m_{j\,i_j}}(n+1)^{r_{\mathfrak{H}}-\#_m\mathfrak{e}_j}}$$

and the ones obtained by permutation of the indices.

It follows that the relabeled hb-graph will have its elements stored in:

$$a_{\pi(i_1)^{m_{j\,i_1}}...\pi(i_j)^{m_{j\,i_j}}(n+1)^{r_{\mathfrak{H}}-\#_m\mathfrak{e}_j}}$$

and all the ones obtained by permutation of the elements of:

$$\left\{\pi\left(i_1\right)^{m_{j\,i_1}},...,\pi\left(i_j\right)^{m_{j\,i_j}},(n+1)^{r_{\mathfrak{H}}-\#_m\mathfrak{e}_j}\right\}$$

This corresponds to a permutation of the elements in the tensor according to the relabeling: the two tensors differ only by a reshuffling of the indices due to the permutation π, which means that the constructed tensor is globally invariant to a relabeling in the original hb-graph.

□

Guarantee 3.3. *The e-adjacency tensor should induce a unique reconstruction of the hb-graph it is originated from.*

Proof. From the way the elements have been constructed, there is a one-to-one mapping between the hb-edge and the ordered indices of the coefficient in the tensor. From these indices, the hb-graph can be reconstructed with no ambiguity.

□

Guarantee 3.4. *Given the choice of two representations the one that can be described with the least possible number of elements should be chosen. Then the sparsest e-adjacency tensor should be chosen.*

Proof. The straightforward tensor requires only one additional element to capture the information. The three representations are anyway economic as only one element each time needs to be described in the hypermatrix for a given hb-edge, the others being obtained by permutation of its indices.

□

Guarantee 3.5. *The e-adjacency tensor should allow direct retrieval of the vertex degrees.*

Proof. The proof will be given in the next Section.

□

3.4.2. Information on hb-graph

3.4.2.1. m-degree of vertices

We built the different tensors such that the retrieval of the vertex m-degree is possible; the null vertex(-ices) added during the Hm-UP give(s) additional information on the structure of the hb-graph.

Claim 3.8. *Let us consider for $j \in [\![n]\!]$, a vertex $v_j \in V$.*

Then for each of the e-adjacency tensors built, it holds:

$$\sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n+n_{\mathcal{A}}]\!]} a_{j j_2 \dots j_{r_{\mathfrak{H}}}} = \sum_{i\,:\,v_j \in \mathfrak{e}_i} m_{i\,j} = deg_m(v_j)\,.$$

Proof. For $j \in [\![n]\!]$, $\sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n+n_{\mathcal{A}}]\!]} a_{jj_2\dots j_{r_{\mathfrak{H}}}}$ has non-zero terms only for the corresponding hb-edges of the original hb-graph $\mathfrak{e}_i$ containing v_j. Such a hb-edge is described by $\mathfrak{e}_i = \left\{v_j^{m_{ij}}, v_{l_2}^{m_{il_2}}, \dots, v_{l_k}^{m_{il_k}}\right\}$. This means that the multiset $\{\{j_2, \dots, j_{r_{\mathfrak{H}}}\}\}$ corresponds exactly to the multiset $\left\{j^{m_{ij}-1}, l_2^{m_{il_2}}, \dots, l_k^{m_{il_k}}\right\}$.

In the straightforward approach, for each $\mathfrak{e}_i$ such that $v_j \in \mathfrak{e}_i$, there are:

$$\frac{(r_{\mathfrak{H}}-1)!}{(m_{ij}-1)!m_{il_2}!\dots m_{il_k}!m_{in+1}!}$$

possible permutations of the indices j_2 to $j_{r_{\mathfrak{H}}}$ and:

$$a_{jj_2\dots j_{r_{\mathfrak{H}}}} = \frac{m_{ij}!m_{il_2}!\dots m_{il_k}!m_{in+1}!}{(r_{\mathfrak{H}}-1)!}.$$

In the silo approach, for each $\mathfrak{e}_i$ such that $v_j \in \mathfrak{e}_i$, there are:

$$\frac{(r_{\mathfrak{H}}-1)!}{(m_{ij}-1)!m_{il_2}!\dots m_{il_k}!m_{in+\#_m\mathfrak{e}_i}!}$$

possible permutations of the indices j_2 to $j_{r_{\mathfrak{H}}}$ and:

$$a_{jj_2\dots j_{r_{\mathfrak{H}}}} = \frac{m_{ij}!m_{il_2}!\dots m_{il_k}!m_{in+\#_m\mathfrak{e}_i}!}{(r_{\mathfrak{H}}-1)!}.$$

In the layered approach, for each $\mathfrak{e}_i$ such that $v_j \in \mathfrak{e}_i$, there are:

$$\frac{(r_{\mathfrak{H}}-1)!}{(m_{ij}-1)!m_{il_2}!\dots m_{il_k}!}$$

possible permutations of the indices j_2 to $j_{r_{\mathfrak{H}}}$ which have all the same value equal to:

$$a_{jj_2\dots j_{r_{\mathfrak{H}}}} = \frac{m_{ij}!m_{il_2}!\dots m_{il_k}!}{(r_{\mathfrak{H}}-1)!}.$$

So, whatever the approach taken is, it holds:

$$\sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n]\!]} a_{jj_2\dots j_{r_{\mathfrak{H}}}} = \sum_{i\,:\,v_j \in \mathfrak{e}_i} m_{ij} = \deg_m(v_j).$$

□

3.4.2.2. Additional vertex information

The additional vertices carry information on the hb-edges of the hb-graph: the information carried depends on the approach taken.

Claim 3.9. *The layered e-adjacency tensor allows the retrieval of the distribution of the hb-edges.*

Proof. For $j \in [\![n_{\mathcal{A}}]\!]$:

$$\sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n+n_{\mathcal{A}}]\!]} a_{n+jj_2\dots j_{r_{\mathfrak{H}}}}$$

has non-zero terms only for the corresponding hb-edges of the m-uniformized hb-graph $\overline{\mathfrak{e}_i}$ containing v_j. Such a hb-edge is described by[5]:

$$\overline{\mathfrak{e}_i} = \left\{ v_k^{m_{i\,k}} : k \in [\![n+n_{\mathcal{A}}]\!] \right\}.$$

This means that the multiset:

$$\left\{ \{ j_2, \dots, j_{r_{\mathfrak{H}}} \} \right\}$$

corresponds exactly to the multiset:

$$\left\{ (n+j)^{m_{i\,n+j}-1} \right\} + \left\{ k^{m_{i\,k}} : k \in [\![n+n_{\mathcal{A}}]\!] \wedge k \neq j \right\}.$$

The number of possible permutations of elements in this multiset is:

$$\frac{(r_{\mathfrak{H}}-1)!}{(m_{i\,n+j}-1)! \prod\limits_{k \in [\![n]\!]} m_{i\,k}! \prod\limits_{\substack{k \in [\![n+1;n+n_{\mathcal{A}}]\!] \\ k \neq j}} m_{i\,k}!}$$

and the elements corresponding to one hb-edge are all equal to:

$$\frac{\prod\limits_{k \in [\![n_{\mathcal{A}}]\!]} m_{i\,k}!}{(r_{\mathfrak{H}}-1)!}.$$

Thus:

$$\sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n+n_{\mathcal{A}}]\!]} a_{n+jj_2\dots j_{r_{\mathfrak{H}}}} = \sum_{j_2,\dots,j_{r_{\mathfrak{H}}} \in [\![n]\!]} m_{i\,n+j} = \deg_m(Y_j).$$

The interpretation differs between the different approaches.

For the silo approach:

There is one added vertex in each hb-edge. The silo of hb-edges of m-cardinality m_s ($m_s \in [\![r_{\mathfrak{H}}-1]\!]$) is associated to the null vertex Y_{m_s}. The multiplicity of Y_{m_s} in each hb-edge of the silo is $r_{\mathfrak{H}} - m_s$.

Hence:

$$\frac{\deg_m(Y_j)}{r_{\mathfrak{H}} - m_s} = \left| \{ \mathfrak{e} : \#_m \mathfrak{e} = m_s \} \right|.$$

The number of hb-edges in the silo m_s is then deduced by the following formula:

$$\left| \{ \mathfrak{e} : \#_m \mathfrak{e} = m_s \} \right| = |\mathfrak{E}| - \sum_{m_s \in [\![r_{\mathfrak{H}}-1]\!]} \frac{\deg_m(Y_j)}{r_{\mathfrak{H}} - m_s}.$$

For the layered approach:

The vertex Y_j corresponds to the layer of level j added to each hb-edge with m-cardinality less or equal to j with a multiplicity of 1.

[5] With the convention, that for $j \in [\![n_A]\!]$:$v_{n+j} = y_j$

Also:

$$\deg_m(Y_j) = |\{\mathfrak{e} : \#_m\mathfrak{e} \leqslant j\}|.$$

Hence, for $j \in [\![2; r_{\mathfrak{H}} - 1]\!]$:

$$|\{\mathfrak{e} : \#_m\mathfrak{e} = j\}| = \deg_m(Y_j) - \deg_m(Y_{j-1}).$$

It yields:

$$|\{\mathfrak{e} : \#_m\mathfrak{e} = 1\}| = \deg_m(Y_1)$$

and:

$$|\{\mathfrak{e} : \#_m\mathfrak{e} = r_{\mathfrak{H}}\}| = |\mathfrak{E}| - \deg_m(Y_{r_{\mathfrak{H}}-1}).$$

For the straightforward approach:

In a hb-edge of m-cardinality $j \in [\![r_{\mathfrak{H}} - 1]\!]$, the vertex Y_1 is added with multiplicity $r_{\mathfrak{H}} - j$. The number of hb-edges with m-cardinality j can be retrieved by considering the elements of $\boldsymbol{A}_{\mathrm{str},\mathfrak{H}}$ of index $(n+1)i_1 \dots i_{r_{\mathfrak{H}}-1}$ where $1 \leqslant i_1 \leqslant \dots \leqslant i_j \leqslant n$ and $i_{j+1} = \dots = i_{r_{\mathfrak{H}}-1} = n+1$ and the elements with indices obtained by permutation.

It follows for $j \in [\![r_{\mathfrak{H}} - 1]\!]$:

$$\begin{aligned} |\{\mathfrak{e} : \#_m\mathfrak{e} = j\}| &= |\{\mathfrak{e} \colon Y_1 \in \mathfrak{e} \wedge m_{\mathfrak{e}}(Y_1) = r_{\mathfrak{H}} - j\}| \\ &= \sum_{\substack{i_1,\dots,i_{r_{\mathfrak{H}}-1} \in [\![n+1]\!] \\ |\{i_k = n+1\}| = r_{\mathfrak{H}} - j - 1}} a_{n+1 i_1 \dots i_{r_{\mathfrak{H}}-1}}. \end{aligned}$$

The terms $a_{n+1 i_1 \dots i_{r_{\mathfrak{H}}-1}}$ of this sum are non-zero only for the corresponding hb-edges $\overline{\mathfrak{e}}$ of the m-uniformized hb-graph having Y_1 with multiplicity $r_{\mathfrak{H}} - j$ in it. Such a hb-edge is described by:

$$\overline{\mathfrak{e}_i} = \left\{v_k^{m_{i\,k}} \colon 1 \leqslant k \leqslant n\right\} + \left\{Y_1^{r_{\mathfrak{H}}-j}\right\}.$$

It means that the multiset:

$$\left\{\{i_1, \dots, i_{r_{\mathfrak{H}}-1}\}\right\}$$

corresponds exactly to the multiset:

$$\left\{k^{m_{i\,k}} \colon k \in [\![n]\!]\right\} + \left\{n + 1^{r_{\mathfrak{H}}-j-1}\right\}.$$

The number of possible permutations in this multiset is:

$$\frac{(r_{\mathfrak{H}} - 1)!}{\prod\limits_{k \in [\![n]\!]} m_{i\,k}!\,(r_{\mathfrak{H}} - j - 1)!}$$

and the elements corresponding to one hb-edge are all equal to:

$$\frac{\prod\limits_{k \in [\![n]\!]} m_{i\,k}! \times (r_{\mathfrak{H}} - j)!}{(r_{\mathfrak{H}} - 1)!}.$$

Hence:

$$\frac{1}{r_{\mathfrak{H}} - j} \sum_{\substack{i_2,\dots,i_{r_{\mathfrak{H}}} \in [\![n+1]\!] \\ |\{i_k = n+1 : k \in [\![2; r_{\mathfrak{H}}]\!]\}| = r_{\mathfrak{H}} - j - 1}} a_{n+1 i_2 \dots i_{r_{\mathfrak{H}}}} = |\{\mathfrak{e} : \#_m\mathfrak{e} = j\}|.$$

The number of hb-edges of m-cardinality $r_{\mathfrak{H}}$ can be retrieved by:

$$|\{\mathfrak{e} : \#_m\mathfrak{e} = r_{\mathfrak{H}}\}| = |\mathfrak{E}| - \sum_{j \in [\![r_{\mathfrak{H}} - 1]\!]} |\{\mathfrak{e} : \#_m\mathfrak{e} = j\}|.$$

□

3.4.3. First results on hb-graph spectral analysis

Let $\mathfrak{H} = (V, \mathfrak{E})$ be a general hb-graph of e-adjacency tensor $\mathcal{A}_{\mathfrak{H}}$ of CHR $\boldsymbol{A}_{\mathfrak{H}} = \left(a_{i_1 \ldots i_{k_{\max}}}\right)$ of order $k_{\max}$ and dimension $n + n_{\mathcal{A}}$.

We write $d_{m,i} = \deg_m(v_i)$ if $i \in [\![n]\!]$ and $d_{m,n+i} = \deg_m(Y_i)$ if $i = n + j$, $j \in [\![n_{\mathcal{A}}]\!]$.

In the e-adjacency hypermatrix $\boldsymbol{A}_{\mathfrak{H}}$, the diagonal entries are no longer equal to zero. As all elements of $\boldsymbol{A}_{\mathfrak{H}}$ are non-negative real numbers, and as we have shown in the previous Section that:

$$\sum_{i_2,\ldots,i_m \in [\![n+n_{\mathcal{A}}]\!]} a_{ii_2\ldots i_{r_{\mathfrak{H}}}} = \begin{cases} d_{m,i} & \text{if } i \in [\![n]\!] \\ d_{m,n+j} & \text{if } i = n + j,\ j \in [\![n_{\mathcal{A}}]\!], \end{cases}$$

it follows:

Claim 3.10. *The e-adjacency tensor $\mathcal{A}_{\mathfrak{H}}$ of a general hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ has eigenvalues λ of its CHR $\boldsymbol{A}_{\mathfrak{H}}$ such that:*

$$|\lambda| \leqslant \max(\Delta_m, \Delta_m^\star) \tag{3.1}$$

where $\Delta_m = \max_{i \in [\![n]\!]}(d_{m,i})$ and $\Delta_m^\star = \max_{i \in [\![n_{\mathcal{A}}]\!]}(d_{m,n+i})$.

Proof. From:

$$\forall i \in [\![1, n]\!],\ \left(\mathcal{A}x^{m-1}\right)_i = \lambda x_i^{m-1}, \tag{3.2}$$

since $a_{ii_2\ldots i_m}$ are non-negative real numbers, it holds for all λ that:

$$|\lambda - a_{i\ldots i}| \leqslant \sum_{\substack{i_2,\ldots,i_m \in [\![n+n_{\mathcal{A}}]\!] \\ \delta_{ii_2\ldots i_m} = 0}} a_{ii_2\ldots i_m}. \tag{3.3}$$

Considering the triangular inequality, it follows:

$$|\lambda| \leqslant |\lambda - a_{i\ldots i}| + |a_{i\ldots i}|. \tag{3.4}$$

Combining 3.3 and 3.4 yield:

$$|\lambda| \leqslant \sum_{\substack{i_2,\ldots,i_m \in [\![n+n_{\mathcal{A}}]\!] \\ \delta_{ii_2\ldots i_m} = 0}} a_{ii_2\ldots i_m} + |a_{i\ldots i}|. \tag{3.5}$$

But, irrespective of the approach taken, if $\{i^{r_{\mathfrak{H}}}\}$ is an hb-edge of the hb-graph, then:

$$|a_{i\ldots i}| = r_{\mathfrak{H}}$$

otherwise:

$$|a_{i\ldots i}| = 0$$

and, thus, writing $\Delta_m = \max_{i \in [\![n]\!]}(\deg_m(v_i))$ and $\Delta_m^\star = \max_{i \in [\![n_{\mathcal{A}}]\!]}(\deg_m(N_i))$ and, using 3.5 and $d_{m,i} = \sum_{\substack{i_2,\ldots,i_m \in [\![n+n_{\mathcal{A}}]\!] \\ \delta_{ii_2\ldots i_m} = 0}} a_{ii_2\ldots i_m} + a_{i\ldots i}$ yield:

$$|\lambda| \leqslant \max(\Delta_m, \Delta_m^\star).$$

□

Remark 3.4.1. *In the straightforward approach:*

$$\begin{aligned}\Delta^{\star}_{m} &= \deg_m(N_1)\\ &= \sum_{j\in[\![r_{\mathfrak{H}}-1]\!]}(r_{\mathfrak{H}}-j)\,|\{\mathfrak{e}:\#_m\mathfrak{e}=j\}|\,.\end{aligned}$$

In the silo approach:

$$\begin{aligned}\Delta^{\star}_{m} &= \max_{j\in[\![r_{\mathfrak{H}}-1]\!]}(\deg_m(N_j))\\ &= \max_{j\in[\![r_{\mathfrak{H}}-1]\!]}((r_{\mathfrak{H}}-j)\,|\{\mathfrak{e}:\#_m\mathfrak{e}=j\}|)\,.\end{aligned}$$

In the layered approach:

$$\begin{aligned}\Delta^{\star}_{m} &= \max_{j\in[\![r_{\mathfrak{H}}-1]\!]}(\deg_m(N_j))\\ &= \max_{j\in[\![r_{\mathfrak{H}}-1]\!]}(|\{\mathfrak{e}:\#_m\mathfrak{e}\leqslant j\}|)\\ &= |\{\mathfrak{e}:\#_m\mathfrak{e}\leqslant r_{\mathfrak{H}}-1\}|\,.\end{aligned}$$

The values of Δ_m are independent of the approach taken.

3.4.4. Categorization of the constructed tensors

3.4.4.1. Classification of the tensors built

The hypermatrices constructed in the three approaches are symmetric and non-negative. This ensures that these hypermatrices have their spectral radius $\rho(\mathcal{A})$ which is an H^{+}-eigenvalue of $\mathcal{A}$. Hence, $\rho(\mathcal{A})$ is associated to a non-negative eigenvector—i.e. all the components of the vector are non-negative.

In [PZ14], a uniform hypergraph tensor is shown to be weakly irreducible, if and only if the hypergraph is connected. In the case of e-adjacency tensors, this result does not hold.

Nonetheless, we can claim the following results:

Claim 3.11. *Let $\mathfrak{H}=(V,\mathfrak{E})$ be a hb-graph which is not m-uniform and where $\bigcup_{\mathfrak{e}\in\mathfrak{E}}\mathfrak{e}^{\star}=V$.*

If $\mathfrak{H}$ is connected, then its straightforward e-adjacency tensor is weakly irreducible.

Proof. This proof combines arguments of [PZ14], with arguments of [QL17] on weak irreducibility of the adjacency tensor of a uniform hypergraph, and some specific arguments related to the e-adjacency tensors we use for hb-graphs.

Let $\mathfrak{H}=(V,\mathfrak{E})$ be a hb-graph which is not m-uniform and where $\bigcup_{\mathfrak{e}\in\mathfrak{E}}\mathfrak{e}^{\star}=V$.

We are going to prove the converse of this claim. Suppose that the straightforward e-adjacency tensor $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ of CHR $\boldsymbol{A}_{\mathrm{str},\mathfrak{H}}\in T_{r_{\mathfrak{H}},n+1}$ is weakly reducible, it means that the associated graph $\mathcal{G}(\mathcal{A}_{\mathrm{str},\mathfrak{H}})$ is not strongly connected, and, hence, its matrix representation $A_{\mathcal{G}(\mathcal{A}_{\mathrm{str},\mathfrak{H}})}=(\alpha_{ij})$ is reducible.

As $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ is symmetric, its associated graph $\mathcal{G}\left(\mathcal{A}_{\mathrm{str},\mathfrak{H}}\right)$ is bidirectional and $A_{\mathcal{G}\left(\mathcal{A}_{\mathrm{str},\mathfrak{H}}\right)} = (\alpha_{ij})$ is symmetric.

This means that there exists a nonempty proper subset J of $[\![n+1]\!]$ such that $\forall i \in J, \forall j \in [\![n+1]\!] \setminus J : \alpha_{ij} = 0$.

As $\mathfrak{H}$ is not m-uniform, J cannot be reduced to the singleton $\{n+1\}$ since the special vertex has to be linked to vertices of the hb-edges of m-cardinality value less than the maximal m-rank.

For symmetric reasons, J cannot be $[\![n]\!]$, otherwise, it would mean that the special vertex $n+1$ is isolated, which is not possible as the hb-graph is not m-uniform.

Thus, there exists at least one $i_1 \in J$ which represents an original vertex of $\mathfrak{H}$ such that $a_{i_1 i_2 \ldots i_{r_{\mathfrak{H}}}} = 0$ when at least one of the indices i_2, ..., $i_{r_{\mathfrak{H}}}$ is in $[\![n+1]\!] \setminus J$. In these $r_{\mathfrak{H}} - 1$ indices, at least one corresponds to an original vertex of $\mathfrak{H}$.

It indicates in this case that the group of original vertices of $\mathfrak{H}$ represented in J are disconnected from the original vertices that are in $[\![n+1]\!] \setminus J$. Hence the hb-graph is disconnected.

□

With: $\bigcup_{\mathfrak{e} \in \mathfrak{E}} \mathfrak{e}^\star = V$, we require that there is no unused vertices in the universe.

If the hb-graph is m-uniform and connected, the straightforward e-adjacency tensor $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ is weakly reducible as the additional vertex is not used and hence is isolated in the associated graph of $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$. In this case, one can use $\mathcal{A}_{\mathrm{str},\mathfrak{H}}|_{[\![n]\!]}$ the principal sub-tensor of $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ related to $[\![n]\!]$ which is weakly irreducible. As $\mathcal{A}_{\mathrm{str},\mathfrak{H}}|_{[\![n]\!]}$ is weakly irreducible, it is strictly non-negative and, the $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ is a non-trivially non-negative tensor, which means by using the theorem from [QL17] that $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ has a positive eigenvalue and hence $\rho\left(\mathcal{A}_{\mathrm{str},\mathfrak{H}}\right) > 0$.

Non-negative tensor weak irreducibility is a desirable property as it ensures that the tensor has a unique positive Perron vector (up to a multiplicative constant) associated to its spectral positive radius $\rho\left(\mathcal{A}\right)$. In this case, the algorithms to calculate the Perron vector converge.

To ensure weak irreducibility, we could transform the straightforward e-adjacency tensor in such a way that its associated graph is always strongly connected. It is sufficient to add the special vertex to each of the hb-edges: in this case, it will force the associated graph to be connected, albeit the spectral radius upper-bound to be increased to the maximum between the maximal m-degree and the number of hb-edges.

Moreover:

Claim 3.12. *The three e-adjacency tensors built for hb-graphs are non-trivially non-negative tensors when the hb-graph is connected and the union of the support of hb-edges covers the vertex set.*

Proof. We already know that for a connected hb-graph which is not m-uniform, the straightforward e-adjacency tensor is weakly irreducible, hence non-trivially non-negative.

For a m-uniform hb-graph, the principal sub-tensor of $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$ composed of its first n indices is weakly irreducible, hence strictly non-negative. The proof is similar to the one of [PZ14] and [QL17] for hypergraphs.

We have already explained in [OLGMM18b] how a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ can be decomposed in layers by considering $\mathfrak{E}_k = \{\mathfrak{e} \in \mathfrak{E}, \sharp_m \mathfrak{e} = k\}$ and the hb-graphs $\mathfrak{H}_k = (V, \mathfrak{E}_k)$ to obtain $\mathfrak{H} = \bigoplus_{k \in [\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_k$.

For the silo e-adjacency tensor, the principal sub-tensor to be considered is the one obtained using $J = [\![n]\!] \cup \{n + k : k \in [\![r_{\mathfrak{H}} - 1]\!] \wedge \mathfrak{E}_k \neq \emptyset\}$. $\mathcal{A}_{\mathrm{sil},\mathfrak{H}}|_J$ is then weakly irreducible, hence strictly non-negative. The tensor $\mathcal{A}_{\mathrm{sil},\mathfrak{H}}$ is, therefore, a non-trivially non-negative tensor.

For the layered e-adjacency tensor $\mathcal{A}_{\mathrm{lay},\mathfrak{H}}$, we consider the principal sub-tensor $\mathcal{A}_{\mathrm{lay},\mathfrak{H}}|_K$ where $K = [\![n]\!] \cup \{n + k : k \in [\![k_{\min}, r_{\mathfrak{H}} - 1]\!]\}$ with $k_{\min} = \min\{k : \mathfrak{E}_k \neq \emptyset\}$. This principal sub-tensor is weakly irreducible as the hb-graph is connected, hence strictly non-negative. Therefore, the tensor $\mathcal{A}_{\mathrm{lay},\mathfrak{H}}$ is non-trivially non-negative.

□

As a consequence the spectral radii of these tensors are positive.

3.4.4.2. Connected components and uniformisation process

The uniformisation process used to built the three tensors leads to the addition of some vertices to the hb-edges; while it does not change the number of hb-edges and the degree of the vertices that are in the original hb-graph, it has an impact on the connectivity of the uniform hb-graph compared to the original one.

Nonetheless, to address this problem, it is always possible to consider each connected component of the original hb-graph separately and build a tensor for each of these connected components. In this case, the number of connected components remains unchanged, even if we modify the internal connectivity of a connected component with the added vertices.

So we can always suppose that we address only connected hb-graphs.

3.5. Evaluation and first choice

3.5.1. Evaluation

We have put together some key features of the e-adjacency tensors proposed in this Thesis: the straightforward approach tensor $\mathcal{A}_{\mathrm{str},\mathfrak{H}}$, the silo approach tensor $\mathcal{A}_{\mathrm{sil},\mathfrak{H}}$ and $\mathcal{A}_{\mathrm{lay},\mathfrak{H}}$ for the layered approach.

The CHR of the constructed tensors have all same order $r_{\mathfrak{H}}$. $\boldsymbol{A}_{\mathrm{sil},\mathfrak{H}}$ and $\boldsymbol{A}_{\mathrm{lay},\mathfrak{H}}$ dimensions are $r_{\mathfrak{H}} - 2$ bigger than $\boldsymbol{A}_{\mathrm{str},\mathfrak{H}}$ ($n - 2$ in the worst case). $\boldsymbol{A}_{\mathrm{str},\mathfrak{H}}$ has a total number of elements $\dfrac{(n+1)^{r_{\mathfrak{H}}}}{(n + r_{\mathfrak{H}} - 1)^{r_{\mathfrak{H}}}}$ times smaller than the two other tensors.

Elements of $\boldsymbol{A}_{\text{str},\mathfrak{H}}$—respectively $\boldsymbol{A}_{\text{sil},\mathfrak{H}}$—are repeated $\dfrac{1}{n_j!}$—respectively $\dfrac{1}{n_{j\,k}!}$—times less than elements of $\boldsymbol{A}_{\text{lay},\mathfrak{H}}$. The total number of non-zero elements filled for a given hb-graph in $\boldsymbol{A}_{\text{str},\mathfrak{H}}$ and $\boldsymbol{A}_{\text{sil},\mathfrak{H}}$ are the same and is smaller than the total number of non-zero elements in $\boldsymbol{A}_{\text{lay},\mathfrak{H}}$.

Since whatever the approach taken, the tensors are symmetric, a unique value is needed to fully describe an hb-edge; moreover, this value depends only on the hb-edge composition as well as the number of elements to be filled.

All tensors are symmetric and allow the reconstruction of the hb-graph from these elements.

Node's degrees can be retrieved as it has been shown previously. Additional information on individual hb-edges is easier to retrieve with the silo and the layered approach.

3.5.2. First choice

Insofar, as the straightforward tensor is weakly irreducible for non m-uniform connected hb-graph and, as this is a sufficient desirable property to choose it, even if this choice is at the price of less practicability to retrieve information on hb-edges, we take $\mathcal{A}_{\text{str},\mathfrak{H}}$ for definition of the e-adjacency tensor of a hb-graph. The information on the shape of the hb-edges is preserved through the added special vertex, which enhances the retrieval of information on the hb-edge cardinality.

3.5.3. Hypergraphs and hb-graphs

Hypergraphs are particular case of hb-graphs and, hence, the e-adjacency tensor defined for hb-graphs can be used for hypergraphs. Since the multiplicity function for vertices of a hyperedge seen as hb-edge has its values in $\{0, 1\}$, the e-adjacency tensor elements differ only by a factorial number due to the cardinality of the hyperedge.

The definition that is retained for the e-adjacency tensor of a hypergraph is:

Definition 3.10. *The e-adjacency tensor of a hypergraph $\mathcal{H} = (V, E)$ having maximal cardinality of its hyperedges $k_{\max}$ is the tensor $\mathcal{A}_{\mathcal{H}}$ of CHR:*

$$\boldsymbol{A}_{\boldsymbol{\mathcal{H}}} \triangleq \left(a_{i_1\dots i_{r_{\mathcal{H}}}}\right)_{1\leqslant i_1,\dots,i_{r_{\mathcal{H}}}\leqslant n}$$

defined by:

$$\boldsymbol{A}_{\boldsymbol{\mathcal{H}}} \triangleq \sum_{i\in[\![p]\!]} c_{e_i}\boldsymbol{R}_{\boldsymbol{e_i}}$$

and where for $e_i = \left\{v_{j_1}, \dots, v_{j_{k_i}}\right\} \in E$, $c_{e_i} = \dfrac{k_{\max}}{k_i}$ is the dilatation coefficient and $\boldsymbol{R}_{\boldsymbol{e_i}} = \left(r_{i_1\dots i_{r_{\mathcal{H}}}}\right)$ is the associated tensor to e_i, having all non-zero elements of same value. The non-zero elements of $\boldsymbol{R}_{\boldsymbol{e_i}}$ are:

$$r_{j_1\dots j_{k_i}(n+1)^{k_{\max}-k_i}} = \frac{(k_{\max}-k_i)!}{k_{\max}!}k_i$$

and all the ones whose indices are obtained by permutation of:

$$j_1 \dots j_{k_i} \, (n+1)^{k_{\max}-k_i} \, .$$

In practice, writing $\boldsymbol{A}_{\boldsymbol{\mathcal{H}}} = \left(a_{l_1 \dots l_{k_{\max}}}\right)$, the element of $\boldsymbol{A}_{\boldsymbol{\mathcal{H}}}$ of indices $l_1, \cdots, l_{k_{\max}}$ such that:

$$\{\{l_1, \cdots, l_{r_{\mathfrak{H}}}\}\} = \left\{j_1, ..., j_{k_i}, [n+1]^{k_{\max}-k_i}\right\},$$

corresponding to a hyperedge $\mathfrak{e}_i = \left\{v_{j_1}, ..., v_{j_{k_i}}\right\}$ of the original hypergraph $\mathcal{H}$, is:

$$a_{l_1 \dots l_{r_{k_{\max}}}} = \frac{(k_{\max}-k_i)!}{(k_{\max}-1)!}.$$

Remark 3.5.1.

As in [OLGMM17a] we compare the *e*-adjacency tensors obtained by [BCM17] and by [SZB19] with the one chosen in this Thesis. The results are presented in Table 3.2.

3.6. Further comments

Extending the concept of hypergraphs to support multisets and introducing hb-graphs have allowed us to define a systematic approach for building the *e*-adjacency tensor of a hb-graph. Conversely, as hypergraphs appear as particular case of hb-graphs, the e-adjacency tensors are applicable to general hypergraphs. Hb-graphs are a good modeling framework for many real problems and have induced already some nice refinements in existing work.

The tensor constructed in [BCM17] can be seen as a transformation of the hypergraph $\mathcal{H} = (V, E)$ into a weighted hb-graph $\mathcal{H}_{\mathcal{B}} = (V, E', w_e)$: with the same vertex set but with hb-edges obtained from the hyperedges of the original hypergraph such that for a given hyperedge all the hb-edges having this hyperedge as support are considered with multiplicities of vertices such that it reaches $k_{\max}$.

We still have to analyze the behavior of our constructed e-adjacency tensor with regard to the diffusion process. The fact that information on hb-edges for hb-graphs and, therefore, for hyperedges in hypergraphs are stored in the e-adjacency tensor should provide a nice explanation of the role of the variety of hb-edge m-cardinality.

	$\mathcal{A}_{\text{str},\mathfrak{H}}$	$\mathcal{A}_{\text{sil},\mathfrak{H}}$	$\mathcal{A}_{\text{lay},\mathfrak{H}}$
Hypermatrix representation	$\boldsymbol{A}_{\text{str},\mathfrak{H}}$	$\boldsymbol{A}_{\text{sil},\mathfrak{H}}$	$\boldsymbol{A}_{\text{lay},\mathfrak{H}}$
Order	$r_{\mathfrak{H}}$	$r_{\mathfrak{H}}$	$r_{\mathfrak{H}}$
Dimension	$n+1$	$n+r_{\mathfrak{H}}-1$	$n+r_{\mathfrak{H}}-1$
Total number of elements	$(n+1)^{r_{\mathfrak{H}}}$	$(n+r_{\mathfrak{H}}-1)^{r_{\mathfrak{H}}}$	$(n+r_{\mathfrak{H}}-1)^{r_{\mathfrak{H}}}$
Total number of elements potentially used by the way the tensor is build	$(n+1)^{r_{\mathfrak{H}}}$	$(n+r_{\mathfrak{H}}-1)^{r_{\mathfrak{H}}}$	$(n+r_{\mathfrak{H}}-1)^{r_{\mathfrak{H}}}$
Number of repeated elements per hb-edge $\mathfrak{e}_j=\left\{v_{i_1}^{m_{ji_1}},\dots,v_{i_j}^{m_{ji_j}}\right\}$	$\frac{r_{\mathfrak{H}}!}{m_{ji_1}!\dots m_{ji_j}!n_j!}$ with $n_j=r_{\mathfrak{H}}-\#_m\mathfrak{e}_j$	$\frac{r_{\mathfrak{H}}!}{m_{ji_1}!\dots m_{ji_j}!n_{jk}!}$ with $n_{jk}=r_{\mathfrak{H}}-\#_m\mathfrak{e}_j$	$\frac{r_{\mathfrak{H}}!}{m_{ji_1}!\dots m_{ji_j}!}$
Number of elements to be filled per hb-edge of size s before permutation	Constant 1	Constant 1	Constant 1
Number of elements to be described to derived the tensor by permutation of indices	$\lvert\mathfrak{E}\rvert$	$\lvert\mathfrak{E}\rvert$	$\lvert\mathfrak{E}\rvert$
Value of elements corresponding to a hb-edge	Dependent of hb-edge composition $\frac{m_{ji_1}!\dots m_{ji_j}!n_j!}{(r_{\mathfrak{H}}-1)!}$	Dependent of hb-edge composition $\frac{m_{ji_1}!\dots m_{ji_j}!n_{jk}!}{(r_{\mathfrak{H}}-1)!}$	Dependent of hb-edge composition $\frac{m_{ji_1}!\dots m_{ji_j}!}{(r_{\mathfrak{H}}-1)!}$
Symmetric	Yes	Yes	Yes
Reconstructivity	Straightforward: delete special vertices	Straightforward: delete special vertices	Straightforward: delete special vertices
Nodes degree	Yes	Yes	Yes
Information on hb-edges	Yes, but not straightforward	Yes	Yes
Spectral analysis	Special vertex increases the amplitude of the bounds	Special vertices increase the amplitude of the bounds	Special vertices increase the amplitude of the bounds
Interpretability in term of hb-graph m-uniformisation	Yes	Yes	Yes

Table 3.1.: Evaluation of the hb-graph e-adjacency tensor depending on construction.

$\mathcal{A}_{\text{str},\mathfrak{H}}$ refers to the e-adjacency tensor built with the straightforward approach.
$\mathcal{A}_{\text{sil},\mathfrak{H}}$ refers to the e-adjacency tensor built with the silo approach.
$\mathcal{A}_{\text{lay},\mathfrak{H}}$ refers to the e-adjacency tensor built with the layered approach.

	$\mathcal{B}_{\mathcal{H}}$	$\mathcal{S}_{\mathcal{H}}$	$\mathcal{A}_{\mathcal{H}}$
Hypermatrix representation	$\boldsymbol{B_{\mathcal{H}}}$	$\boldsymbol{S_{\mathcal{H}}}$	$\boldsymbol{A_{\mathcal{H}}}$
Order	$k_{\max}$	$k_{\max}$	$k_{\max}$
Dimension	n	n	$n+1$
Total number of elements	$n^{k_{\max}}$	$n^{k_{\max}}$	$(n+1)^{k_{\max}}$
Total number of elements potentially used by the way the tensor is build	$n^{k_{\max}}$	$n^{k_{\max}}$	$(n+1)^{k_{\max}}$
Number of non-zero elements for a given hypergraph	$\sum_{s=1}^{k_{\max}} \alpha_s \lvert E_s \rvert$ with $\alpha_s = p_s(k_{\max}) \frac{k_{\max}!}{k_1!...k_s!}$	$\sum_{s=1}^{k_{\max}} s! \lvert E_s \rvert$	$\sum_{s=1}^{k_{\max}} \alpha_s \lvert E_s \rvert$ with $\alpha_s = \frac{k_{\max}!}{(k_{\max}-s)!}$
Number of repeated elements per hyperedge of size s	$\frac{k_{\max}!}{k_1!...k_s!}$	s	$\frac{k_{\max}!}{(k_{\max}-s)!}$
Number of elements to be filled per hyperedge of size s before permutation	Varying $p_s(k_{\max}) = \binom{k_{\max}-1}{s-1}$	Varying s if prefix is considered as non-permuting part	Constant 1
Number of elements to be described to derived the tensor by permutation of indices	$\sum_{s=1}^{k_{\max}} p_s(k_{\max}) \lvert E_s \rvert$	$\sum_{s=1}^{k_{\max}} s \lvert E_s \rvert$	$\lvert E \rvert$
Value of elements corresponding to a hyperedge	Dependent of hyperedge composition $\frac{s}{\alpha_s}$	Dependent of hyperedge composition $\frac{1}{(s-1)!}$	Dependent of hyperedge size $\frac{(k_{\max}-s)!}{(k_{\max}-1)!}$
Symmetric	Yes	No	Yes
Reconstructivity	Need computation of duplicated vertices	Need computation of duplicated vertices	Straightforward: delete special vertices
Nodes degree	Yes	Yes	Yes
Hyperedge cardinality	Not straightforward	Not straightforward	Yes
Spectral analysis	Yes	Yes	Special vertices increase the amplitude of the bounds
Interpretability in term of hypergraph / hb-graph (m-)uniformisation	No / No	No / No	No / Yes

Table 3.2.: Evaluation of the hypergraph e-adjacency tensor.

$\mathcal{B}_{\mathcal{H}}$ designates the adjacency tensor defined in [BCM17].
$\mathcal{S}_{\mathcal{H}}$ designates the adjacency tensor defined in [SZB19].
$\mathcal{A}_{\mathcal{H}}$ refers to the e-adjacency tensor as defined in this Thesis.

4. m-uniformisation processes and exchange-based diffusion

Highlights

Prerequisites: Section 1.3 and, Chapters 2 and 3.

4.1. Motivation

An expectation that has not been formulated while constructing the e-adjacency tensors, as we wanted to expand the current adjacency tensors for uniform hypergraphs, is an expectation on the diffusion process.

Expectation 4.1. *The construction of the e-adjacency tensor should have minimal impact on the matrix exchange-based diffusion process of a connected hb-graph.*

As we need a hb-graph m-uniformisation to allow the storing of the elements in a single cubical tensor, we transform the original hb-graph $\mathfrak{H}$ of incident matrix H of dimension $n \times p$ in a m-uniform hb-graph $\mathfrak{H}_{(m)}$ of incident matrix $H_{(m)}$ of dimension $(n + n_A)\, p$ where n_A is the number of special vertices added during the m-uniformisation process.

The matrix exchange-based diffusion process on a hb-graph depends on the incident matrix of the hb-graph. Considering the stationary states for vertices and hb-edges, we can measure the similarity between the two diffusions by considering the state of the vertices / hb-edges of the original hb-graph and the state of the vertices / hb-edges in the m-uniformized hb-graph. Let π_V (respectively $\pi_{V(m)}$) be the stationary state of vertices and $\pi_{\mathfrak{E}}$ (respectively $\pi_{\mathfrak{E}(m)}$) the stationary state of the hb-edges of the original hb-graph (respectively of the m-uniformized hb-graph).

We can reformulate mathematically the problem by finding, if there exists, an e-adjacency tensor $\mathcal{T}_0 \in S_{m,n} \cap N_{m,n}$ such that:

$$\mathcal{T}_0 = \underset{\mathcal{T} \in S_{m,n} \cap N_{m,n}}{\operatorname{argmin}} \left(d\left(\pi_V, \pi_{V(m)[n]}\right) + d\left(\pi_{\mathfrak{E}}, \pi_{\mathfrak{E}(m)}\right)\right)$$

where $\pi_{V(m)[n]}$ is the truncated vector $\pi_{V(m)}$ to its first n coordinates (the ones representing the states of the original vertices) and $d(\cdot,\cdot)$ is a distance, such as the Euclidean distance.

If this problem is tractable, which is in fact an open question, finding its solution is definitively NP-hard. Evaluating the impact on the diffusion at the end of the construction of the proposed tensors is then already a first step.

Research question 4.1. *What is the impact of the Hb-graph m-Uniformisation Process on the diffusion process?*

4.2. Impact of the m-uniformisation process on the exchange-based diffusion

We consider a connected unweighted hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ with no repeated hb-edges, incident matrix H and m-rank $r_{\mathfrak{H}}$. We now focus on the impact of the m-uniformisation process on the exchange-based diffusion that is involved, i.e. the formalization of the stable states for the m-uniformized hb-graphs obtained during the construction of the tensors.

In Chapter 3, we built the e-adjacency tensors by m-uniformizing the original hb-graph with the addition of special vertices to the hb-edge universe and the filling of the hb-edges with sufficient multiplicities on these vertices. We call this approach the **additional vertices filling m-uniformisation approach**.

An other approach, taken in [BCM17] and [SZB19], was to share the elements in between free places of the built tensor, i.e. elements of the tensor having repeated indices: this process can be assimilated to adding hb-edges to the corresponding hypergraph, viewed as a hb-graph. The original hb-graph is m-uniformized by splitting hb-edges that are not complete (in the sense that they have a m-cardinality strictly less than the m-rank of the original hb-graph) into multiple copies of these hb-edges and filling them by repeating the original vertices in different ways depending on the approach. We call this approach the **hb-edge splitting m-uniformisation approach**.

Both approaches have a potential impact on the exchange-based diffusion since the original structure is modified. A third approach would consist in using a combination of the two previous ones. Nonetheless, as this approach potentially combines the drawbacks, and particularly increases the number of hb-edges, we do not think it is worth considering it, since the additional vertices filling m-uniformisation approach does not require any increase of the number of vertices.

4.2.1. Additional vertices filling m-uniformisation approach

We formulate the calculus for the general case.

We write $\mathfrak{H}_{\text{proc}} \overset{\Delta}{=} (V_{\text{proc}}, \mathfrak{E}_{\text{proc}}, w_{\text{proc}})$ the hb-graph obtained by additional vertices filling m-uniformisation process of the original hb-graph and write its incident matrix $H_{\text{proc}} \in M_{(n+n_{\text{proc}}),p}$. We have: $V_{\text{proc}} = V \cup \{y_i : i \in [\![n_{\text{proc}}]\!]\}$.

The hb-edges $\mathfrak{e}_{\text{proc},j} \in \mathfrak{E}_{\text{proc}}$ of $\mathfrak{H}_{\text{proc}}$ are obtained from those of $\mathfrak{H}$ for all $j \in [\![p]\!]$ by considering the fusion: $\mathfrak{e}_{\text{proc},j} \triangleq \mathfrak{e}_j \oplus \left\{ \left\{ y_k^{m_{n+k,\,j}} : k \in [\![n_{\text{proc}}]\!] \right\} \right\}$ where the multiplicity $m_{n+k,\,j}$ depends only on the m-uniformisation process and on the m-cardinality of the original hb-edge $\mathfrak{e}_j$. The multiplicities and weights are given in Table 4.1 for the different m-uniformisation processes presented in Chapter 3.

m-uniformisation process	n_{proc}	$m_{n+k,\,j}$	$w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},j}\right)$
straightforward	1	$m_{n+1,\,j} = r_{\mathfrak{H}} - \#_m \mathfrak{e}_j$	$\dfrac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_j}$
silo	$r_{\mathfrak{H}} - 1$	$m_{n+k,\,j} = \begin{cases} r_{\mathfrak{H}} - \#_m \mathfrak{e}_j & \text{if } k = \#_m \mathfrak{e}_j \\ 0 & \text{otherwise} \end{cases}$	$\dfrac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_j}$
layered	$r_{\mathfrak{H}} - 1$	$m_{n+k,\,j} = \begin{cases} 1 & \text{if } k \in [\![\#_m \mathfrak{e}_j ; r_{\mathfrak{H}} - 1]\!] \\ 0 & \text{otherwise} \end{cases}$	$\dfrac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_j}$

Table 4.1.: Number and multiplicity of special vertices and weights introduced for the three m-uniformisation process introduced in Chapter 3, with: $k \in [\![n_{\text{proc}}]\!]$, $j \in [\![p]\!]$.

The incident matrix of $\mathfrak{H}_{\text{proc}}$ is $H_{\text{proc}} \triangleq (h_{\text{proc},\,ij})_{\substack{i \in [\![n+n_{\text{proc}}]\!] \\ j \in [\![p]\!]}}$, where:

$$\forall i \in [\![n + n_{\text{proc}}]\!], \forall j \in [\![p]\!] : h_{\text{proc},\,ij} \triangleq m_{ij}.$$

The hb-edge weights correspond to those given during the m-uniformisation process, written for all $j \in [\![p]\!]$: $w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},j}\right)$.

We deduce the stable states in the exchange-based diffusion process on the m-uniformized hb-graph $\mathfrak{H}_{\text{proc}}$.

For the stable state of vertices:

Proposition 4.1. *Writing $\pi_{V_{proc}} = (\pi_{V_{proc}i})_{i \in [\![n+n_{proc}]\!]}$ the stable state of the exchange-based diffusion on $\mathfrak{H}_{proc}$, it holds for all $i \in [\![n + n_{proc}]\!]$:*

$$\pi_{V_{proc}i} \triangleq D_{w,\mathfrak{H},proc} \sum_{j \in [\![p]\!]} \frac{w_{proc}\left(\mathfrak{e}_{proc,j}\right)}{r_{\mathfrak{H}}} m_{ij}$$

with:

$$D_{w,\mathfrak{H},proc} \triangleq r_{\mathfrak{H}} \left(\sum_{k \in [\![n+n_{proc}]\!]} d_{w,v_k} \right)^{-1}$$

that depends only on the m-uniformisation process on the original hb-graph structure.

Proof. We have:$\pi_{V_{\text{proc}}} = (\pi_{V_{\text{proc}}i})_{i \in [\![n+n_{\text{proc}}]\!]}$.

Hence, we have for all $i \in [\![n + n_{\text{proc}}]\!]$, as shown in Chapter 2:

$$\pi_{V_{\text{lay}}i} = \frac{d_{w,v_i}}{\sum\limits_{k \in [\![n+r_{\mathfrak{H}}-1]\!]} d_{w,v_k}},$$

where, for $i \in [\![n + n_{\text{proc}}]\!] : d_{w,v_i} = \sum_{j \in [\![p]\!]} m_{ij} \frac{w_{\text{proc}}(\mathfrak{e}_{\text{proc},j})}{r_{\mathfrak{H}}}$.

Writing: $D_{w,\mathfrak{H},\text{proc}} = r_{\mathfrak{H}} \left(\sum_{k \in [\![n+n_{\text{proc}}]\!]} d_{w,v_k} \right)^{-1}$ the result follows.

□

In the three m-uniformisation processes considered in Table 4.1, the weights attached to the hb-edges of the m-uniformized hb-graphs are for all $j \in [\![p]\!]$ given by:

$$w_{\text{proc}}(\mathfrak{e}_{\text{proc},j}) = \frac{r_{\mathfrak{H}}}{\#_m \mathfrak{e}_j}.$$

With this weighting, it holds: $\pi_{V_{\text{proc}}i} = D_{w,\mathfrak{H},\text{proc}} \sum_{j \in [\![p]\!]} \frac{m_{ij}}{\#_m \mathfrak{e}_j}$.

Hence, the stable state of vertices depends on the relative multiplicity compared to the m-cardinality of the hb-edges: thus, the more isolated a vertex in a hb-edge, the higher the ranking of this vertex.

Remark 4.2.1. *We introduced the technical coefficient in the Hm-UP to preserve the number of hb-edges. Keeping this coefficient at a constant value c—and in particular to 1—independently of the hb-edge—i.e. for all $j \in [\![p]\!] : w_{proc}(\mathfrak{e}_{proc,j}) = c$—, we have:*

$$\pi_{V_{proc}i} = D_{w,\mathfrak{H},proc} c \sum_{j \in [\![p]\!]} m_{ij} \propto deg_m(v_i)$$

This means that the ranking of the vertices achieved in the exchange-based diffusion is preserved in this case (at the cost of having more difficulties to retrieve the m-degree of vertices from the e-adjacency tensor).

All other weightings of the original hb-edges have a potential influence on the ranking of vertices during the diffusion process.

To compare the vertex stable state obtained with the m-uniformized hb-graph and the original hb-graph, one can compute the distance between these two stable states. It is given by:

Proposition 4.2.

$$d\left(\pi_V, (\pi_{V_{proc}i})_{i \in [\![n]\!]}\right) = \sqrt{\sum_{i \in [\![n]\!]} \left(\sum_{j \in [\![p]\!]} \left(\frac{1}{\#_m \mathfrak{H}} - \frac{D_{w,\mathfrak{H},proc} w_{proc}(\mathfrak{e}_{proc,j})}{r_{\mathfrak{H}}} \right) m_{ij} \right)^2}.$$

Proof. Immediate as: $d\left(\pi_V, (\pi_{V_{\text{proc}}i})_{i \in [\![n]\!]}\right)^2 = \sum_{i \in [\![n]\!]} (\pi_{Vi} - \pi_{V_{\text{proc}}i})^2$. □

Also, the perturbation of diffusion in the m-uniformized hb-graph is dependent on the shape of the original hb-graph and from how far away it is from m-uniformity.

Other features such as the Kendall's tau or the scaled Spearman footrule coefficients can be calculated to compare the rankings obtained from the two diffusions.

For the stable state of the hb-edges:

Proposition 4.3. *Writing* $\pi_{\mathfrak{E}_{proc}} = \left(\pi_{\mathfrak{E}_{proc}j}\right)_{j\in[\![p]\!]}$ *the stable state of hb-edges in the exchange-based diffusion on* $\mathfrak{H}_{proc}$*, we have for all* $j \in [\![p]\!]$:

$$\pi_{\mathfrak{E}_{proc}j} = D_{w,\mathfrak{H},proc} w_{proc}\left(\mathfrak{e}_{proc,j}\right).$$

Proof. $\pi_{\mathfrak{E}_{\mathrm{proc}}j} = \dfrac{w_{\mathrm{proc}}\left(\mathfrak{e}_{\mathrm{proc},j}\right) \times \#_m\left(\mathfrak{e}_{\mathrm{proc},j}\right)}{\sum\limits_{k\in[\![n+n_{\mathrm{proc}}]\!]} d_{w,v_k}} = \dfrac{w_{\mathrm{proc}}\left(\mathfrak{e}_{\mathrm{proc},j}\right) \times r_{\mathfrak{H}}}{\sum\limits_{k\in[\![n+n_{\mathrm{proc}}]\!]} d_{w,v_k}} = D_{w,\mathfrak{H},\mathrm{proc}} w_{\mathrm{proc}}\left(\mathfrak{e}_{\mathrm{proc},j}\right).$

□

The stable state no longer reflects the m-cardinality of the hb-edge.

Remark 4.2.2. *Again the stable state of the diffusion for the hb-edges is highly disrupted by the weights of the Hm-UP process. If these weights are put equal to 1, it holds for all* $j \in [\![p]\!]$: $\pi_{\mathfrak{E}_{proc}j} = \dfrac{1}{p}$ *which implies that the diffusion is non discriminative on the hb-edges.*

4.2.2. Hb-edge splitting m-uniformisation approach

We do not formulate systematically the approach but, instead, give the outlines and focus on the m-uniformisation process induced by the tensor proposed in [BCM17] for a general hypergraph $\mathcal{H} = (V, E)$ of rank $r_{\mathcal{H}}$.

In Appendix Section E.1, for the tensor proposed in [BCM17], we have noticed that hyperedges of lower cardinality than the rank $r_{\mathcal{H}}$ are stored in elements of repeated indices of the e-adjacency tensor with rank $r_{\mathcal{H}}$ and dimension $|V|$. To keep the tensor symmetric, all the s-compositions of $r_{\mathcal{H}}$ in $s = \#e_j$ elements have to be considered and from there all the combinations of indices reflecting the s-composition. Hence, the process cannot be interpreted in term of hypergraph uniformisation since the vertices must be repeated and it requires a hb-graph m-uniformisation process: we propose this interpretation in this Section.

We have also mentioned that in the tensor proposed in [BCM17] a given hyperedge $e_j = \{v_{l_1}, ..., v_{l_s}\}$ is stored through a set of elements of the e-adjacency tensor of rank $r_{\mathcal{H}}$ and dimension $n = |V|$ whose indices corresponds to permutations of the different multisets $\left\{\left\{l_i^{k_i}, i \in [\![s]\!]\right\}\right\}$ such that: $\forall i, k_i \geqslant 1$ and $\sum\limits_{i\in[\![s]\!]} k_i = r_{\mathcal{H}}$.

Hence, taking a hb-graph m-uniformisation interpretation means that the hyperedge e_j considered as a hb-edge is transformed into a collection of hb-edges of universe V. To refine this interpretation, we consider $\mathcal{P}_{\#e_j}\left(r_{\mathcal{H}}\right)$ the set of compositions of $r_{\mathcal{H}}$ in $\#e_j$ elements defined by:

$$\mathcal{P}_{\#e_j}\left(r_{\mathcal{H}}\right) \triangleq \left\{ \left(k_i\right)_{i\in[\![\#e_j]\!]} : \forall i \in [\![\#e_j]\!], k_i \geqslant 1 \wedge \sum_{i\in[\![\#e_j]\!]} k_i = r_{\mathcal{H}} \right\}.$$

Indexing the elements of $\mathcal{P}_{\#e_j}(r_{\mathcal{H}})$ with a bijective mapping:

$$\psi : [\![\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})]\!] \to \mathcal{P}_{\#e_j}(r_{\mathcal{H}}),$$

we associate a family of hb-edges $\left(\mathfrak{e}_{\text{proc},(j,p_j)}\right)_{p_j \in [\![\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})]\!]}$ to e_j such that for all $p_j \in [\![\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})]\!]$:

$$\mathfrak{e}_{\text{proc},(j,p_j)} \triangleq \left\{ \left\{ v_{l_j}^{k_{i_j}} : \psi(p_j) = \left(k_{l_j}\right)_{l_j \in [\![\#e_j]\!]} \right\} \right\},$$

considering that $e_j = \left\{v_{l_1}, \cdots, v_{l_{\#e_j}}\right\}$.

We write:

$$m_{i(j,p_j)} \triangleq \begin{cases} k_{l_j} & \text{if } \exists l_j : v_{l_j} \in e_j \wedge v_i = v_{l_j}; \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the initial hypergraph $\mathcal{H}$ is transformed into a hb-graph

$$\mathfrak{H}_{\text{proc}} = (V_{\text{proc}}, \mathfrak{E}_{\text{proc}}, w_{\text{proc}})$$

obtained by a hb-edge splitting m-uniformisation process, of incident matrix

$$H_{\text{proc}} \in M_{n,p_{\text{proc}}}.$$

We have $V_{\text{proc}} = V$. An original hb-edge $\mathfrak{e}_j$ is split into $\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})$ hb-edges $\mathfrak{e}_{\text{proc},j,k}$. Thus, the hb-edge family of the hb-edge splitting hb-graph is:

$$\mathfrak{E}_{\text{proc}} \triangleq \left(\left(\mathfrak{e}_{\text{proc},(j,p_j)}\right)_{p_j \in [\![\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})]\!]} \right)_{j \in [\![p]\!]}.$$

For $j \in [\![p]\!] \wedge p_j \in [\![\#\mathcal{P}_{\#e_j}(r_{\mathcal{H}})]\!]$, $w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},(j,k_j)}\right) = \#e_j$.

In order to build the incident matrix of $\mathfrak{H}_{\text{proc}}$, we re-index $\mathfrak{E}_{\text{proc}}$ by considering the bijective mapping:

$$\phi : \begin{cases} [\![p]\!] \times \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right] & \to \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right] \\ (j, p_j) & \mapsto \sum_{k \in [\![j-1]\!]} \#\mathcal{P}_{\#e_j}(r_{\mathcal{H}}) + p_j \end{cases}.$$

$H_{\text{proc}} = (h_{il})$ is built using the multiplicities of the different vertices for $i \in [\![n]\!]$, $l \in \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right]$:

$$h_{il} = m_{i\phi^{-1}(l)}.$$

For the stable state of vertices:

Proposition 4.4. *Writing $\pi_{V_{proc}} \triangleq (\pi_{V_{proc}i})_{i \in [\![n]\!]}$ the stable state of the exchange-based diffusion on $\mathfrak{H}_{proc}$, it holds for all $i \in [\![n]\!]$:*

$$\pi_{V_{proc}i} \triangleq D_{w,\mathfrak{H},proc} \sum_{l \in \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right]} \frac{w_{proc}\left(\mathfrak{e}_{proc,\phi^{-1}(l)}\right)}{r_{\mathfrak{H}}} m_{i\phi^{-1}(l)}$$

with $D_{w,\mathfrak{H},proc} \triangleq r_{\mathfrak{H}} \left(\sum_{k \in [\![n]\!]} d_{w,v_k} \right)^{-1}$ *that depends only on the m-uniformisation process on the original hb-graph structure.*

Proof. We have:

$$\pi_{V_{\text{proc}}} = \left(\pi_{V_{\text{proc}}i}\right)_{i \in [\![n]\!]}.$$

Hence, we have for all $i \in [\![n]\!]$, as shown in Chapter 2:

$$\pi_{V_{\text{proc}}i} = \frac{d_{w,v_i}}{\sum_{k \in [\![n]\!]} d_{w,v_k}},$$

where, for $i \in [\![n]\!]$:

$$d_{w,v_i} = \sum_{l \in \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right]} \frac{w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right)}{r_{\mathfrak{H}}} m_{i\phi^{-1}(l)}.$$

Writing: $D_{w,\mathfrak{H},\text{proc}} = r_{\mathfrak{H}} \left(\sum_{k \in [\![n]\!]} d_{w,v_k} \right)^{-1}$ the result follows.

□

We can remark that now the stable state of the diffusion on the m-uniformized hb-graph is dependent on the way the hyperedges are split and filled. The relationship with the initial diffusion is no more straightforward, and the m-uniformisation process disturbs the diffusion process in a way that seems difficult to be interpreted.

For the stable state of the hb-edges:

Proposition 4.5. *Writing* $\pi_{\mathfrak{E}_{proc}} = \left(\pi_{\mathfrak{E}_{proc}l}\right)_{l \in \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right]}$ *the stable state of hb-edges in the exchange-based diffusion on* $\mathfrak{H}_{proc}$*, we have for all* $l \in \left[\!\left[\sum_{k \in [\![p]\!]} \#\mathcal{P}_{\#e_k}(r_{\mathcal{H}}) \right]\!\right]$:

$$\pi_{\mathfrak{E}_{proc}l} = D_{w,\mathfrak{H},proc} w_{proc}\left(\mathfrak{e}_{proc,\phi^{-1}(l)}\right).$$

Proof.

$$\begin{aligned}
\pi_{\mathfrak{E}_{\text{proc}}l} &= \frac{w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right) \times \#_m\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right)}{\sum_{k \in [\![n]\!]} d_{w,v_k}} \\
&= \frac{w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right) \times r_{\mathfrak{H}}}{\sum_{k \in [\![n]\!]} d_{w,v_k}} \\
&= D_{w,\mathfrak{H},\text{proc}} w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right).
\end{aligned}$$

□

Writing $\phi^{-1}(l) = (j, k_j)$ where $j \in [\![p]\!]$, we have: $w_{\text{proc}}\left(\mathfrak{e}_{\text{proc},\phi^{-1}(l)}\right) = \#e_j$. Hence, the stable state of the hb-edges reflects the m-cardinality of the original hb-graph, i.e. the cardinality of the hyperedges of the original hypergraph.

4.3. e-adjacency tensors require compromises

We start by giving some lemma:

Lemma 4.1. *A splitting hb-edge m-uniformisation process with no additional vertices preserves only the $e^\star$-adjacency of the original non m-uniform hb-graph.*

Proof. As the hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ is non m-uniform, there exists a hb-edge $\mathfrak{e} \in \mathfrak{E}$ such that $\#_m\mathfrak{e} < r_{\mathfrak{H}}$. As there are no additional vertices added during the m-uniformisation, the hb-edge is filled only by additional vertices of its support. The multiplicity of the corresponding vertices is therefore changed compared to the original multiplicity, thus the e-adjacency is not preserved and only the $e^\star$-adjacency is preserved.

□

Lemma 4.2. *Preserving globally e-adjacency in a m-uniformisation process of a non m-uniform hb-graph requires at least one additional vertex.*

Proof. Either we extend the number of hb-edges by a splitting process, and, as seen in the previous lemma, additional vertices are required, in order to keep the multiplicity of existing vertices and to allow the global preservation of e-adjacency. Or we add one or more special vertices to fill the hb-edges of m-cardinality $s < r_{\mathfrak{H}}$ up to a m-cardinality of $r_{\mathfrak{H}}$, while preserving the multiplicity of the vertices of the original hb-graph and thus the e-adjacency.

□

The presence of the additional vertices avoids to disturb the global e-adjacency, even if the strict e-adjacency is not preserved.

Remark 4.3.1. *The preservation of $e^\star$-adjacency in a m-uniformisation process is not sufficient to preserve the stable state of vertices in the exchange-based diffusion of a connected hb-graph.*

Proof. If it was the case, it would mean that both a hb-graph and its support hypergraph, that have same $e^\star$-adjacency would lead to the same exchange-based diffusion process, which is not always the case, even for a connected m-uniform hb-graph having a uniform hypergraph as support, as the former is taking into account the multiplicities and the latter not.

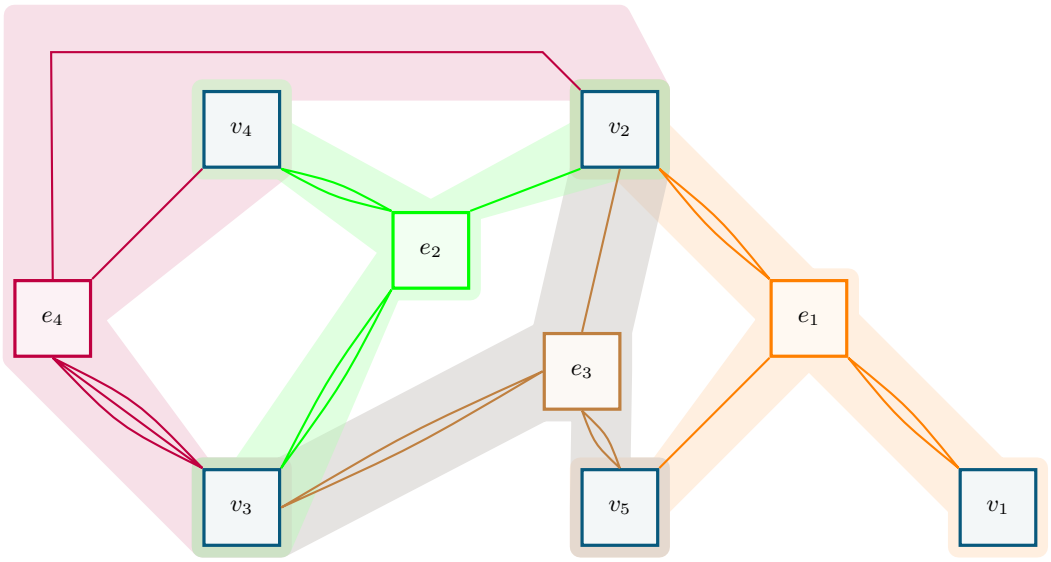


Figure 4.1.: Representation of the 5-m-uniform hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ with $V = \{v_i : i \in [\![5]\!]\}$ and $\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![4]\!]}$, where: $\mathfrak{e}_1 = \{\{v_1^2, v_2^2, v_5^1\}\}$, $\mathfrak{e}_2 = \{\{v_2^1, v_3^2, v_4^2\}\}$, $\mathfrak{e}_3 = \{\{v_2^1, v_3^2, v_5^2\}\}$ and $\mathfrak{e}_4 = \{\{v_2^1, v_3^3, v_4^1\}\}$.

For instance, we consider the 5-m-uniform hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ as represented in Figure 4.1 , with $V = \{v_i : i \in [\![5]\!]\}$ and $\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![4]\!]}$, where: $\mathfrak{e}_1 = \{\{v_1^2, v_2^2, v_5^1\}\}$, $\mathfrak{e}_2 = \{\{v_2^1, v_3^2, v_4^2\}\}$, $\mathfrak{e}_3 = \{\{v_2^1, v_3^2, v_5^2\}\}$ and $\mathfrak{e}_4 = \{\{v_2^1, v_3^3, v_4^1\}\}$ of incident matrix:

$$H = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 \\ 0 & 2 & 2 & 3 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \end{bmatrix}$$

brings to the stable state of the vertices: $\pi_V = \frac{1}{20}\begin{pmatrix} 2 & 5 & 7 & 3 & 3 \end{pmatrix}^\intercal$ while the support hypergraph $\underline{\mathfrak{H}} = (V, \underline{\mathfrak{E}})$ of $\mathfrak{H}$ of incident matrix:

$$\underline{H} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

leads to the stable state for the vertices: $\underline{\pi_V} = \frac{1}{13}\begin{pmatrix} 2 & 4 & 3 & 2 & 2 \end{pmatrix}^\intercal$ which leads to an inversion of ranking between the vertices v_2 and v_3, and to tie v_1, v_4 and v_5.

□

Theorem 4.1. *There is no Hm-UP process that preserves both e-adjacency of the original hb-graph and preserves the exchange-based diffusion simultaneously on vertices and hb-edges.*

Proof. Suppose that such a process exists, then it transforms the original hb-graph of incident matrix H into a m-uniformized hb-graph of incident matrix H_{proc} having similar stable states of exchange-based diffusion i.e. stable states that rank vertices and hb-edges similarly to the diffusion on the original hb-graph while preserving e-adjacency in the m-uniformized hb-graph.

The strict e-adjacency preservation excludes the hb-edge splitting m-uniformisation approach without additional vertex where initial hb-edges are split and filled with different vertices of the original universe, thus modifying the e-adjacency of the vertices, preserving only the $e^\star$-adjacency (see lemma).

In the other m-uniformisation approach, the hb-edges are filled with additional vertices in multiplicity sufficient such that each hb-edge is filled to have a m-cardinality equal to the m-rank. This approach preserves the e-adjacency at least globally for the hb-edges that have a m-cardinality lower than the m-rank and strictly for the ones that are already at the upper value of m-cardinality.

In this last approach, the added vertices, called special vertices, do not modify the m-degree of the existing vertices but only the m-cardinality of the hb-edges of the original hb-graph. Two cases can occur: the first case consists in achieving the m-uniformisation without weighting the hb-edges, and, hence, the stable state in the m-uniformized hb-graph differs for the original vertices only with a multiplicative constant (as we have shown in the Section 4.2.1). The stable state of the hb-edges differs from the original hb-graph due to the m-uniformisation. The second case consists in weighting the hb-edges; in this case if the weighting is hb-edge dependent the stable state of the vertices can also be modified in the m-uniformized hb-graph compared to the original hb-graph. For the hb-edges, the same effect occurs than in the unweighted case. Hence the result.

□

Nonetheless, in the unweighted version of the additional vertices filling m-uniformisation approach, the stable state of hb-edges can be adjusted to use only the stable state vertices of the original hb-graph. In this case, the original state can be retrieved—at the cost of having difficulties to retrieve the degree of vertices in the constructed tensor.

4.4. Further comments

We have shown in this Chapter that whatever the m-uniformisation approach taken, it has an impact on the exchange-based diffusion either at the vertex level or (non exclusive) at the hb-edge level. The additional vertices filling m-uniformisation approach can bring to the same ranking than the original hb-graph at the cost of having more troubles to handle the m-degree of the vertices. If we keep the proposed tensors, then we have to keep in mind that the diffusion process has to be informed of this deformed structure.

5. Diffusion in hb-graphs (tensor approach)

Highlights

Prerequisites: Section 1.3 and, Chapters 2, 3 and 4.

5.1. Motivation

So far, we have used the incident matrix to enhance diffusion on hb-graphs. Building an e-adjacency tensor for general hb-graphs and analyzing the impact of the m-uniformisation has led us to the conclusion that the m-uniformisation process impacts the diffusion process. Hence, the necessity to have either an informed diffusion or to have a diffusion that operates at the individual levels of the layered decomposition of an hb-graph.

In this Chapter, we want to address the following research question:

Research question 5.1. *How to enable a tensor-based diffusion in general hb-graphs?*

We start by giving in Section 5.2 the related work and mathematical background on diffusion over networks, seen as graphs or hypergraphs. We then propose in Section 5.3 different lines of thoughts to tackle the problem of diffusion in general hb-graphs.

5.2. Related work

Diffusion kernels for graphs have been proposed in [KL02] linking them to exponential kernels and stochastic processes on graphs, as well as lazy random walks. In [BN03], the authors study the Laplace Beltrami operator on differentiable functions on a manifold and show the tight link with the heat flow. In [LL05], diffusion kernels are introduced that exploit the geometric structure of statistical models, particularly the multinomial kernel which allows to learn from discrete data; it is applied to some text corpus.

5.2.1. Operators

Definitions and results of this section are based on [Kat13], which is a reference book on operators. Originally, operators refer to transformations that manipulate functions and give back functions. Some people are restricting to transformations on a same space.

An **operator** $O : E \to F$ is a mapping between two topological vector spaces E and F with scalars in a field $\mathbb{K}$.

An operator $O : E \to F$ is **linear** if:

$$\forall (\lambda, \mu) \in \mathbb{K}^2, \forall (f_1, f_2) \in E^2,\ O(\lambda f_1 + \mu f_2) = \lambda O(f_1) + \mu O(f_2).$$

In practice, the operator can be defined on a subspace D of E called **definition domain** of the operator.

The **rank** of an operator $O : E \to F$ is the dimension of $O(E)$.

The **kernel** of O is the pre-image of 0_F.

In finite dimension, it holds $\text{rank}(O) + \dim(\text{Ker}(O)) = \dim(E)$. In finite dimension, a linear operator is uniquely represented by a matrix.

The linear sum of two operators is an operator.

5.2.2. Diffusion operators

The results of this section are based on the attached reference[1].

A **diffusion operator** L on $\mathbb{R}^n$ is defined as a differential operator that can be written:

$$L = \sum_{i,j \in [\![n]\!]} \sigma_{ij}(x) \frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i \in [\![n]\!]} b_i(x) \frac{\partial}{\partial x_i}$$

where b_i and σ_{ij} are continuous functions on $\mathbb{R}^n$ and such that for all $x \in \mathbb{R}^n$: $(\sigma_{ij}(x))_{i,j \in [\![n]\!]}$ is a symmetric and non-negative matrix.

The **Laplace operator** $\Delta = \sum_{i \in [\![n]\!]} \frac{\partial^2}{\partial x_i^2}$ on $\mathbb{R}^n$ is a diffusion operator.

The author shows that diffusion operators satisfy a maximum principle:

Proposition 5.1. ***Maximum principle for diffusion operators***

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth function (such that $f \in \mathcal{C}^\infty(\mathbb{R}^n)$) that reaches a local minimum at x. If L is a diffusion operator, then $Lf(x) \geqslant 0$.

Reciprocally, combined with linearity, the **maximum principle** characterizes a diffusion operator.

Theorem 5.1. *Let $L : \mathcal{C}^\infty(\mathbb{R}^n) \to \mathcal{C}^0(\mathbb{R}^n)$ be an operator such that L is linear and such that if $f \in \mathcal{C}^\infty(\mathbb{R}^n)$ has a local minimum at x, $Lf(x) \geqslant 0$.*

Then L is a diffusion operator.

[1] https://fabricebaudoin.wordpress.com/2013/06/21/lecture-1-diffusion-operators/

This theorem is a particular case of the Courrège's theorem given in [Cou65] that gives the necessary and sufficient condition for a linear operator to satisfy the positive maximum principle; an operator $A : \mathcal{C}_c^\infty(\mathbb{R}^n) \mapsto \mathcal{C}(\mathbb{R}^n)$ is said to satisfy the positive maximum principle, if for any function f of the space of smooth and compact supported functions $\mathcal{C}_c^\infty(\mathbb{R}^n)$ that has a global maximum and, that is non-negative in x, then $Af(x)$ is non-negative.

5.2.3. Diffusion processes in graphs

In [FW93], the authors study different classical stochastic processes leading to diffusion processes on graphs, by studying diffusion in narrow tubes: each time they give the associated second-order differential operator associated to the diffusion process.

In [Mer95], a full survey on graph Laplacian is realized. For small vibrations, the oscillations of a point on the membrane are approximately vertical, and depends only on the place on the drum and the time, i.e. $z = f(x, y, t)$ with $f \in \mathcal{C}^2(\mathbb{R}^2)$ and the equation of the movement is given by the wave equation $\dfrac{\partial^2 f}{\partial t^2} = c^2 \Delta f$, where Δ is the Laplacian operator on $\mathbb{R}^2$, as it has been defined in the previous section.

If $\dfrac{\partial^2 f}{\partial t^2}$ is supposed to be oppositely proportional to the vertical move, i.e. $\dfrac{\partial^2 f}{\partial t^2} = -kf$, we have $\Delta f = -\dfrac{k}{c^2} f$.

Forgetting time, and considering a square grid of unit size h around (x, y), i.e. taking finite differences, we have:

$$f(x+h, y) = f(x, y) + h\frac{\partial f}{\partial x}(x, y) + h^2 \frac{\partial^2 f}{\partial x^2}(x, y) + o\left(h^2\right). \tag{5.1}$$

We can also write:

$$f(x-h, y) = f(x, y) - h\frac{\partial f}{\partial x}(x, y) + h^2 \frac{\partial^2 f}{\partial x^2}(x, y) + o\left(h^2\right). \tag{5.2}$$

By summing of (5.1) with (5.2) and rearranging, it holds that:

$$\frac{\partial^2 f}{\partial x^2}(x, y) \approx \frac{1}{h^2}\left(f(x+h, y) + f(x-h, y) - 2f(x, y)\right). \tag{5.3}$$

Similarly:

$$\frac{\partial^2 f}{\partial y^2}(x, y) \approx \frac{1}{h^2}\left(f(x, y+h) + f(x, y-h) - 2f(x, y)\right). \tag{5.4}$$

Globally, gathering (5.3) with (5.4), we have:

$$\Delta f \approx \frac{1}{h^2}\left(f(x+h, y) + f(x-h, y) + f(x, y+h) + f(x, y-h) - 4f(x, y)\right).$$

It means that the Laplacian, close to (x, y) is the sum of the difference of values in between the neighbor values and the point itself.

Coming back to the grid and taking $h = 1$, it holds that the Laplacian of the 5 points on the grid is:

$$L = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

that corresponds to the matrix view of the operator Δ.

Generalizing the concept for a graph, it becomes natural to define the **Laplacian of a graph** as $L \triangleq D - A$ with $D = \text{diag}\left((d_i)_{i\in[\![n]\!]}\right)$ with d_i the degree of the vertex v_i and $A = (a_{ij})_{i,j\in[\![n]\!]}$ the adjacency matrix of the graph.

Coming back to the oscillating membrane, it holds:

$$4f(x,y) - f(x+h,y) - f(x-h,y) - f(x,y+h) - f(x,y-h) = \frac{kh^2}{c^2} f(x,y)$$

Writing $\mu = \dfrac{kh^2}{c^2}$, μ appears to be an eigenvalue of Δ, and solving $\Delta f = \mu f$ is equivalent to finding the eigenpairs of the Laplacian operator. Hence, the problem reduces to finding eigenpairs of L on the grid of finite elements.

A lot of work has been done on the eigenvalues of the Laplacian and connecting these eigenvalues to different features of the graph. The interested reader can refer to [Mer95].

5.2.4. Diffusion processes in hypergraphs

There are two kinds of approaches to define a Laplacian for hypergraph: the matrix and the tensor approach. The matrix approach relies more or less on the adjacency matrix of a hypergraph, which is a pairwise view of the hypergraph and focuses on extending the properties obtained for the graph Laplacian to the hypergraph Laplacian. In the random walks, the transition matrix are expressed using the incident matrix decorated by additional matrices depending on the way the hypergraph is viewed as a graph.

5.2.4.1. Matrix Laplacians

Mainly three different approaches exist to define matrix Laplacians. The first approach consists in using the adjacency matrix of the hypergraph. The second approach consists in using the incident matrix. The third one relies on random walks achieved on the hypergraph to approximate the Laplacian. Whatever the method taken, as matrices are used, there is an underlying pairwise approach, either directly when considering the adjacency matrix, or indirectly with the incident matrix or with random walks. We have gathered in Table 5.1 the different methods.

In [Chu93], a Laplacian based on homological considerations is introduced: the Laplacian of a k-uniform hypergraph is defined as the matrix $L \triangleq D - A + \rho\left(K + (k-1)I\right)$ where $\rho = \dfrac{d}{n}$ with d corresponding to the average degree, n being the number of vertices of the hypergraph, D the diagonal matrix constituted of the degree of the vertices, A

		Adjacency matrix	Incident matrix
Hypergraph as a graph		[ZSC99]	[Rod03]
Homological approach		[Chu93]	
Random walk based	Simple		[ZHS07][CR19]
	Higher-order		[LP11]

Table 5.1.: Ways of defining matrix Laplacian of a hypergraph.

the adjacency matrix with coefficients corresponding to the binary presence of two given vertices in one hyperedge, K the matrix associated to the complete graph and I the identity matrix. This Laplacian is viewed as a self-adjoint operator.

An other matrix approach was introduced in [ZSC99] where the hypergraph is viewed in its 2-section. The Laplacian of the hypergraph is built from the adjacency matrix of the 2-section of the hypergraph, considering its adjacency matrix $A = (a_{ij})$ where a_{ij} corresponds to the weight of the edge between vertex v_i and v_j having the vertices v_i and v_j in common. The **Laplacian degree** is defined as $\delta_l(v_i) \triangleq \sum_{j \in [\![n]\!]} a_{ij}$ and the Laplacian is defined as $L \triangleq D - A$, where $D \triangleq \operatorname{diag}\left((\delta_l(v_i))_{i \in [\![n]\!]}\right)$.

In [Rod03], the authors take a similar approach, several bounds are then given on different hypergraph features. This approach is more geometric in the sense that the hypergraph is seen as a multipartite graph or a weighted graph.

In [ZHS07], the approach is pairwise based using a Laplacian matrix defined from the degree normalised adjacency matrix. This approach is deeply based on the transition matrix of a random walk defined on the hypergraph: being on a vertex, a random incident hyperedge is chosen proportionally to the hyperedge weights and then a vertex is chosen uniformly at random. Defining the matrix $\Theta \triangleq D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$ and the Laplacian matrix as $\Delta \triangleq I - \Theta$ enhance a spectral partition of the hypergraph.

In [LP11], the authors start by classifying the two types of approaches taken by previous authors to define a Laplacian for hypergraphs. Either the previous authors have a geometric / homological point of view by considering the self-joint operator on the function of the vertices, like it has been done for graphs in [Chu93] or they consider a symmetrization of the transition matrix of the random walk performed on a graph based structure. They introduce high-order random walks and define from them generalized Laplacian matrices on hypergraphs.

In [CR19], the authors propose a hypergraph Laplacian corresponding to a symmetrization of the random probability transition similar to the modified random walk we proposed in Chapter 2. The hypergraph has hyperedge-based weighting of the vertices, i.e. is a hb-graph. Writing P the probability transition matrix and π the stationary state, they define a Laplacian as $L \triangleq \pi - \dfrac{\pi P + P^T \pi}{2}$ using a weighted directed graph that is a directed 2-section of the hb-graph adding double edges between any vertices and loops on every vertices. Nonetheless, as it is a matrix representation of a 2-section, it remains an approximation of an ideal non-linear Laplacian.

5.2.4.2. Tensor Laplacians

In [LQY13], the authors introduce a Laplacian tensor for m-uniform hypergraph by considering for a hypergraph $\mathcal{H} = (V, E)$ a polynomial $f_{\mathcal{H}}(\boldsymbol{x}) \triangleq \sum_{i \in [\![n]\!]} \sum_{j \in [\![n]\!]} \delta_{ij} (x_i - x_j)^m$ where

$$\delta_{ij} \triangleq \begin{cases} 1 & \text{if } i < j \wedge \exists e_l \in E : v_i \in e_l \wedge v_j \in e_l \\ 0 & \text{otherwise.} \end{cases}$$

The authors define the **Laplacian tensor** $\mathcal{L}$ as $\mathcal{L}\boldsymbol{x}^m \triangleq f_{\mathcal{H}}(\boldsymbol{x})$ and the associated characteristic tensor of the hypergraph as $\mathcal{C} = -\mathcal{L}$. They then focus on m-uniform hypergraphs with m even: the maximum Z-eigenvalue of $\mathcal{C}$ is shown to be 0 and they show that the corresponding eigenspace has for dimension the number of connected components in the hypergraph. They also use the second maximal Z-eigenvalue to give a bound on the bi-partition width of a hypergraph—the bi-partition width is the minimal number of hyperedges that has to be cut such that the vertex set is cut in two equal or one-unity difference parts.

In [Qi14], the **Laplacian tensor of a uniform hypergraph** is simply defined as: $\mathcal{L} \triangleq \mathcal{D} - \mathcal{A}$ and the signless Laplacian as $\mathcal{L} \triangleq \mathcal{D} + \mathcal{A}$ where $\mathcal{D}$ is the diagonal tensor composed of the degrees of the vertices. H-, H^+- and H^{++}-eigenvalues of these tensors are then studied. An interesting definition is the one for a hyperedge $e_j = \{v_{i_1}, ..., v_{i_k}\}$ of a k-uniform hypergraph of a k-th order n-dimensional symmetric tensor $\mathcal{L}(e_j)$ such that for $x \in \mathbb{C}^n$:

$$\mathcal{L}(e_j) x^k = \sum_{l \in [\![k]\!]} x_{i_l}^k - k x_{i_1} ... x_{i_k}.$$

In this case, we have: $\mathcal{L}x^k = \sum_{e_j \in E} \mathcal{L}(e_j) x^k$.

This link between Laplacian tensor and multivariate polynomial is used in [DWL19] for studying the Laplacian of general hypergraphs. As defined similarly to the e-adjacency tensor of [BCM17], this polynomial is of degree $r_{\mathcal{H}}$.

In [HQ15], the normalised signed Laplacian of a k-uniform hypergraph is defined as $\mathcal{L} \triangleq \mathcal{I} - \mathcal{A}$ where $\mathcal{A}$ is the eigenvalue normalized $\overline{k}$-adjacency tensor defined in Section E.1.3.2. The remaining work focuses on studying the H- and H^+-eigenvalues of this normalised Laplacian.

In [BCM17], two signed Laplacians are defined for general hypergraphs, one unnormalized and one normalized: they are direct generalizations to general hypergraphs of the unnormalized version of [Qi14] and the normalized version of [HQ15] for uniform hypergraph.

We can note that none of these works are focusing on the diffusion process that is behind the definition of the Laplacian.

5.2.4.3. Hypergraph Laplacian and diffusion process

In two successive papers, [CLTZ18] and [CTWZ19], the authors study a diffusion process on hypergraphs. In [CLTZ18], the diffusion process sends the flow of information from

vertices with maximal density in the hyperedge to the one with minimal density. In [CTWZ19], this diffusion process is refined: the flow of information goes from vertices with higher density to vertices of the hyperedge with lower density using a mediator that relays this information to the lower density vertices. In both papers, they consider the diffusion operator linked to the diffusion process and deduce some spectral properties. A vertex acts as mediator depending on a parameter that is hyperedge dependent and transmits only a difference of flow depending on the minimum and maximal values of the density of the other vertices in the hyperedge.

The Laplacian associated to this diffusion is defined as the operator L_w on the density $f \in \mathbb{R}^n$, where n is the number of vertices such that $L_w f = -\dfrac{df}{dt}$. Spectral properties of this Laplacian are then studied. This approach keeps a kind of pairwise relationship, diffusing the information only between higher density vertices and the lower ones.

5.3. A diffusion operator for general hb-graphs

As we have seen in Chapter 2, whatever the approach taken, the diffusion is disturbed by the m-uniformisation process involved to store the information into a symmetrical cubic tensor. In this Section, we propose different hints to handle this problem.

5.3.1. Global exchange-based diffusion

We start by giving here an extension of the exchange-based diffusion proposed in Chapter 2 using the e-adjacency tensor built in Chapter 3.

We consider for the moment a connected non-uniform unweighted hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ with $|V| = n$ and $|\mathfrak{E}| = p$; we write $\boldsymbol{A}$ its e-adjacency tensor defined in Chapter 3.

At time t, we set a distribution of values over the vertex set: $\alpha_t : V \cup V_s \to \mathbb{R}$.

We write $P_{V,t} = (\alpha_t(v_i))_{i \in [\![n+1]\!]}$ the column state vector of the vertices at time t, including the special vertex added in the m-uniformisation process. The initialisation is similar to the one done in Chapter 2.

Using the e-adjacency tensor proposed in Chapter 3, the information between vertices can be shared directly from one vertex to another group of vertex by using this tensor. In this approach, the state at time $t+1$ of the vertices is retrieved using the state at time t, using the multilinear matrix multiplication:

$$\lambda P_{V,t+1}^{[r_{\mathfrak{H}}-1]} = \boldsymbol{A} P_{V,t}^{r_{\mathfrak{H}}-1}.$$

The left part is due to the fact that we want to keep homogeneity of the dimension—like it is achieved in dimensional equations in physics—and the presence of the coefficient is due to a normalisation that has to take place. Searching the stable state of this equation is then equivalent to finding the H-eigenvectors of $\boldsymbol{A}$ associated to the eigenvalue λ. It also imposes to consider that the diffusion is the one of the m-uniformized structure, which we know it is far from being perfect from Chapter 4.

The intermediate state of the hb-edges cannot be retrieved in this case; however, it is always possible, either to consider the dual of the hb-graph and its e-adjacency tensor

(in this case the vertices correspond to the original hb-graph) or to use the incident matrix as it was done in Chapter 2.

5.3.2. Taking into account the different layers of uniformity.

5.3.2.1. Tensor approach

Keeping in mind that the diffusion is disturbed for general non-uniform hb-graphs, we can start by keeping the previous approach of tensor Laplacians.

We consider as usual a hb-graph $\mathfrak{H} = (V, \mathfrak{E})$ with vertex set $V = \{v_i : i \in [\![n]\!]\}$ and hb-edge family $\mathfrak{E} = (\mathfrak{e}_j)_{j\in[\![p]\!]}$.

Similarly to the approach taken for uniform hypergraph, we define the **Laplacian hypermatrix** as:

$$\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{A}$$

where $\boldsymbol{D}$ corresponds to the diagonal hypermatrix of the m-degrees of vertices of order $r_{\mathfrak{H}}$, including the one(s) added in the m-uniformisation process. More formally, $\boldsymbol{D}$ is a hypermatrix of order $r_{\mathfrak{H}}$ and dimension $n + n_{\mathcal{A}}$, with $n_{\mathcal{A}}$ the number of added vertices during the m-uniformisation process, defined by:

$$\boldsymbol{D} \triangleq \operatorname{diag}\left((d_{m\,i})_{i\in[\![n+n_{\mathcal{A}}]\!]}\right)$$

with for $i \in [\![n + n_{\mathcal{A}}]\!] : d_{m\,i} \triangleq \deg_m(v_i)$.

Let us consider as in Chapter 3, the attached variable x_i to the vertex v_i with $i \in [\![n]\!]$ and y_i the variable attached to the special vertex Y_i with $i \in [\![n_{\mathcal{A}}]\!]$. We consider $\boldsymbol{z}_{n_{\mathcal{A}}} = (z_i)_{i\in[\![n+n_{\mathcal{A}}]\!]}$ such that: $z_i = x_i$ for $i \in [\![n]\!]$ and $z_{n+i} = y_i$ for $i \in [\![n_{\mathcal{A}}]\!]$.

We define the m-uniformized L-polynomial[2] as:

$$L_{\text{m-u}}(\boldsymbol{z}_{n_{\mathcal{A}}}) \triangleq \sum_{i\in[\![n+n_{\mathcal{A}}]\!]} d_i z_i^{r_{\mathfrak{H}}} - \sum_{i_1,\ldots,i_{r_{\mathfrak{H}}}\in[\![n+n_{\mathcal{A}}]\!]} a_{i_1\ldots i_{r_{\mathfrak{H}}}} \prod_{j\in[\![r_{\mathfrak{H}}]\!]} z_{i_j}.$$

It holds:

$$\boldsymbol{L}\boldsymbol{z}_{n_{\mathcal{A}}}^{r_{\mathcal{H}}} = L_{\text{m-u}}(\boldsymbol{z}_{n_{\mathcal{A}}}).$$

The impact of the m-uniformisation on this Laplacian relies on the exponent of the variables in the degree part and in the coefficients of the built tensor, as well as the additional variables induced by the m-uniformisation.

5.3.2.2. Polynomial approach

A-, D-, L- and Q-polynomials We now consider the decomposition of $\mathfrak{H} = (V, \mathfrak{E})$ in different layers of m-uniform hb-graphs using the decomposition operation 3.3.6.

[2] We cannot use the term Laplacian polynomial as this is reserved for the characteristic polynomial of the Laplacian matrix for graphs.

We define $\mathfrak{E}_k \triangleq (\mathfrak{e} : \mathfrak{e} \in \mathfrak{E} \wedge \#_m \mathfrak{e} = k)$ as the family of hb-edges of $\mathfrak{H}$ of m-cardinality k, eventually empty, for $k \in [\![r_{\mathfrak{H}}]\!]$ and consider, for all $k \in [\![r_{\mathfrak{H}}]\!]$, the k-m-uniform hb-graph $\mathfrak{H}_k \triangleq (V, \mathfrak{E}_k)$. It holds: $\mathfrak{H} = \bigoplus_{k \in [\![r_{\mathfrak{H}}]\!]} \mathfrak{H}_k$.

Each of the k-m-uniform hb-graph $\mathfrak{H}_k$ is associated to a k-adjacency tensor $\boldsymbol{A}_k$ of order k and dimension n.

The corresponding signed Laplacian is: $\boldsymbol{L}_k \triangleq \boldsymbol{D}_k - \boldsymbol{A}_k$ where $\boldsymbol{D}_k = \text{diag}\left(\left(d_{m(k)i}\right)_{i \in [\![n]\!]}\right)$ where for $k \in [\![r_{\mathfrak{H}}]\!]$ and $i \in [\![n]\!]$, $d_{m(k)i}$ corresponds to the m-degree of the vertex i in the k-th hb-graph $\mathfrak{H}_k$.

The unsigned Laplacian is:$\boldsymbol{L}_k \triangleq \boldsymbol{D}_k + \boldsymbol{A}_k$.

We then define, for each $k \in [\![r_{\mathfrak{H}}]\!]$, four polynomials, with $\boldsymbol{z}_0 = (z_1, ..., z_n)$ as defined in Section 5.3.2.1:

- $D_k(\boldsymbol{z}_0) \triangleq \sum_{i \in [\![n]\!]} d_{m(k)i} z_i^k$;
- $A_k(\boldsymbol{z}_0) \triangleq \sum_{i_1,...,i_k \in [\![n]\!]} a_{(k)i_1...i_k} \prod_{j \in [\![k]\!]} z_{i_j}$;
- $L_k(\boldsymbol{z}_0) \triangleq D_k(\boldsymbol{z}_0) - A_k(\boldsymbol{z}_0)$;
- $Q_k(\boldsymbol{z}_0) \triangleq D_k(\boldsymbol{z}_0) + A_k(\boldsymbol{z}_0)$.

and we define four families of polynomials for a general hb-graph $\mathfrak{H}$ as:

- the **D-polynomial family**: $D_w(\boldsymbol{z}_0) \triangleq \sum_{k \in [\![r_{\mathfrak{H}}]\!]} w_k D_k(\boldsymbol{z}_0)$;
- the **A-polynomial family**: $A_w(\boldsymbol{z}_0) \triangleq \sum_{k \in [\![r_{\mathfrak{H}}]\!]} w_k A_k(\boldsymbol{z}_0)$;
- the **L-polynomial family**: $L_w(\boldsymbol{z}_0) \triangleq D_w(\boldsymbol{z}_0) - A_w(\boldsymbol{z}_0)$;
- the **Q-polynomial family**: $Q_w(\boldsymbol{z}_0) \triangleq D_w(\boldsymbol{z}_0) + A_w(\boldsymbol{z}_0)$.

where the coefficients $w = (w_k)_{k \in [\![r_{\mathfrak{H}}]\!]}$ are weights given to each layer potentially with different values.

Remark 5.3.1. *Using here the $\overline{k}$-adjacency hypermatrix defined in Section 3.3.5, we can formulate the reduction of $A_k(\boldsymbol{z}_0)$ as:*

$$A_k(\boldsymbol{z}_0) \triangleq \sum_{\substack{\mathfrak{e}_j \in \mathfrak{E}_k \\ e_j = \left\{ v_{i_1}^{m_{ji_1}}, ..., v_{i_{k_j}}^{m_{ji_{k_j}}} \right\}}} k \prod_{l \in [\![k_j]\!]} z_{i_l}^{m_{ji_l}}.$$

Remark 5.3.2. *Different strategies can be foreseen for weights. The first approach consists in choosing $w = (1)_{k \in [\![r_{\mathfrak{H}}]\!]}$: this puts on an equal footing every hb-edge, whatever its m-cardinality.*

If we want to have a polynomial that is similar in a sense to the m-uniformized Laplacian tensor polynomial, the choice of $w = (c_k)_{k \in [\![r_{\mathfrak{H}}]\!]}$ is the most relevant.

A family of Laplacian hypermatrices Whatever the weights chosen, an homogenized hb-graph L-polynomial can be obtained directly from the L-polynomial by considering:

$$L_{hw}\left(\boldsymbol{z}_1\right) \triangleq y_0^{r_{\mathfrak{H}}} L_w\left(\frac{\boldsymbol{z}_0}{y_0}\right).$$

In this case, we can store the information into a hypermatrix, which is a **layered Laplacian**: $\boldsymbol{L}_{hw}$ defined as:

$$\boldsymbol{L}_{hw}\boldsymbol{z}_1^{r_{\mathfrak{H}}} \triangleq L_{hw}\left(\boldsymbol{z}_1\right).$$

The same can be done with the Q-, D- and A- polynomials, defining:

- $\boldsymbol{Q}_{hw}$: $\boldsymbol{Q}_{hw}\boldsymbol{z}_1^{r_{\mathfrak{H}}} \triangleq Q_{hw}\left(\boldsymbol{z}_1\right)$ with $Q_{hw}\left(\boldsymbol{z}_1\right) \triangleq y_0^{r_{\mathfrak{H}}} Q_w\left(\frac{\boldsymbol{z}_0}{y_0}\right)$
- $\boldsymbol{A}_{hw}$: $\boldsymbol{A}_{hw}\boldsymbol{z}_1^{r_{\mathfrak{H}}} \triangleq A_{hw}\left(\boldsymbol{z}_1\right)$ with $A_{hw}\left(\boldsymbol{z}_1\right) \triangleq y_0^{r_{\mathfrak{H}}} A_w\left(\frac{\boldsymbol{z}_0}{y_0}\right)$
- $\boldsymbol{D}_{hw}$: $\boldsymbol{D}_{hw}\boldsymbol{z}_1^{r_{\mathfrak{H}}} \triangleq D_{hw}\left(\boldsymbol{z}_1\right)$ with $D_{hw}\left(\boldsymbol{z}_1\right) \triangleq y_0^{r_{\mathfrak{H}}} D_w\left(\frac{\boldsymbol{z}_0}{y_0}\right)$.

The advantage of these layered Laplacians (signed and unsigned) is that they take into account each of the layers of m-uniformity contained in the hb-graph.

Some remarks on the polynomials There are two parts in a L-polynomial of a general hb-graph: one that is related to the vertex degrees in the different layers corresponding to the D-polynomial and the second part to the adjacency in the layers of the hb-graph and corresponding to the A-polynomial. Having a good comprehension of the role of each part is fundamental.

These polynomials, and particularly the A-polynomial, opens the door to all the results on multivariate polynomials. Particularly, it could be used for hb-graph reduction using the well-known techniques for multivariate polynomials[3], and to define and find normal forms of a set of hb-graphs, i.e. hb-graphs that cannot be reduced to a given set of hb-graphs.

A-polynomial for hypergraphs are already used, without the m-cardinality coefficient for searching hypergraph extremal problems, i.e. determining the maximal number of hyperedges a hypergraph should have to prevent a forbidden sub-hypergraph to occur; in this case, the Lagrangian used is defined as the polynomial sum of the hb-edge monomials.

We conjecture that finding the stationary state of the exchange-based diffusion for a connected hb-graph as we have done in Chapter 6 with a matrix approach can be reformulated here as the search for the value closest to zero of the L-polynomial under the constraints that $\sum_{i\in[\![n]\!]} z_i = 1$, with for all $i \in [\![n]\!] : 0 < z_i \leqslant 1$ and with equi-weights.

We then transform the problem into a constrained optimization problem, where a lot of techniques are available.

[3]https://www.usna.edu/Users/cs/roche/courses/cs487/mvpoly.pdf

5.3.2.3. First properties of the layered Laplacian hypermatrix

Elements of the layered Laplacian hypermatrix We write:

$$\boldsymbol{A}_k = \left(a_{(k)i_1\dots i_k}\right)_{i_1,\dots,i_k\in[\![n]\!]}.$$

The non-zero elements of $\boldsymbol{A}_k$ correspond to hb-edges of $\mathfrak{H}$ of m-cardinality k. Such an hb-edge $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}},\dots,v_{j_{k_i}}^{m_{ij_{k_i}}}\right\} \in \mathfrak{E}_k$ is stored in $\boldsymbol{A}_k$—as we have already mentioned it in Section 3.3.3—in $\dfrac{k!}{m_{ij_1}!\dots m_{ij_l}!}$ elements:

$$a_{(k)\sigma\left(j_1^{m_{ij_1}},\dots,j_l^{m_{ij_l}}\right)} = \frac{m_{ij_1}!\dots m_{ij_l}!}{(k-1)!}$$

where $\sigma \in \mathcal{S}_{[\![k]\!]}$[4].

Hence, the total value of the elements of $\boldsymbol{A}_k$ associated to $\mathfrak{e}_i = \left\{v_{j_1}^{m_{ij_1}},\dots,v_{j_{k_i}}^{m_{ij_{k_i}}}\right\} \in \mathfrak{E}_k$ is k.

When coming to $\boldsymbol{A}_{hw} = \left(a_{i_1\dots i_{r_\mathfrak{H}}}\right)_{i_1,\dots,i_{r_\mathfrak{H}}\in[\![n+1]\!]}$, the only non-zero elements which are the ones corresponding to $\mathfrak{e}_i$ have to be stored in $n_{\mathfrak{e}_i} = \dfrac{r_\mathfrak{H}!}{m_{ij_1}!\dots m_{ij_l}!\,(r_\mathfrak{H}-k)!}$ elements, considering that the missing indices will correspond to $r_\mathfrak{H}-k$ times the $n+1$-th place, i.e. the special vertex place.

It holds:

$$a_{\sigma_{(r_\mathfrak{H})}\left(j_1^{m_{ij_1}},\dots,j_l^{m_{ij_l}}[n+1]^{r_\mathfrak{H}-k}\right)} = \frac{k}{n_{\mathfrak{e}_i}},$$

i.e.:

$$a_{\sigma_{(r_\mathfrak{H})}\left(j_1^{m_{ij_1}},\dots,j_l^{m_{ij_l}}[n+1]^{r_\mathfrak{H}-k}\right)} = \frac{1}{\begin{pmatrix} r_\mathfrak{H} \\ k \end{pmatrix}} \times a_{(k)\sigma_{(k)}\left(j_1^{m_{ij_1}},\dots,j_l^{m_{ij_l}}\right)}$$

where $\sigma_{(i)} \in \mathcal{S}_{[\![i]\!]}$, with $i = k$ or $r_\mathfrak{H}$.

The non-zero elements of $\boldsymbol{D}_k = \left(d_{(k)i_1\dots i_k}\right)_{i_1,\dots,i_k\in[\![n]\!]}$ corresponds to elements of the diagonal. It holds, for $i \in [\![n]\!]$:

$$d_{(k)i^k} = d_{m(k)i}.$$

In $\boldsymbol{D}_{hw} = \left(d_{i_1\dots i_{r_\mathfrak{H}}}\right)_{i_1,\dots,i_{r_\mathfrak{H}}\in[\![n+1]\!]}$, $d_{(k)i^k}$ is stored in the elements with indices that are obtained by permutation of the elements of the multiset of indices $\left\{i^k,[n+1]^{r_\mathfrak{H}-k}\right\}$, i.e. in one of the $\begin{pmatrix} r_\mathfrak{H} \\ k \end{pmatrix}$ elements $d_{\sigma_{(r_\mathfrak{H})}\left(i^k[n+1]^{r_\mathfrak{H}-k}\right)}$.

Hence:

$$d_{\sigma_{(r_\mathfrak{H})}\left(i^k[n+1]^{r_\mathfrak{H}-k}\right)} = \frac{1}{\begin{pmatrix} r_\mathfrak{H} \\ k \end{pmatrix}} d_{(k)i^k}.$$

[4] $\mathcal{S}_{[\![k]\!]}$ designates the set of permutations on $[\![k]\!]$.

The only non-zero elements of $\boldsymbol{D}_{hw}$ have indices in $\left\{i^k, [n+1]^{r_{\mathfrak{H}}-k}\right\}$ for $k \in [\![r_{\mathfrak{H}}]\!]$.

We write $\boldsymbol{L}_{hw} = \left(l_{i_1\dots i_{r_{\mathfrak{H}}}}\right)_{i_1,\dots,i_{r_{\mathfrak{H}}} \in [\![n+1]\!]}$.

It holds:

$$l_{i_1\dots i_{r_{\mathfrak{H}}}} = d_{i_1\dots i_{r_{\mathfrak{H}}}} - a_{i_1\dots i_{r_{\mathfrak{H}}}}.$$

and similarly, writing $\boldsymbol{Q}_{hw} = \left(q_{i_1\dots i_{r_{\mathfrak{H}}}}\right)_{i_1,\dots,i_{r_{\mathfrak{H}}} \in [\![n+1]\!]}$.

It holds:

$$q_{i_1\dots i_{r_{\mathfrak{H}}}} = d_{i_1\dots i_{r_{\mathfrak{H}}}} + a_{i_1\dots i_{r_{\mathfrak{H}}}}.$$

Remark 5.3.3. *The diagonal of $\boldsymbol{D}_{hw}$ and $\boldsymbol{L}_{hw}$ have always a non-zero element; otherwise, the m-rank would not be $r_{\mathfrak{H}}$.*

Remark 5.3.4. *When the hb-graph is a uniform hypergraph, we retrieve the classical Laplacian: in this case, there are only zero elements on the $n+1$-th component of each face. The classical Laplacian for a uniform hypergraph is then retrieved by extracting the first n-th components of each face.*

Remark 5.3.5. *The L-polynomial attached to the hb-graph can be retrieved from $\boldsymbol{L}_{hw}$ by considering $\boldsymbol{L}_{hw}\boldsymbol{z}_1^{r_{\mathfrak{H}}}$ with $y_0 = 1$.*

The same occurs with A-, Q- and D-polynomials.

Some additional comments on the layered Laplacian hypermatrix If the hb-graph is m-uniform, the elements of the n+1-th row of each face of the layered Laplacian hypermatrix contain only zero elements, i.e. the layered Laplacian reduces by withdrawing those rows to the Laplacian that would have been obtained directly derived from the $\overline{k}$-adjacency hypermatrix as defined by [CD12].

This layered hypermatrix is symmetric and methods included in [QL17] to study such hypermatrices are available.

We keep as remaining research questions the following—non-exhaustive—research questions:

- What are the links between the hb-graph layered Laplacian eigenvalues and the hb-graph features such as the algebraic connectivity?
- How do we define the algebraic connectivity for hb-graphs?
- How does the diffusion process based on this layered Laplacian evolve with time?
- How can we use this layered Laplacian for clustering?
- Can we evaluate the perturbation on the exchange-based diffusion from the L-polynomial before uniformisation?

6. Hb-graph framework

Highlights

This chapter is based on [OLGMM18d] and on [OLGMM19b], which has been accepted to SOFSEM 2020 as [OLGMM20].

Prerequisites: Section 1.3.

6.1. Motivation

Having insight into non-numerical data calls for the gathering of instances: classically (multi-entry) frequency arrays of occurrences are used. To get further insight into data instances of a given type, one can regroup them using their links to instances of another type—used as reference. It generates a family of co-occurrences that can be viewed as a facet of the information space. Navigating across the different facets is achieved by iterating this process between different types of interest while keeping the same reference type: any of these types can be used as a reference. For instance, in scientific publications, different information are linked in an article: the article reference, the authors, the main keywords... All this metadata can potentially give insights into the information space and can be chosen as reference to build co-occurrences. Choosing as reference for instance the article reference, facets depict co-occurrence networks, either of homogeneous type, such as co-authors or co-keywords, or of heterogeneous types, i.e. combining multiple types together.

We have presented a first model using hypergraphs in [OLGMM18d]. But, as co-occurrences can potentially contain repetitions or require an individual weighting: modeling them requires multisets instead of sets.

Hb-graphs, by definition, introduce a potential refinement for storing information, and thus open up refined visualizations and processings. This observation brings us to the research question of this Chapter:

Research question 6.1. *Are hb-graphs pertinent to achieve interactive navigation and visualisation of facets in an information space?*

We start by giving the related work that motivates this approach, before fixing the assumptions and expectations on the framework. We then propose a mathematical model, that we evaluate based on our expectations. We use a publication dataset as a breadcrumb trail example.

6.2. Related work

Discovering knowledge in an information space requires to gather meaningful information, either hierarchically or semantically organized. Semantic provides support to the definition of facets within an information space [Ran62].

The idea of visualizing search query results using graphs has been soon exposed in [CK97]. There are several proposals in the literature to explore large datasets by using graph—possibly multipartite—representations instead of the verbatim presentation.

In [vHP09], the authors present an interaction model to allow the user to interact directly with the local context graph of a topic of interest: they extend to graphs the degree of interest introduced in [Fur86] for fish eye visualisation to focus on the vertices that are close enough from the subject the surfer is looking for taking into account the a priori degree of interest and the distance to the a priori, but also the user interest. The surfer performs a query and retrieves a list of subjects in a verbatim mode. The a priori is performed from the initial query, and the dragging of one of the subject to the graph part of the visualisation marks the user interest and the local context graph is built. Faceted search is also enabled when it is possible. The graph is based on this degrees of interest.

Previous approaches using a reference to articulate the different facets of an information space exist. In [NJY+10], the authors present a 2D multifaceted representation of an information space by using multi-relational graph combined with an optimized density map to highlight patterns and switching between the different facets. The facets are considered as classes of entities: the entities are interconnected using a similarity graph of the extracted entities. Facets are then superposed to reach a global facet visualisation. Similar entities are regrouped using an optimized density map. It is applied to two studies, one on HIV infection and the other one on Diabetes. It allows to visualize global and local relations. There is interactivity and filtering.

In [DRRD12], the authors give the possibility to change of reference between three facets simultaneously shown by using one of them as a pivot; this pivot makes the connection between a facet and the two others, but is limited to a small amount of pivots to be visualized at the same time. In [JCCB13], an interactive exploration of implicit and explicit relations in faceted datasets is proposed. The space of visualisation is shared between different metadata with cross findings between metadata, partitioning the space in categories. In [ADLGP17], the authors propose a visual analytics graph-based framework that reveals this information and facilitates knowledge discovery. It provides insights into different facets of an information space based on user-selected perspectives. The dataset is stored as a labeled graph in a graph database. Choosing a perspective as reference and a facet as dimension, paths of the labeled graph are retrieved with the same dimension extremities going through reference vertices. Visualisation

comes in the form of navigable node-link graphs: edges materialize common references between vertices and are seen as pairwise collaboration between two vertices.

[TCR10] shows how the keeping of multi-adic relationships can help in gaining understanding in the network evolution. In [OLGMM17b], we have shown the interest of using hypergraphs to simplify the visualisation of co-occurrence networks.

In this chapter, we provide the elements for a hb-graph-based framework that supports interactions between the different facets of an information space for optimal knowledge discovery. The dataset—mostly textual—refers to physical entities with unique individual references. Data instances are attached to metadata instances. We suppose that there is no metadata instance that does not have a data instance attached to it.

6.3. Modeling co-occurrences in datasets

Starting by giving the axioms and postulates needed for the framework, we then formulate the expectations on the framework. After giving the mathematical model, we check that the expectations are fulfilled. We finally give an example of implementation of the framework.

6.3.1. Axioms and postulates on the information space

We consider an information space, constituted of physical entities. Physical entities are described by metadata to which correspond data instances stored in a database. Metadata instances have types that are numerical, textual, or others such as pointers to images or videos or sound records. Metadata can be grouped by kinds. For instance, in a publication database, the physical entity corresponds to a publication, which has a publication id, and some attached metadata such as organizations, keywords,... In a company database, the physical entity would be the company, with a company id, and some attached metadata such as economic activity, number of employees, countries, suppliers,...

We formulate the following axioms—in the sense of common propositions considered as universally true—and postulates—in the sense of common propositions that capture the specificity of a structure, that are discussable, but asserted here as true—on the information space and we will work in this shaped information space.

Axiom 6.1. *The information space is constituted of physical entities that are described by metadata and data.*

This is a strong axiom that is needed to ensure a navigable schema: the physical entities ensure the link between the different facets of the information space.

Axiom 6.2. *Each physical entity has a unique identification.*

This assumption is linked to the previous one and ensures that we do not have duplicated information.

Axiom 6.3. *The types associated to metadata can be of varied sorts: numerical, textual, links, pointers to resources,...*

This axiom is not restrictive in fact, as it allows data to be of various types.

Postulate 6.1. *A facet of the information space is the simultaneous choice of:*

- *one or multiple types of the metadata used to build co-occurrences of this(these) metadata type(s), called visualisation type(s),*
- *and of one or multiple types—eventually the same than the ones used for the co-occurrences—called reference type(s).*

This postulate defines a facet of an information space, as a perspective on the dataset that enhances homogeneous—or even heterogeneous—co-occurrence networks built using one—or multiple—references. Facets with homogeneous co-occurrences built with only one type of references are called simple facets.

Postulate 6.2. *The information space can be decomposed into at least three simple facets—including the one of physical entities id.*

This postulate can be relaxed to two facets, but in this case no navigation is possible or even to one facet that corresponds to the facet of physical entities id. In the latter, the visualisation will show only hb-graphs in which hb-edges represent ids and have support reduced to singleton vertices.

Postulate 6.3. *There are at least two kinds of metadata which are either nonnumerical, or that can be represented into categories.*

Relaxing this postulate to a single kind of metadata disables navigation between the different facets: it is always possible, even if it is not desirable. As any numerical data can be treated in categories, the framework can accept any kind of data until they are represented as text or categories.

6.3.2. Hypothesis and expectations of the hb-graph framework

The hypothesis we formulate, related to the research question 6.1, is the following:

Hypothesis 6.1. *Hb-graphs are pertinent to achieve a refined hypergraph framework that supports interactive navigation and visualisation of facets in an information space, including the support of multiple references and heterogeneous co-occurrences.*

To check this hypothesis, we refine it in expectations of what the theoretical framework should answer. The fulfillment of the expectations will allow to validate Hypothesis 6.1: hence, the visual querying of an information space and the browsing inside the information space by finding the links of co-occurrences inside a dataset will be validated on a theoretical basis.

Expectation 6.1. *The framework should model the information space using hb-graphs.*

Hb-graphs not only have like hypergraphs the capacity to handle multi-adic relationships—when graphs can handle only pairwise relationships—, but also individual hb-edge based multiplicities of vertices. Multiple authors—[New01b, TCR10]—acknowledge the importance of modeling co-occurrence networks by hypergraphs, as it enhances higher order relationships than the pairwise model enables. As we have already underlined it, the database schema can be modeled, since [FMU82], using hypergraphs which are anyway a particular case of hb-graphs. Handling multisets allow to store additional information, either with a view to reflecting the vertex multiplicity or individual weighting.

Expectation 6.2. *The framework should provide the visualisation of co-occurrences as hb-graph representations.*

This expectation is linked with the previous one. Pairwise relationships hide a part of the information in many cases as it is shown in [OLGMM17b].

Expectation 6.3. *The framework should enable navigation and interaction between the different facets of the information space.*

This expectation is fundamental as we want to have a full browsing of the information space, thereby allowing to query the dataset in all sort of manners.

Expectation 6.4. *The framework should enhance the visual querying of the information space.*

We want a self contained browsing space that allows to query and display co-occurrences that are retrieved from the dataset.

Expectation 6.5. *The zooming in the dataset should be possible through restrictions to sub-hb-graphs.*

Selecting a subset of the vertices displayed should allow to view their relations with the other facets.

Expectation 6.6. *Heterogeneous co-occurrences should be possible.*

This expectation is needed as the visualization of different facets on a single facet is then possible. It could have already been introduced with hypergraphs. To reach this expectation, the visualisation must be achieved by an extra-node representation of the hb-graph, as the 2-section of the hb-graph would lead to hazardous interpretation.

Expectation 6.7. *Multiple types of references should be possible.*

This expectation is needed to support co-occurrences built from references of different types that constitute a partition of the dataset.

METADATA	Schema hypergraph ↓	Related to database structure	$\mathcal{H}_{\text{Sch}} = (V_{\text{Sch}}, E_{\text{Sch}})$
	Extended schema hypergraph ↓	Store possible additional processings	$\overline{\mathcal{H}_{\text{Sch}}} = \left(\overline{V_{\text{Sch}}}, \overline{E_{\text{Sch}}}\right)$
	Extracted extended schema hypergraph ↓	U: set of metadata of interest (visualisation and reference)	$\mathcal{H}_X = (V_X, E_X)$ where $V_X = U$, $E_X = \left\{e \cap U : e \in \overline{E_{\text{Sch}}}\right\}$
	Reachability hypergraph ↙↓↘	Hyperedges are connected components $E_{\text{cc}} (\subset V_X)$ of $\mathcal{H}_X$	$\mathcal{H}_R = (V_R, E_R)$ $V_R = V_X$ $E_R = \{E_{\text{cc}} : E_{\text{cc}} \text{ c.c. of } H_X\}$
	Navigation hypergraph ↓	Choose: $e_r \in E_R$ references $R_{\text{ref}} \subset e_r$	$\mathcal{H}_N = (V_N, E_N)$ $V_N = V_R \backslash R_{\text{ref}}$ $E_N = \{e_r \backslash R : R \subseteq R_{\text{ref}} \wedge R \neq \emptyset\}$
DATA	Facet visualisation hb-graphs	Co-occurrence networks as hb-graphs	

Table 6.1.: Synthesis of the framework.

6.3.3. The hb-graph framework

The hb-graph framework is built on two main parts: one at the metadata level and the other at the data level: the first level enhances navigation while the second one provides the facet visualisation hypergraphs. The enhancement of navigation is achieved by considering different hypergraphs that originate from the database schema that will be defined in Section 6.3.3.1: the schema hypergraph, the extended schema hypergraph, the extracted extended schema hypergraph, the reachability hypergraph and the navigation hypergraph. The different schemata are summarized in Table 6.1. Once the navigation hypergraph is constructed, choosing one of its hyperedge and one or multiple references inside enhances the construction of the different visualisation hb-graphs that will be associated to the different facets of the information space as it will be explained in Section 6.3.3.2. Visualisation hb-graphs are hb-graph of co-occurrences built from the data instances that are linked to the reference instances.

To illustrate the different aspects of the hb-graph framework, we take as thumbnail an example based on a publication dataset. In this case, the possible metadata types are: *publication id*, title, abstract, *authors*, affiliations, addresses, *author keywords*, *publication categories*, *countries*, *organizations*, and eventually some processed metadata types—i.e. corresponding to data not directly contained in the publication dataset—such as *processed keywords*, *continent*,...[1]

[1]Metadata of interest for visualisation or referencing are in italic

6.3.3.1. Enhancing navigation

Relational database schemata are hypergraphs of metadata instances where the hyperedges gather table metadata: normalized forms are linked to the properties of the hypergraphs modeling them—[Fag83]. In graph databases, the schema[2] represents the relationships between the vertex types. The **schema hypergraph** $\mathcal{H}_{\text{Sch}} = (V_{\text{Sch}}, E_{\text{Sch}})$ represents these relationships as hyperedges.

The schema hypergraph is possibly enriched with additional processings: the **extended schema hypergraph** $\overline{\mathcal{H}_{\text{Sch}}} = \left(\overline{V_{\text{Sch}}}, \overline{E_{\text{Sch}}}\right)$ stores this information.

Each data instance stored in the dataset is labeled using a labeling function applied to the vertices of $\overline{V_{\text{Sch}}}$. Hyperedges of the extended schema itself can be labeled by another labeling function using another set of labels.

Types of visual or referencing interest are selected in a subset U of $\overline{V_{\text{Sch}}}$ to generate $\mathcal{H}_X = (V_X, E_X)$, the **extracted extended schema hypergraph**, where $V_X = U$ and $E_X = \left\{e \cap U : e \in \overline{E_{\text{Sch}}}\right\}$.

From $\mathcal{H}_X$, we build the **reachability hypergraph** $\mathcal{H}_R = (V_R, E_R)$. V_R corresponds to the vertex set of the extracted extended schema hypergraph V_X. The hyperedges of $\mathcal{H}_R$ are the connected components of $\mathcal{H}_X$. We assume that in each hyperedge of the reachability hypergraph, there is at least one metadata type or a combination of metadata types that can be chosen as the **physical reference**. The data instance related to this reference is supposed to be unique. For instance, in a publication dataset the physical reference is the publication id of the publication itself. In the example, the extracted hypergraph has only one component {publication id, authors, processed keywords, subject categories}, as shown in Figure 6.1.

Each hyperedge $e_r \in E_R$ of $\mathcal{H}_R$ leads to one new **navigation hypergraph** $\mathcal{H}_N = (V_N, E_N)$ by choosing a non-empty subset R_{ref} of e_r of possible reference types of interest. The choice of a subset R of R_{ref} allows to consider the remaining vertices of $e_r \backslash R$ as visualisation vertex types, that will be used to generate the facet visualisation hb-graphs and are called the visualisation types. Hence: $E_N = \{e_r \backslash R : R \subseteq R_{\text{ref}} \wedge R \neq \emptyset\}$. The simplest case happens when there is only one reference of interest selected at a time in R_{ref}; we restrict ourselves to this case for the moment, i.e. we consider for E_N the set $E_{N/1} = \{e_r \backslash R : R \subseteq R_{\text{ref}} \wedge |R| = 1\}$.

In the publication database example, many navigation hyperedges are possible. For instance, the navigation hyperedge when choosing as reference publication ids is {authors, publication categories, processed keywords} while when using processed keywords as reference the navigation hyperedge becomes: {authors, publication category, publication ids}.

6.3.3.2. Visualisation hb-graphs corresponding to facets

In [OLGMM18d], we use sets to store co-occurrences. Nonetheless, in many cases, it is worth storing additional information by attaching a multiplicity—with non-negative integer or real values—to the elements of co-occurrences. A small example emphasizes

[2] although not required [MEP+14]

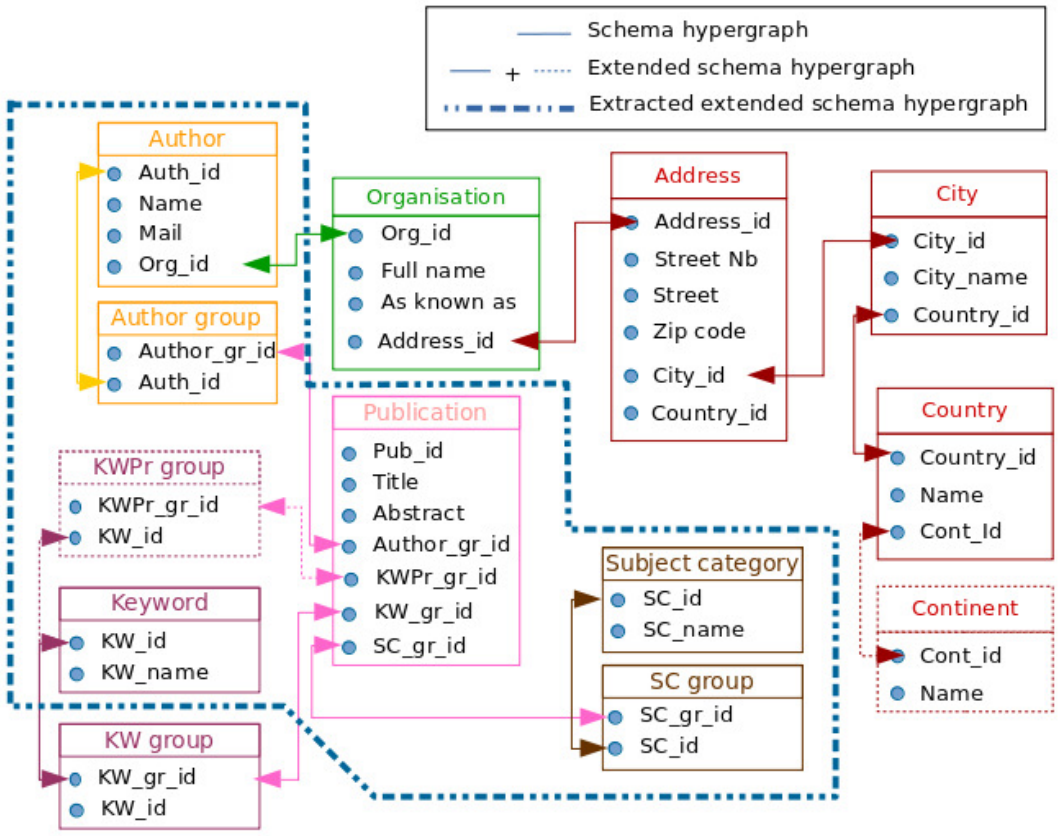


Figure 6.1.: Schema hypergraph, Extended schema hypergraph, Extracted extended schema hypergraph: exploded view shown on an example of publication dataset.

the interest of moving towards multisets: we consider the publication network of Figure 6.2. In this example, building co-occurrences accounting for occurrence multiplicity not only induces a refined visualisation, with distinguishable hb-edges in between some of the vertices (augmented reality and 3D) but also yields to refined rankings of both vertices and hb-edges, as it is also mentioned in [OLGMM19a].

In a dataset $\mathcal{D}$, each physical entity $d \in \mathcal{D}$ corresponds to a unique physical reference r. d is described by a set of data instances of different types that are in $\alpha \in \overline{V_{\text{Sch}}}$. We write $\mathcal{I}$ the set of data instances in $\mathcal{D}$, and t the type application that gives the type of an instance.

Hb-edges must have a common universe to account for the extension of the framework to hb-graphs. We consider for each type α, its instance set:

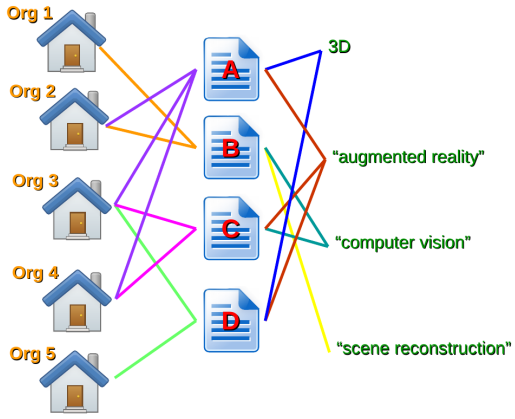
$$U_\alpha = \{i : i \in \mathcal{I} \wedge t\,(i) = \alpha\}$$

of instances of $\mathcal{D}$ of type α.

We write $\mathfrak{A}_{\alpha,r} = (U_\alpha, m_{\alpha,r})$ the multiset of universe U_α, of the values of type α—possibly none—that are attached to d, the physical entity of reference r. The support of $\mathfrak{A}_{\alpha,r}$ is:

$$\mathfrak{A}^\star_{\alpha,r} = \left\{a_{i_1}, ..., a_{i_{k_r}}\right\}.$$

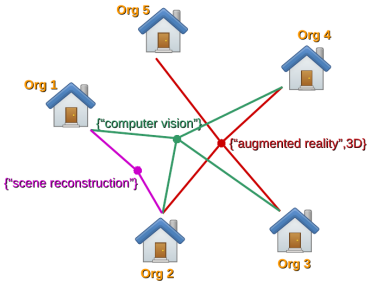
(a) A publication network



(b) Co-occurrences of organizations with keyword as reference

scene reconstruction	$\{\{\text{Org } 1^1, \text{Org } 2^1\}\}$
computer vision	$\{\{\text{Org}1^1, \text{Org}2^1, \text{Org } 3^1, \text{Org } 4^1\}\}$
augmented reality	$\{\{\text{Org } 2^1, \text{Org } 3^3, \text{Org } 4^2, \text{Org } 5^1\}\}$
3D	$\{\{\text{Org } 2^1, \text{Org } 3^2, \text{Org } 4^1, \text{Org } 5^1\}\}$

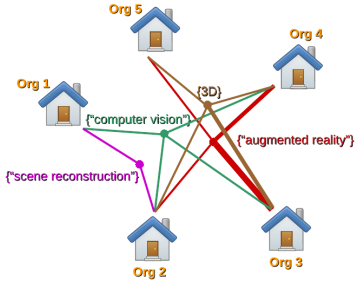
(b.i) Support hypergraph.



Vertex weighted degree ranking:
1. Org 2;
2. Org 3; Org 4;
4: Org 5; Org 1.

Hyperedge weighted cardinality ranking:
1. CV, AR, 3D; 2. SR

(b.ii) Hb-graph.



Vertex weighted m-degree ranking:
1. Org 3;
2. Org 2; Org 4;
4: Org 5; Org 1.

Hb-edge weighted m-cardinality ranking:
1. AR; 2: 3D; 3: CV; 4: SR

Figure 6.2.: A simplified publication network.

Hence, we abusively write:

$$\mathfrak{A}_{\alpha,r} = \left\{ \left\{ a_{i_1}^{m_{\alpha,r}(a_{i_1})}, ..., a_{i_{k_r}}^{m_{\alpha,r}(a_{i_{k_r}})} \right\} \right\}$$

omitting the elements of U_α that are not in the support of $\mathfrak{A}_{\alpha,r}$.

d is entirely described by its reference r and the family of multisets, corresponding to homogeneous co-occurrences of the different types α in $\overline{V_{\text{Sch}}}$ linked to the physical reference, i.e. $\left(r, (\mathfrak{A}_{\alpha,r})_{\alpha \in \overline{V_{\text{Sch}}}}\right)$.

In Figure 6.2, the publication id is the physical reference. Taking as reference the publication id, the co-occurrence of organizations for the Publication A is: $\left\{\text{Org } 2^1, \text{Org } 3^1, \text{Org } 4^1\right\}$ and the one of keywords is: $\left\{3\text{D}^1, \text{augmented reality}^1\right\}$. The example in Figure 6.2 (b) shows a reference that is not the physical reference.

Type heterogeneity in co-occurrences enables simultaneous view of different types in a single facet. To allow type heterogeneity in co-occurrences, we consider a partition Γ_S of the different types in $\overline{V_{\text{Sch}}}$. Each type belonging to an element γ of the partition Γ_S will be visualized simultaneously in a co-occurrence: it enriches the navigation process, enabling heterogeneous co-occurrences. An interesting case is when γ has a semantic meaning and elements of γ appear as an "is a" relationship. For instance, in a publication database, an organization can correspond to "institute" and "company". Also, we consider $\mathfrak{A}_{\gamma,r} = (U_\gamma, m_{\gamma,r})$, where $U_\gamma = \bigcup_{\alpha \in \gamma} U_\alpha$, of support $\mathfrak{A}^\star_{\gamma,r} = \bigcup_{\alpha \in \gamma} \mathfrak{A}^\star_{\alpha,r}$ such that

$$m_{\gamma,r}(a) = \begin{cases} m_{t(a),r}(a) & \text{if } a \in \mathfrak{A}^\star_{\gamma,r}; \\ 0 & \text{otherwise.} \end{cases}$$

d is entirely described in the case of heterogeneous co-occurrences by $\left(r, (\mathfrak{A}_{\gamma,r})_{\gamma \in \Gamma_S}\right)$. The homogeneous co-occurrences are retrieved when all $\gamma \in \Gamma_S$ are singletons.

Performing a search on the dataset $\mathcal{D}$ retrieves a set $\mathcal{S}$ of physical references r. In the single-reference-restricted navigation hypergraph, each hyperedge $e_N \in E_{N/1}$ describes accessible facets relatively to a chosen reference type $\rho \in V_N \backslash e_N$. Given a partition $\gamma \in \Gamma_N$, where $\Gamma_N = \{\gamma \cap e_N : \gamma \in \Gamma_S\}$ is the induced partition of e_N related to the partition Γ_S of $\overline{V_{\text{Sch}}}$, the associated facet shows the visualisation hb-graph $\mathfrak{H}_{\gamma/\rho,\mathcal{S}}$ where the hb-edges are the heterogeneous co-occurrences of types in γ relatively to reference instances of type ρ (γ/ρ for short) retrieved from the different references in $\mathcal{S}$.

We then build the co-occurrences γ/ρ by considering the set of all values of type ρ attached to all the references $r \in \mathcal{S}$: $\Sigma_\rho = \bigcup_{r \in \mathcal{S}} \mathfrak{A}^\star_{\rho,r}$. Each element s of Σ_ρ is mapped to a set of physical references $R_s = \{r : s \in \mathfrak{A}_{\rho,r}\} \in \mathcal{P}(\mathcal{S})$ in which they appear: we write r_ρ the mapping. The multiset of values $\mathfrak{e}_{\gamma,s}$ of types $\alpha \in \gamma$ relatively to the reference instance s is $\mathfrak{e}_{\gamma,s} = \biguplus_{r \in R_s} \mathfrak{A}_{\gamma,r}$.

The **raw visualisation hb-graph** for the facet of heterogeneous co-occurrences γ/ρ attached to the search $\mathcal{S}$ is then defined as:

$$\mathfrak{H}_{\gamma/\rho,\mathcal{S}} \triangleq \left(\bigcup_{r \in \mathcal{S}} \mathfrak{A}^\star_{\gamma,r}, (\mathfrak{e}_{\gamma,s})_{s \in \Sigma_\rho} \right).$$

Since some hb-edges can possibly point to the same sub-mset of vertices, we build a reduced visualisation weighted hb-graph from the raw visualisation hb-graph. To achieve this, we define: $g_\gamma : s \mapsto \mathfrak{e}_{\gamma,s}$ and $\mathcal{R}$ the equivalence relation such that: $\forall s_1 \in \Sigma_\rho$, $\forall s_2 \in \Sigma_\rho$: $s_1 \mathcal{R} s_2 \Leftrightarrow g_\gamma(s_1) = g_\gamma(s_2)$.

Considering a quotient class $\overline{s} \in \Sigma_\rho/\mathcal{R}$[3], we write $\overline{\mathfrak{e}_{\gamma,\overline{s}}} = g_\gamma(s_0)$ where $s_0 \in \overline{s}$.

We consider:

$$\overline{\mathfrak{E}_\gamma} \triangleq \left\{\overline{\mathfrak{e}_{\gamma,\overline{s}}} : \overline{s} \in \Sigma_\rho/\mathcal{R}\right\}$$

and:

$$\mathfrak{E}_\gamma \triangleq \left\{\{\mathfrak{e}_{\gamma,s} : s \in \Sigma_\rho\}\right\}.$$

It holds:

$$\overline{\mathfrak{E}_\gamma} = \mathfrak{E}_\gamma^\star$$

since we have: $\overline{\mathfrak{e}_{\gamma,\overline{s}}} \in \overline{\mathfrak{E}_\gamma}$ of multiplicity $w_\gamma(\overline{\mathfrak{e}_{\gamma,\overline{s}}}) = |\overline{s}|$ in this multiset and thus:

$$\mathfrak{E}_\gamma = \left\{\overline{e_{\gamma,\overline{s}}}^{w_\gamma\left(\overline{e_{\gamma,\overline{s}}}\right)} : \overline{s} \in \mathcal{S}_\rho/\mathcal{R}\right\}.$$

Let $\tilde{g_\gamma} : \overline{s} \in \Sigma_\rho/\mathcal{R} \mapsto \mathfrak{e} \in \overline{\mathfrak{E}_\gamma}$, then $\tilde{g_\gamma}$ is bijective. ${\tilde{g_\gamma}}^{-1}$ allows to retrieve the class associated to a given hb-edge; hence the associated values of Σ_ρ to this class—which will be an important point for navigation. The references associated to $\mathfrak{e} \in \overline{\mathfrak{E}_\gamma}$ are $\bigcup_{s \in {\tilde{g_\gamma}}^{-1}(\mathfrak{e})} r_\rho(s)$. The **reduced visualisation weighted hb-graph** for the search $\mathcal{S}$ is defined as:

$$\mathfrak{H}_{\gamma/\rho,w_\gamma,\mathcal{S}} \triangleq \left(\bigcup_{r \in \mathcal{S}} \mathfrak{A}^\star_{\gamma,r}, \overline{\mathfrak{E}_\gamma}, w_\gamma\right).$$

Using the support hypergraph of the visualisation hb-graphs retrieves the results given in the case of homogeneous co-occurrences in [OLGMM18d]: hence the hypergraph framework appears to be a particular case of the hb-graph framework.

6.3.3.3. Navigability through facets

As for a given search $\mathcal{S}$ and a given reference ρ, the sets Σ_ρ and R_s, $s \in \Sigma_\rho$ are fixed, the navigability can be guaranteed between the different facets. We consider a group of types γ, its visualisation hb-graph $\mathfrak{H}_{\gamma/\rho,w_\gamma}$ and a subset A of the vertex set of $\mathfrak{H}_{\gamma/\rho,w_\gamma}$. We target another group of types γ' of heterogeneous co-occurrences referring to ρ for visualisation. Figure 6.3 illustrates the navigation.

We suppose that the user selects elements of A as vertices of interest from which s·he·x wants to switch facet. The hb-edges of $\overline{\mathfrak{E}_\gamma}$ which contain at least one element of A are gathered in:

$$\overline{\mathfrak{E}_\gamma}\Big|_A = \left\{\mathfrak{e} : \mathfrak{e} \in \overline{\mathfrak{E}_\gamma} \wedge (\exists x \in \mathfrak{e} : x \in A)\right\}.$$

Using the function ${\tilde{g_\gamma}}^{-1}$, we retrieve the corresponding class of references of type ρ associated to the elements of $\overline{\mathfrak{E}_\gamma}\Big|_A$ to build the set of references:

$$\overline{V}\Big|_A = \left\{{\tilde{g_\gamma}}^{-1}(\mathfrak{e}) : \mathfrak{e} \in \overline{\mathfrak{E}_\gamma}\Big|_A\right\}$$

[3] $\Sigma_\rho/\mathcal{R}$ is the quotient set of Σ_ρ by $\mathcal{R}$

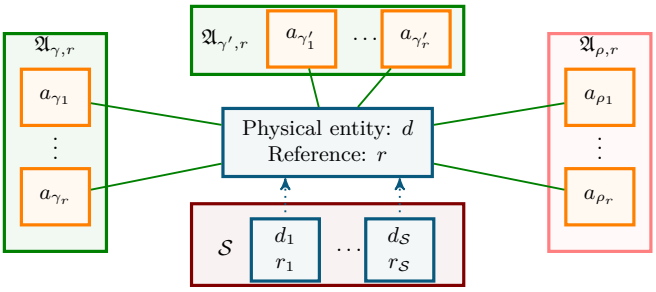


Figure 6.3.: Navigating between facets of the information space.

of type ρ involved in the building of co-occurrences of type γ'. Each of the classes in $\overline{V}\Big|_A$ contains instances of type ρ that are gathered in a set $\mathcal{V}_{\rho,A}$. Each element of $\mathcal{V}_{\rho,A}$ is linked to a set of physical references by r_ρ. Hence we obtain the physical reference set involving elements of A: $\mathcal{S}_A = \bigcup_{s\in\nu_{\rho,A}} R_s$.

The raw visualisation hb-graph $\mathfrak{H}_{\gamma'/\rho}\Big|_A = \left(\bigcup_{r\in\mathcal{S}_A} \mathfrak{A}^\star_{\gamma',r}, (\mathfrak{e}_{\gamma',s})_{s\in\mathcal{V}_{\rho,A}}\right)$ in the targeted facet is now enhanced using $\mathcal{S}_A$ as search set $\mathcal{S}$. To obtain the reduced weighted version we use the same approach as above. The multiset of co-occurrences retrieved includes all occurrences that have co-occurred with the references attached to one of the elements of A selected in the first facet. Of course, if $A = \mathfrak{A}_{\gamma,S}$, the reduced visualisation hb-graph contains all the instances of type γ' attached to physical entities of the search $\mathcal{S}$.

In Figure 6.2.b(ii), with $A = \{\text{Org1}\}$, two hb-edges can be retrieved: computer vision—attached to PubB and PubC—and scene reconstruction—PubB. Hence: $S_A = \{\text{PubB}, \text{PubC}\}$. Switching to the Publication facet and keeping as reference keywords, two hb-edges $\left\{\text{PubB}^1, \text{PubC}^1\right\}$ and $\left\{\text{PubB}^1\right\}$ are retrieved. The same with $A = \{\text{Org1}, \text{Org2}\}$ retrieves all the co-occurrences of Publications with reference to keywords.

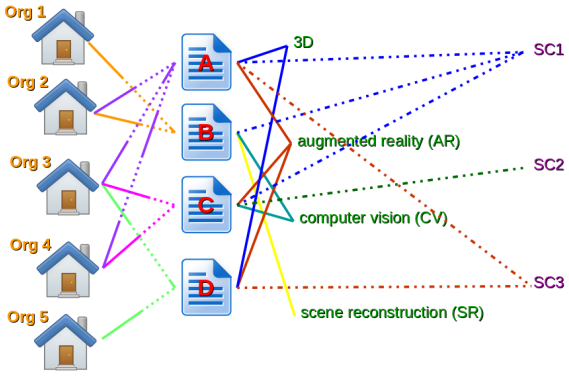
It can be of interest to put a filter on $\mathfrak{H}_{\gamma'/\rho}\Big|_A$ to retain only the references used in the first facet to build the co-occurrences. In this case, we obtain a sub-hb-graph of $\mathfrak{H}_{\gamma'/\rho}\Big|_A$ that is limited to the references of co-occurrences involved in the previous facet.

An other example is given in Figure 6.4. We navigate between organization co-occurrences and the subject category co-occurrences both built using keywords as reference. We select one of the organization in the organization facet. We show how the construction of the hb-graph work for the targeted facet. Additionally, we present how the possible filtering of the targeted hb-graph works and how it restricts the view to the references of the first facet.

The reference type can always be shown in one of the facets as a visualisation hb-graph where all the hb-edges are constituted of the references themselves with multiplicity the number of time the reference occurs in the hb-graph.

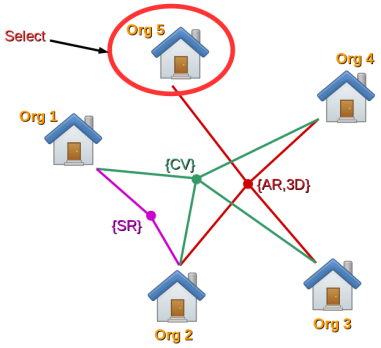
Ultimately, by building a multi-dimensional network organized around groups of types, one can retrieve very valuable information from combined data sources. This process can be extended to any number of data sources as long as they share one or more types.

(a) The publication network with four facets.

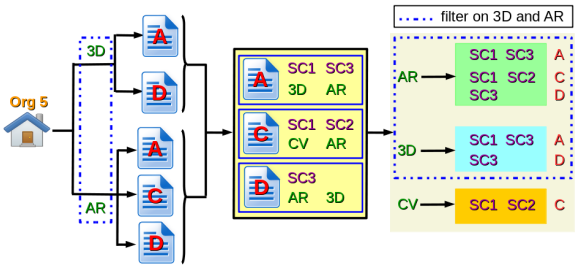


(b) Organization co-occurrences with keyword as reference.

scene reconstruction	Pub B	$\{\{\text{Org } 1^1, \text{Org } 2^1\}\}$
computer vision	Pub B, Pub C	$\{\{\text{Org}1^1, \text{Org}2^1, \text{Org } 3^1, \text{Org } 4^1\}\}$
augmented reality	Pub A, Pub C, Pub D	$\{\{\text{Org } 2^1, \text{Org } 3^3, \text{Org } 4^2, \text{Org}\,5^1\}\}$
3D	Pub A, Pub D	$\{\{\text{Org } 2^1, \text{Org } 3^2, \text{Org } 4^1, \text{Org}\,5^1\}\}$

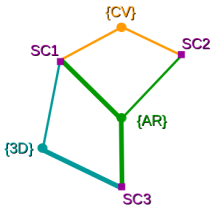


(c) Building subject category co-occurrences with keywords as reference.



(d) Visualization of the subject category co-occurrences with keywords as reference.

(d.i) Unfiltered.



(d.ii) Filtered.

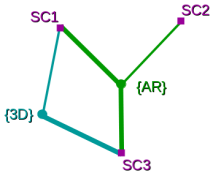


Figure 6.4.: Navigating between facets in the publication network: visualizing organization co-occurrences with reference keywords and switching to subject categories with reference keywords.

Otherwise, the reachability hypergraph is not connected, and only separated navigations are possible in this case.

6.3.3.4. Change of reference

Changing reference follows similar principles than navigating through facets. We consider a group of types γ, its visualisation hb-graph $\mathfrak{H}_{\gamma/\rho,w_\gamma}$ and a subset A of the vertex set of $\mathfrak{H}_{\gamma/\rho,w_\gamma}$. But, this time, we target the same group of types γ of heterogeneous co-occurrences referring to a new reference type ρ' for visualisation.

We still suppose that the user selects elements of A as vertices of interest from which s·he·x wants to switch facet. The hb-edges of $\overline{\mathfrak{E}_\gamma}$, which contain at least one element of A, are again gathered in:

$$\overline{\mathfrak{E}_\gamma}\Big|_A = \left\{\mathfrak{e} : \mathfrak{e} \in \overline{\mathfrak{E}_\gamma} \wedge (\exists x \in \mathfrak{e} : x \in A)\right\}.$$

Using the function $\tilde{g_\gamma}^{-1}$, we retrieve the corresponding class of references of type ρ associated to the elements of $\overline{\mathfrak{E}_\gamma}\Big|_A$, and build the set of references:

$$\overline{V}\Big|_A = \left\{\tilde{g_\gamma}^{-1}(\mathfrak{e}) : \mathfrak{e} \in \overline{\mathfrak{E}_\gamma}\Big|_A\right\}$$

of type ρ involved in the building of co-occurrences of type γ. Each of the classes in $\overline{V}\Big|_A$ contains instances of type ρ that are gathered in a set $\mathcal{V}_{\rho,A}$. Each element of $\mathcal{V}_{\rho,A}$ is linked to a set of physical references by r_ρ. Hence, we obtain the physical reference set involving elements of A :

$$\mathcal{S}_A = \bigcup_{s \in \nu_{\rho,A}} R_s.$$

And, we can now enhance the raw visualisation hb-graph:

$$\mathfrak{H}_{\gamma/\rho'}\Big|_A = \left(\bigcup_{r \in \mathcal{S}_A} \mathfrak{A}^\star_{\gamma,r}, (\mathfrak{e}_{\gamma,s})_{s \in \mathcal{V}_{\rho',A}}\right)$$

with the new reference type using $\mathcal{S}_A$ as search set $\mathcal{S}$. To obtain the reduced weighted version, we use the same approach as before. The multiset of co-occurrences retrieved includes all occurrences that have co-occurred with the references attached to one of the elements of A selected in the first facet.

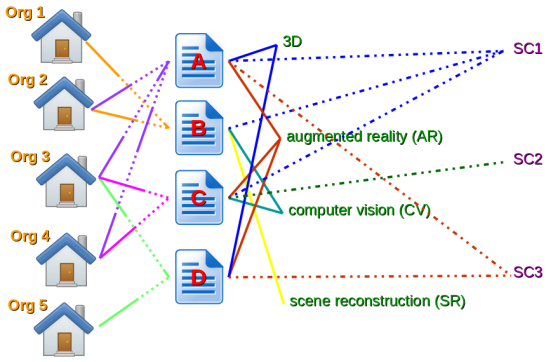
It is worth mentioning that a change of reference from ρ to ρ' may result in additional elements of type γ in the corresponding co-occurrences. This is illustrated on the publication network as shown in Figure 6.5.

6.3.3.5. The case of multiple references

Extending co-occurrences to multiple reference types chosen in $e_R \in E_R$ is not straightforward. There are two ways of doing so: a disjunctive and a conjunctive way. We consider the set $R \subset e_R$ of references and $e_N = e_R \backslash R$ the visualisation types.

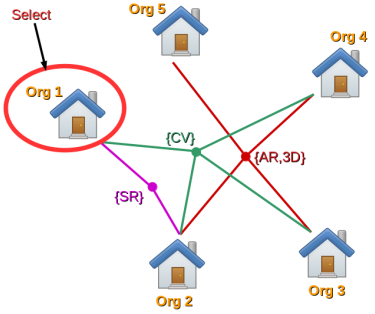
In the disjunctive way, each co-occurrence is built as before, considering successively each type $\rho \in R$. This is particularly adapted for types that are partitioning the physical

(a) The publication network with four facets.

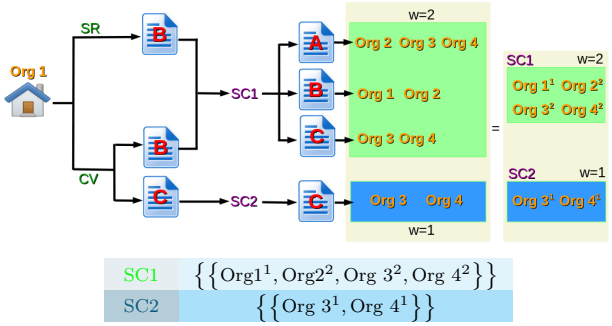


(b) Organization co-occurrences with keyword as reference.

scene reconstruction	Pub B	$\{\{\text{Org } 1^1, \text{Org } 2^1\}\}$
computer vision	Pub B, Pub C	$\{\{\text{Org}1^1, \text{Org}2^1, \text{Org } 3^1, \text{Org } 4^1\}\}$
augmented reality	Pub A, Pub C, Pub D	$\{\{\text{Org } 2^1, \text{Org } 3^3, \text{Org } 4^2, \text{Org}\,5^1\}\}$
3D	Pub A, Pub D	$\{\{\text{Org } 2^1, \text{Org } 3^2, \text{Org } 4^1, \text{Org}\,5^1\}\}$



(c) Building organization co-occurrences with subject categories as reference.



SC1	$\{\{\text{Org}1^1, \text{Org}2^2, \text{Org } 3^2, \text{Org } 4^2\}\}$
SC2	$\{\{\text{Org } 3^1, \text{Org } 4^1\}\}$

(d) Visualization of the organization co-occurrences with subject categories as reference.

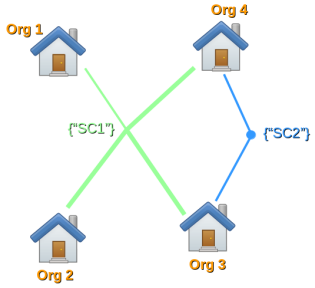


Figure 6.5.: Changing reference in the publication network: visualizing organization co-occurrences with reference keywords and switching to reference subject categories.

references. It is the case for instance in the aggregation of two databases on two different kind of physical data, such as publication and patent, where the co-occurrence of the chosen navigation type is built referring either to a publication or (non-exclusive) to a patent. The hb-graphs obtained are built by extending the family of hb-edges.

In the conjunctive approach, we start by building the cross product of instances of the references and retrieve co-occurrences of elements for which the data d is attached to the corresponding values of the cross-reference instances. Hence, co-occurrences are restricted to the simultaneous presence of reference instances attached to the physical entity.

6.3.4. The DataHbEdron

The **DataHbEdron**—the name we have given to the tool we have developed based on these mathematical principles—provides soft navigation between the different facets of the information space. Each facet of the information space corresponding to a visualisation type includes a visualisation hb-graph viewed in its 2D extra-node representation with a normalised thickness on hb-edges [OLGMM18b]. To enhance navigation, the different facets are embedded in a 2.5D representation called the DataHbEdron. The DataHbEdron[4] can be toggled into either a cube with six faces—Figure 6.6(a)—or a carousel shape with n faces—Figure 6.6(b)—to ease navigation between facets. The reference face shows a traditional verbatim list of references corresponding to the search output.

Faces of the DataHbEdron show different facets of the information space: the underlying visualisation hb-graphs support the navigability through facets. Hb-edges are selectable across the different facets; since each hb-edge is linked to a subset of references, the corresponding references can be used to highlight information in the different facets as well as in the face containing the reference visualisation hb-graph.

6.3.5. Validation and discussion

6.3.5.1. Fulfillment of expectations

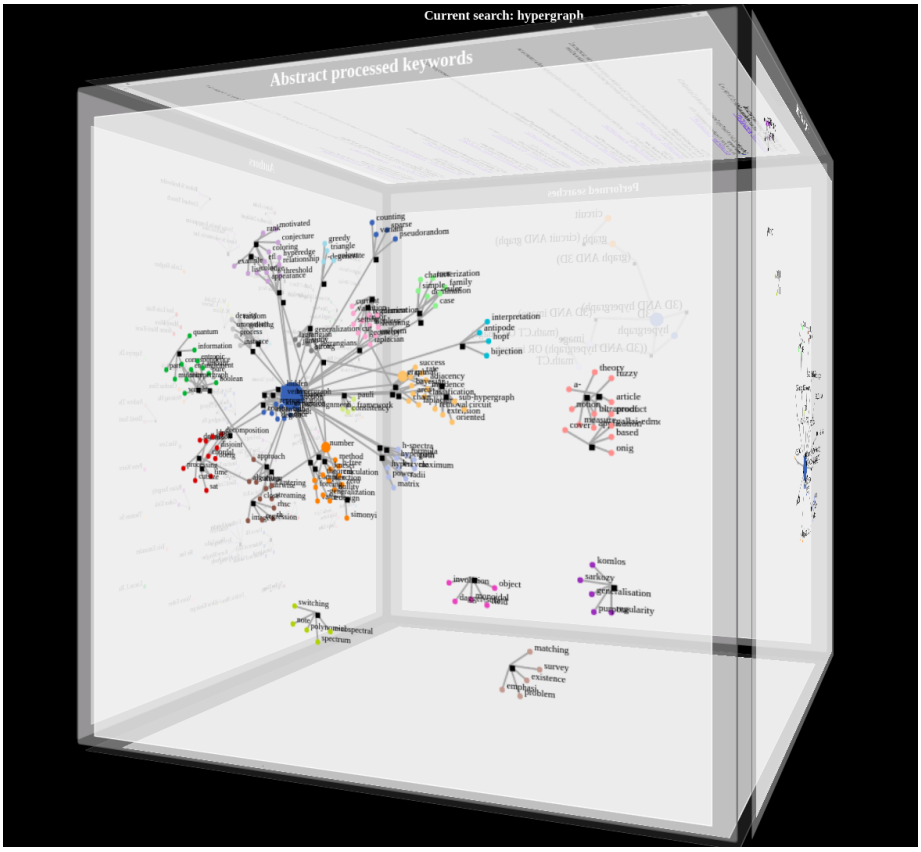
Now that the hb-graph framework has been built, we can check that each expectation expressed in Section 6.3.2 is fulfilled.

Guarantee 6.1. *The framework should model the information space using hb-graphs.*

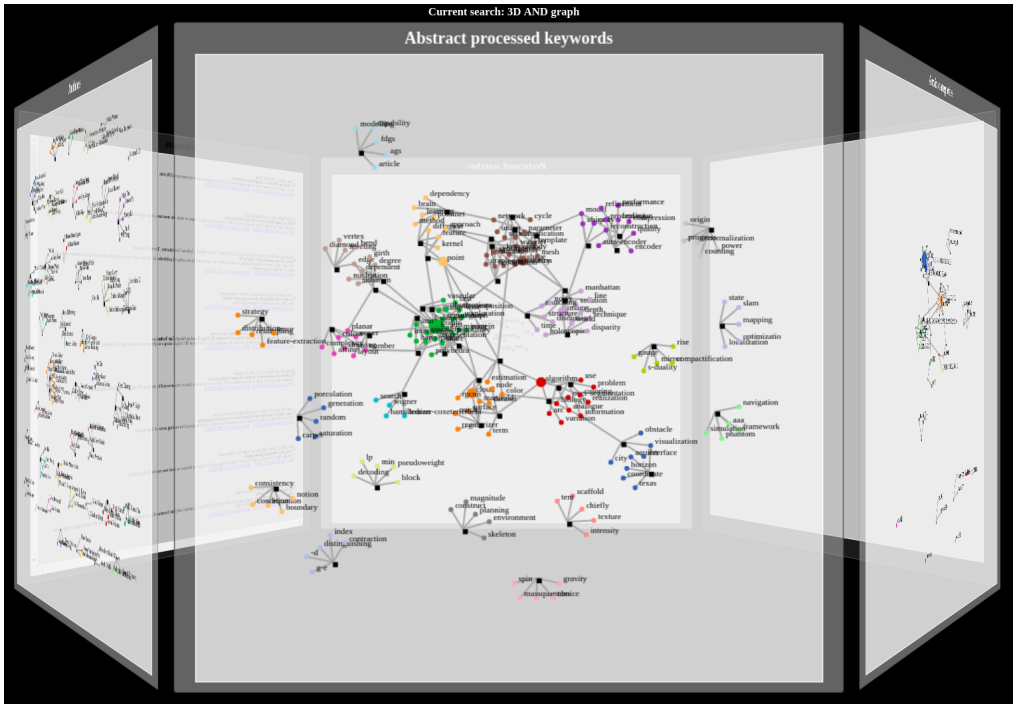
We reach this expectation, either by using hypergraphs—particular case of hb-graphs—at the metadata level or by using hb-graphs at the visualisation level. Even if the database schema can always be modeled by a graph as in [ADLGP17], there are concepts that better fit using hypergraphs to store the schema[5]. The other related work limit themselves to a few facets and the toggling between those facets; moreover, they do not take into account multisets.

[4] An animated presentation can be found on my research page: https://www.infos-informatique.net

[5] A good example is given in https://blog.grakn.ai/modelling-data-with-hypergraphs-edff1e12edf0: if a marriage can be easily modeled by a table that contains an id, a husband id, and a wife id, which is already a triple, a divorce has an id, an id of marriage, a petitioner id and a respondent id, which is a quadruple. In both cases, hypergraphs fit to model this situation.



(a) cube shape.



(b) carousel shape.

Figure 6.6.: DataHbEdron: (a) cube shape (b) carousel shape.

Guarantee 6.2. *The framework should provide the visualisation of co-occurrences as hb-graph representations.*

We build the visualisation hb-graph to show co-occurrences of a metadata type referring to one or multiple chosen metadata references. The framework is intrinsically supporting the concept of hb-graphs as it was done with hypergraphs on a few facets in [DRRD12] by considering multi-partite graphs. In [ADLGP17], the co-occurrences retrieved are systematically transformed in graphs and direct pairwise link.

Guarantee 6.3. *The framework should enable navigation and interaction between the different facets of the information space.*

The navigability in between facets has been detailed in the model, by showing the fundamental role of the references and of the physical references. This link is direct when handling hb-graphs: this is a well known technique found in image retrieval for fusing hypergraphs of different features such as image features and tags to make learning out of it [ACYL17]. Using hb-graphs to ensure navigability between facets enforce to consider paths in between the elements of the schema.

Using the incident matrices associated to the support hypergraph of the different visualisation hb-graphs and the incident matrix of the physical references related to the references chosen, the navigation between facet can be achieved only by transposition and matrix multiplication of these binary incident matrices. It opens up to lightweight applications where the only information given is the incident matrix of the physical entities related to the different facets of the information space.

Guarantee 6.4. *The framework should enhance the visual querying of the information space.*

Each facet is constituted of a visualisation hb-graph that is represented using an extra-node representation of the hb-graph. The extra-node refers to the reference instances that have this hb-edge in common. These reference instances are linked to the physical entities they are attached to. Also, the interaction between a facet and the facet of physical entities (that comprises only vertices of the physical entities) can be enhanced. On the opposite, as physical entities are linked to the references the co-occurrences are attached to, the co-occurrences can be highlighted in the different facets. Authorizing interactivity with extra-node and vertices can also enhance new search and eventually enable the building of complex queries. This aspect will be developed with a use case in Chapter D.1.

Guarantee 6.5. *The zooming in the dataset should be possible through restrictions to sub-hb-graphs.*

We have shown that we can restrict ourselves to a group of vertices before switching between the different facets.

Guarantee 6.6. *Heterogeneous co-occurrences should be possible.*

We handled this case in Section 6.3.3.2.

Guarantee 6.7. *Multiple types of references should be possible.*

It has been handled in Section 6.3.3.5 with two possible cases: the conjunctive, using cross product, and the exclusive disjunctive.

Finally, this theoretical hb-graph framework obeys to the Mantra of Visual analytics: "Analyze First – Show the Important – Zoom, Filter and Analyze Further – Details on Demand" as it is exposed in [KAF+08].

6.3.5.2. Comparison with existing solutions

	Verbatim browser	PivotPath	PivotSlice	CS core	DataEdron cube	DataHbEdron
output	linear	tripartite graph	graph	graph	linear & hypergraph	linear & hb-graph
#facets	1	3	many	many	4	many
view per facet	no	no	no	yes	yes	yes
simultaneous facet views	no	yes	yes	no	yes	yes
heterogeneous co-occurrences	x	no	no	yes	no	yes
multiple references	x	no	no	disjunctive	no	conjunctive, disjunctive
zoom in data	new query	no	yes	yes	no	yes
filter data	new query	no	yes	yes	no	by visual queries
visual query	no	no	yes, restricted to current search	yes, restricted to current search	no	yes, even with new search
redundancy in co-occurrences	x	no	no	no	no	yes
information extraction	limited	pivot change	elaborated questions	elaborated questions	elaborated questions	elaborated questions
combination of facets	no	no	yes	yes	yes	yes
type of ranking	binary cosine similarity	no	number of references per vertex	number of references per vertex	hyperedges and vertices	hb-edges and vertices

Table 6.2.: Elements of comparison between different multi-faceted visualization frameworks.

The validity of our framework is asserted by completeness and robustness of the mathematical construction: we have achieved the possibility to navigate inside the dataset by showing co-occurrences in a sufficiently refined way to support all the information extracted. As this model has been instantiated through a user interface in the use case of Arxiv, but, also, as mentioned previously, on some other sample data using csv files, its versatility is guaranteed. We have gathered in Table 6.2 some of the non-exhaustive features to compare our solution with the others. The user interface uses a 2.5D approach, but it is out of the scope of this Thesis to make any claim on the quality of the interactions a user can have with such an interface.

6.3.6. A use case

We applied this framework to perform searches and visual queries on the Arxiv database. The results are visualized in the DataHbEdron allowing simultaneous visualisation of the different facets of the information space. The tool developed is now part of the Collaboration Spotting family[6]. When performing a search, the standard Arxiv API[7]

[6] http://collspotting.web.cern.ch/
[7] https://arxiv.org/help/api/index

is used to query the Arxiv database. The queries can be formulated either by a text entry or done interactively directly using the visualisation: queries include single words or multiple words, with AND, OR and NOT possible and parenthesis grouping. The querying history is stored and presented as a hb-graph: it is also interactive allowing the visual building of complex queries and the refinement of the queries that were already performed. Each time a new query is formulated, the Arxiv API is used to retrieve the corresponding metadata.

When performing a search on Arxiv, the query is transformed into a vector of words. The most relevant documents are retrieved based on a similarity measure between the query vector and the word vectors associated to individual documents. Arxiv relies on Lucene's built-in Vector Space Model of information retrieval and the Boolean model[8]. The Arxiv API returns the metadata associated to the document with the highest scores for the query performed. We keep only the first n answers, with n tunable by the user. This metadata, filled by the authors during their submission of a preprint, contains different information such as authors, Arxiv categories and abstract.

The information space contains four main facets: the first facet shows the Arxiv reference visualisation hb-graph with a contextual sentence related to the query, and links to Arxiv article's presentation and to Arxiv article's pdf. This first facet layout is similar to classical textual search engines—Figure 6.7.

Figure 6.7.: First facet of the DataHbEdron: a well-known look like of a classical verbatim interface.

The second facet corresponds to co-authors of the articles using as reference the pub-

[8]https://lucene.apache.org/core/2_9_4/scoring.html

lication itself. The third facet is the co-keywords extracted from the abstracts. The fourth facet is showing the Arxiv categories involved in the reference.

Co-keywords are extracted from the abstracts using TextBlob, a natural language processing Python library[9]. We extract only nouns using the tagged text, where lemmatization and singularization have been done.

Nouns in the abstract of each document are scored with TF-IDF, the Term Frequency-Invert Document Frequency. Scoring each noun in each abstract of the retrieved documents generates a hb-graphs $\mathcal{H}_Q$ of universe the nouns contained in the abstracts. Each hb-edge contains a set of nouns extracted from a given abstract with a multiplicity function that represents the TF-IDF score of each noun. We keep only the first w words related to an abstract, where w is tunable by the end-user, in order to limit the size of the hb-edges.

The fifth facet shows the queries that have been performed during the session: the graph of those queries can be saved. The sixth facet is reserved to show additional information such as the pdf of publications.

Any node on the facets is interactive, allowing to highlight information from one facet to another one by showing the hb-edges that are mapped through the references. Queries can be built using the vertices of the hb-graph, either isolated or by combining them with the current search terms using AND, OR and NOT by keyboard shortcuts and mouse. The only query that has to be performed by typing is the first one. Merging queries from different users is immediate as it is only hb-edges in a hb-graph. The queries are potentially evolving, gathered, saved and re-sketched months later.

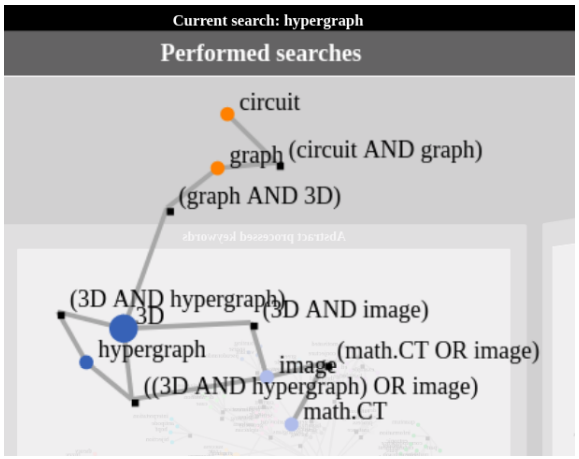


Figure 6.8.: Performed search.

We added the possibilities to display additional information related to authors using DBLP[10], but also to have information on keywords using both DuckDuckGo[11] for disambiguation and Wikipedia[12] to enhance Arxiv information.

[9] https://textblob.readthedocs.io/en/dev/
[10] https://dblp.org/
[11] https://duckduckgo.com/
[12] https://www.wikipedia.org/

6.3.7. Intermediate conclusion

The framework supports dataset visual queries, possibly contextual, that either result from searches on related subjects or refine the current search: it enables full navigability of the information space. It provides powerful insights into datasets using simultaneous facet visualisation of the information space directly constructed from the query results. This framework is versatile enough to enhance user insight into many other datasets, particularly textual and multimedia ones.

6.4. Further comments

The full interactivity of facets is achieved; potentially, facets have to support large visualisation hb-graphs composed of several hundred vertices and edges. The support of the visualisation of such hb-graphs relies mainly on the fine spreading of the information displayed over the space of visualisation.

Different strategies can be used including component detection and their circular placement. But in random uniform and general hypergraphs, as well as in most of the real co-occurrence networks a giant component occurs—[SPS85, GZCN09, dP15]—and this component requires special attention on the placement of the vertices for visualisation.

The core idea is to focus on the central vertices to spread the information; we focus our attention in the remaining parts of this Thesis on the detection of such vertices. The idea is to use diffusion on the hb-graph to achieve the ranking of vertices and hb-edges. Hb-graphs hold higher order relationships that cannot be captured by matrices, as they describe pairwise relations either of adjacency or of incidence. Modeling diffusion and capturing higher order relations require to use tensors. Hence our quest, from the next Chapter, to model adjacency in hb-graphs and to capture these higher order links.

7. Multi-diffusion in an information space

Highlights

This chapter is based on ideas of a talk presented at CTW 2019[1].

Prerequisites: Section 1.3 and, Chapters 2 and 6.

7.1. Motivation

In Chapter 6, we have shown how an information space can be modeled by using a hb-graph framework, composed of multiple facets showing visualisation hb-graphs and a reference hb-graph that displays the reference on the corresponding facet. In Chapter 2, we have obtained an efficient way of ranking vertices and hb-edges: it can be applied to the different visualisation hb-graphs separately. Nonetheless, we have shown that a mapping exists between the references used to generate the co-occurrence networks and the co-occurrences themselves. In this Chapter, we tackle the problem of aggregating the reference rankings obtained from the different facets. In the literature, the problem is known as rank aggregation or indifferently as ranking aggregation.

The research question we want to answer in this Chapter is the following:

Research question 7.1. *How to rank the references in an information space taking into account individual rankings obtained from the different facets?*

This question can be viewed as a refinement of the query results, which are traditionally based on the decreasing values of the binary cosine similarity between the query and the results. If the binary cosine similarity performs well in finding a list—not necessarily ordered—of results that matches the query and, if it is also fast to calculate, it is less informative for the results that have a non-zero similarity, and certainly not sufficient for performing an efficient ranking. Moreover, additional information related to the

[1] Published in the online proceedings of CTW 2019:
http://wwwhome.math.utwente.nl/~ctw/CTW2019ProceedingsFinal.pdf.

different facets can be retrieved. In this chapter, we aim at showing the influence of the different facet rankings to obtain an aggregated ranking. We propose a modified weighted approach of a classical method used for aggregation ranking and show the influence of weights on randomly generated information spaces and on real queries. The intuition behind this method is to let the possibility to tune the aggregated ranking by putting more emphasize on some facets in an information space.

We start by surveying the existing work and by giving the mathematical background of the approach prior to propose our new way of aggregating the rankings in an information space.

7.2. Mathematical background and related work

7.2.1. The problem of ranking aggregation

We consider a list $\mathscr{L}$ of N elements and m rankings $(\sigma_i)_{i\in[\![m]\!]}$ of the list $\mathscr{L}$. We write $\underset{\longleftrightarrow}{\mathscr{L}}$ the set of all possible rankings—including possible ties—of the list $\mathscr{L}$. The aim of ranking aggregation is to find a consensus order $\sigma^\star \in \underset{\longleftrightarrow}{\mathscr{L}}$ that aggregates the rankings $(\sigma_i)_{i\in[\![m]\!]}$ while reflecting them as much as possible. This can be achieved either by using directly ranks or by using a score function.

Achieving a consensus by using directly ranks implies to use a distance d and an aggregation function Aggr_r—often a sum, or a weighted sum—and the problem of consensus is formalized as finding $\sigma^\star$ such that:

$$\sigma^\star = \underset{\sigma\in\underset{\longleftrightarrow}{\mathscr{L}}}{\operatorname{argmin}} \mathrm{Aggr}_r\left(d\left(\sigma_i,\sigma\right)\right).$$

When using a function s that scores each ranking σ_i, the consensus is obtained by using a function Aggr_s, that is used to minimize, maximize, average,... the different scores obtained. Hence, the problem of consensus consists in finding $\sigma^\star$ such that it corresponds to a ranking obtained from the sorting of $\mathrm{Aggr}_s\left(s\left(\sigma_i\right)\right)$:

$$\sigma^\star = \mathrm{Rank}\left(\mathrm{Sort}\left(\mathrm{Aggr}_s\left(s\left(\sigma_i\right)\right)\right)\right).$$

7.2.2. Classical distances between rankings

To compare two rankings σ_1 and σ_2 of a list $\mathscr{L}$, different distances have been defined.

The **Spearman footrule distance** defined in [Spe87] between two rankings σ_1 and σ_2 of a list $\mathscr{L}$ corresponds to the displacement distance in absolute value between the rankings:

$$F\left(\sigma_1,\sigma_2\right) \overset{\Delta}{=} \sum_{l\in\mathscr{L}} \left|\sigma_1\left(l\right)-\sigma_2\left(l\right)\right|.$$

In [Ken38] and [Ken48], two distances between two rankings σ_1 and σ_2 of a list $\mathscr{L}$ are introduced. The first is the **Kendall tau ranking distance** $K\left(\sigma_1,\sigma_2\right)$ that corresponds to the number of inverted ranking between the two rankings. More formally:

$$K(\sigma_1,\sigma_2)\overset{\Delta}{=}\left|\left\{(i,j):i<j,(\sigma_1(i)<\sigma_1(j)\wedge\sigma_2(i)>\sigma_2(j))\vee(\sigma_1(i)>\sigma_1(j)\wedge\sigma_2(i)<\sigma_2(j))\right\}\right|$$

where $\sigma_k(i)$ is the ranking of i in the list σ_k where $k \in [\![2]\!]$.

Formulated an other way:

$$K(\sigma_1, \sigma_2) \triangleq \sum_{\{i,j\} \in [\![m]\!]} \overline{K}_{i,j}(\sigma_1, \sigma_2)$$

with:

$$\overline{K}_{i,j}(\sigma_1,\sigma_2) \triangleq \begin{cases} 1 & \text{if } (\sigma_1(i)<\sigma_1(j) \wedge \sigma_2(i)>\sigma_2(j)) \vee (\sigma_1(i)>\sigma_1(j) \wedge \sigma_2(i)<\sigma_2(j)) \\ 0 & \text{otherwise.} \end{cases}$$

At last, the **normalised Kendall tau distance** is defined as:

$$K_n(\sigma_1, \sigma_2) \triangleq \frac{K(\sigma_1, \sigma_2)}{\binom{n}{2}}.$$

7.2.3. Origin of the ranking aggregation problem

The ranking aggregation problem seeks to resolve the global ranking of elements for a given list of rankings of those elements. The ranking aggregation problem has its origin in election studies, before the French Revolution, and the research of a scrutiny that is fair to represent voters choice. In 1770, Borda presented to the French Royal Academy of Science a way of choosing the winner over the elector's preferences for candidates by using what is now known as the Borda method. The Borda method presented in [dB81] is a weighted vote system where the electors rank a list of the first k candidates they prefer, where k is common to all voters. Each candidate is given a number of points that corresponds to the number of candidates that are less preferred than himself. The candidate with the highest sum of points wins. Borda proposed this method to have an alternative to the method of Condorcet that Borda thought to be too complicated: in the Condorcet method presented in [No85], the candidate that is elected, called now the Condorcet winner, is the majority winner, if s·he·x exists, of a series of vote on a head-to-head basis between all the candidates. The fact that this method does not give automatically a winner is called the Condorcet paradox[2]. Nonetheless, the Borda's method was largely criticized by Condorcet as it forces to have large consensus, but fails to the majority criterion and is subject to scrutiny manipulations, since, if some of the voters are insincere, they can vote for an outsider they do not like in order to defeat a leading candidate and to promote a second choice candidate.

7.2.4. Kemeny-Young order

The **Kemeny-Young order** based on the Kendall tau distance was introduced in [Kem59]: it aims at finding a ranking that minimizes the number of disagreements between the global ranking and each of the rankings by evaluating the sum of the Kendall tau distance between a candidate ranking and the different rankings. Mathematically

[2] A typical example leading to the Condorcet paradox can be found on https://fr.wikipedia.org/wiki/Paradoxe_de_Condorcet.

speaking, considering m rankings $(\sigma_i)_{i\in[\![m]\!]}$ of a list of n objects, it searches the order $\sigma^\star$ such that it solves:

$$\sigma^\star = \underset{\sigma\in R}{\arg\min} K(\sigma, \sigma_1, ..., \sigma_m)$$

where:

$$K(\sigma, \sigma_1, ..., \sigma_m) = \sum_{k\in[\![m]\!]} K(\sigma, \sigma_k).$$

This method has the advantage to find a Condorcet winner if there exists. Moreover, in [YL78], the Kemeny's order is shown to be the unique preference function, for a given electorate, that gives a consensus, which is neutral—i.e. symmetric in the treatment of alternatives—, consistent—the preferred candidate does not depend on how the electorate is split—and Condorcet—i.e. rank alternatives in function of the majority.

7.2.5. A revival of interest

Ranking aggregation has found a revival of interest with the raise of Internet: [DKNS01] is an authoritative paper on the subject where the authors develop techniques to overcome the difficulties of finding a common ranking to a list of rankings as finding a Kemeny order is shown to be NP-hard even for $m = 4$ rankings.

To this end, they consider the extended Condorcet Criterion introduced in [To98] which states that "If the set of alternatives can be partitioned in such a way that all members of a subset of this partition defeat all alternatives belonging to subsets with a higher index [N/A: (ranking)], then the former should obtain a better rank than the latter." This allows to propose an admissible partition to recursively split the ties into a subset of the partition such that it can be solved separately.

Using any favorite aggregation method, in [DKNS01], the authors propose a local optimization, called local Kemenization, such that it helps in improving the overall $K(\sigma, \sigma_1, ..., \sigma_m)$ by just successively transposing adjacent pair of elements if it decreases the value of K.

In [SZ09], the authors compare different methods of rank aggregation and distinguish three categories of ranking methods which include positional methods, comparison sort methods and hybrid methods. Positional methods aim at finding an optimal permutation of the elements—ranking—in which the position of each element is "close to its average position"; whereas comparison methods use directly the comparison between the ranking of the elements and suppose to have a binary relation on the elements that compares their ranking. The binary relation ranks an element i before an element j if a majority of the list of rankings has ordered them in this order. This relation is not transitive and, hence, is not even a pre-order relation, as only reflexivity is ensured. Hybrid methods include methods that are both positional and comparison methods: they include the Markov chains introduced by [DKNS01] for web pages, especially the Markov chain named MC4. MC4 is based on a random walk where the surfer decides to move from a page P to a page Q on the majority rule based on the ranking of the two pages. The algorithm is given in Algorithm 7.1. The algorithm is shown to outperform other aggregation methods, both without or with local Kemenization.

In [Lin10], three ranking method categories are proposed: distributional based that is not described in [SZ09], heuristic, that includes both Borda's method and Markov Chain approaches, and stochastic optimization methods, aiming at minimizing a criteria.

Algorithm 7.1 MC4 algorithm as proposed in [DKNS01].

- Current state page P.
- Choose Q uniformly among all the pages ranked.
- If $\sigma_i(Q) < \sigma_i(P)$ for a majority of the rankings $(\sigma_i)_{1 \leqslant i \leqslant m}$:
  go to Q

else:
  stay on P.

In [TN11], the authors tackle the problem of multi-modality fusion for image and video re-ranking by considering different similarity graphs built from different features of a dataset, called modalities. They consider two main iterative approaches to find a modality agreement that use different strategies: one approach uses semi-supervised learning by considering the different rankings, and the other one uses a random walk by creating a context graph which is a linear fusion of the different modality graphs.

In [DL18], the authors propose a multiplex PageRank based on topologically biased random walks. Multiplex networks are series of graphs based on different features of an information space. Biases are introduced to give more or less weight to the topological properties of nodes such as degree, strength or clustering in the considered random walks. They consider different facets of an information space and work on some collaboration and citation networks. The biases are introduced through exponents of features.

In [WZS16], the authors use a unified hypergraph of different features and labels of social images to build heterogeneous hyperedges with an incident matrix similar to the one of a hb-graph, using a softmax similarity. This heterogeneous hypergraph is used to learn labels on vertices, using different learn weights on the hyperedges.

In [ACYL17], the authors use a multi-hypergraph—multi in the sense of multiple—fusion framework for person re-identification through non overlapping cameras: they build different hypergraphs based on feature descriptor using a star representation—i.e. an extra-node representation—where the extra-node represents in this case the centroid of the feature. Each vertex belongs to different hyperedges depending on the strength of connection—based on a heat kernel—that is calculated using a distance between the vertex and the extra-node viewed as the centroid of the hyperedge. The incident matrix constructed is similar to the one of a hb-graph. To fuse the hypergraphs, they formalize an objective function that combines an overall regularization term based on the hypergraph Laplacian applied to a relevance vector, a graph weight regularization—using the Laplacian of a similarity matrix between two hypergraphs filtered by a heat kernel with weights on hypergraphs—, and an empirical loss function depending on weights. They alternatively update the relevance vector and the weights on hypergraphs until convergence. Their results overtake other methods on two popular benchmark datasets for person re-identification.

In [GLG+13], the authors propose a visual analysis tool to achieve multi-attribute rankings: they allow the sub-selection of attributes and normalize the attribute value to have a common scale. The aggregation score is obtained by either doing the sum of the pondered normalised attributes or by taking the maximum of the score obtained on the different sub-lists. The final ranking is based on those scores: the higher the score, the higher the rank. The whole set of scores is given using a multi-column presentation,

with columns used as rank separators that allow to have not only the aspect of slope graph but help to read the final ranking.

As it is emphasized in [LAYK+18], on the Internet, many websites use ranking policies for search results, advertisement, recommender systems. When a new ranking is proposed, the evaluation can be done either online—i.e. directly in production—using A/B testing between the proposed ranking and the previous, or offline. The offline evaluation requires to use a click model as it is proposed in [CSdR13]. Other studies use additional data to evaluate the ranking policy. For instance, in [LAYK+18], the authors use click logs of previous searches to construct estimators to evaluate the number of clicks expected from the new ranking; it allows them to avoid ranking with poor results.

7.2.6. Comparison of rankings

We consider two rankings σ_1 and σ_2 of a list $\mathscr{L}$ of n elements indexed by $i \in [\![n]\!]$.

The **strict Kendall tau rank correlation coefficient**, written τ, corresponds to the ratio of the difference in the number of strict concordant pairs and strict discordant pairs related to the number of pairs. Its explicit expression is:

$$\tau(\sigma_1, \sigma_2) \triangleq \frac{\sum\limits_{i,j \in [\![n]\!] : i<j} \operatorname{sgn}(\sigma_1(i) - \sigma_1(j)) \operatorname{sgn}(\sigma_2(i) - \sigma_2(j))}{\binom{n}{2}}.$$

This coefficient does not take into account the pairs that are matching, which can be important in case of ties.

For this, we can define the **matching correlation coefficient** ν defined as:

$$\nu_{\text{match}}(\sigma_1, \sigma_2) \triangleq \frac{\sum\limits_{i,j \in [\![n]\!] : i<j} \mathbf{1}_{(\sigma_1(i)==\sigma_1(j))} \mathbf{1}_{(\sigma_2(i)==\sigma_2(j))}}{\binom{n}{2}},$$

where $\mathbf{1}_{(\phi_b)} = \begin{cases} 1 & \text{if } \phi_b; \\ 0 & \text{otherwise,} \end{cases}$ with ϕ_b a Boolean condition statement.

We introduce a **large Kendall tau rank correlation coefficient** τ_L that takes into account the agreement and disagreement on ties.

We define the **disagreement on matching correlation coefficient**:

$$\nu_{\text{disagree}}(\sigma_1,\sigma_2) \triangleq \frac{\sum\limits_{i,j \in [\![n]\!] : i<j} \left(\mathbf{1}_{(\sigma_1(i)!=\sigma_1(j))} \mathbf{1}_{(\sigma_2(i)==\sigma_2(j))} + \mathbf{1}_{(\sigma_1(i)==\sigma_1(j))} \mathbf{1}_{(\sigma_2(i)!=\sigma_2(j))} \right)}{\binom{n}{2}}$$

and:

$$\tau_L(\sigma_1, \sigma_2) \triangleq \tau(\sigma_1, \sigma_2) + \nu_{\text{match}}(\sigma_1, \sigma_2) - \nu_{\text{disagree}}(\sigma_1, \sigma_2).$$

The **scaled Spearman footrule correlation coefficient** ρ is defined as:

$$\rho\left(\sigma_1,\sigma_2\right) \triangleq \frac{\sum\limits_{i\in[\![n]\!]} \left|\sigma_1\left(i\right)-\sigma_2\left(i\right)\right|}{\left(\begin{array}{c} n \\ 2 \end{array}\right)}.$$

Finally, we will also use the **Jaccard index @**k, $k \in \mathbb{N}\backslash\{0\}$. To achieve this purpose, we extract sub-lists related to the two rankings corresponding to the top k elements $\ell_i = \left\{l : l \in [\![n]\!] \wedge \sigma_i\left(l\right) \leqslant k\right\}$, $i = 1, 2$. The Jaccard index @k of the rankings σ_1 and σ_2 is defined as:

$$J@k\left(\sigma_1,\sigma_2\right) \triangleq \frac{\#\left(\ell_1 \cap \ell_2\right)}{\#\left(\ell_1 \cup \ell_2\right)}.$$

and the **overlap coefficient @**k, $k \in \mathbb{N}\backslash\{0\}$, of the rankings σ_1 and σ_2 is defined as :

$$O@k\left(\sigma_1,\sigma_2\right) \triangleq \frac{\#\left(\ell_1 \cap \ell_2\right)}{\min\left(\#\ell_1,\#\ell_2\right)}.$$

7.3. Multi-diffusing in an information space

After having simplified the notations used in Chapter 6, we start studying the different strategies that could be taken to aggregate the rankings of references. We then expose the two strategies retained.

7.3.1. Laying the first stones

We consider an information space with $K+1$ facets $\left(\mathcal{F}_k\right)_{k\in[\![K+1]\!]}$ containing respectively data instances of different types $\left(T_k\right)_{k\in[\![K+1]\!]}$, one of them being chosen as the type of reference. We can always consider that the data instances of the reference type are attached to the $K+1$-th facet, even if we need to re-index the facet and type sequences. We write $R = V_{K+1}$ the set of references.

The data instances of a given type T_k, with $k \in [\![K+1]\!]$, constitute a set V_k attached to the k-th facet. We write $\overline{\mathfrak{H}_k} = \left(V_k, \overline{\mathfrak{E}_k}, w_{e,k}\right)$ the reduced visualisation hb-graph of the facet $\mathcal{F}_k$.

We consider:

$$\rho_k : R \to \overline{\mathfrak{E}_k}$$

such that:

$$\forall r \in R, \exists \overline{\mathfrak{e}} \in \overline{\mathfrak{E}_k} : \ \rho_k(r) = \overline{\mathfrak{e}}.$$

ρ_k is surjective.

The attached references to a given hb-edge can be found by defining:

$$\nu_k : \overline{\mathfrak{E}_k} \to \mathcal{P}\left(R\right),$$

such that:

$$\forall \overline{\mathfrak{e}} \in \overline{\mathfrak{E}_k} : \nu_k\left(\overline{\mathfrak{e}}\right) = \left\{r \in R : \rho_k\left(r\right) = \overline{\mathfrak{e}}\right\}.$$

7.3.2. Strategies for ranking references

To achieve the ranking of the references that are at the origin of the different facet visualisation hb-graphs, we can think of different strategies.

A first strategy is to rank the hb-edges of each facet separately by using, for instance the diffusion by exchange on each visualisation hb-graph for the different facets. The ranking aggregation takes place at the level of the references, by considering the different rankings of hb-edges the references are related to. This strategy permits to use some classical methods of ranking aggregation such as the Borda's or Condorcet's methods. Moreover, to enhance personalized rankings, we can put weights on the different facets: the rank aggregation can then be achieved by using a modified Borda method for instance or MC4 in [DKNS01], but might exclude some methods such as the Condorcet one.

Other strategies consider various hb-graphs involving both the references and the hb-edges attached to them, either at the hb-edge level or at the vertex level.

The second strategy would be to consider a hb-graph of hb-edges attached to a reference and to enhance diffusion by exchange on it. However, in real world cases, this hb-graph is potentially insufficiently connected, as hb-edges of the different facets might not repeat so often.

The third strategy consists in adding connectivity in the hb-graph proposed in the second strategy, by reporting the lattice of intersection of the hb-edges of each facet into $\mathfrak{H}$, enriching the hb-edges of $\mathfrak{H}$ with additional hb-edges that reflect the hb-edge connections in each facets. One possible pitfall of this method is that the facet hb-edges are loosing in a way the higher order information of the intersection as two facet hb-edges will be connected if they intersect in the facet.

Other strategies consist of more sophisticated diffusion based on modified hb-graphs: they are out of scope of this chapter, but can be kept in mind as further possible refinements. Considering all these arguments, we only consider the first strategy in this Chapter.

7.3.3. Ranking aggregation by modified weighted MC4

In this approach, we consider that for each facet the exchange-based diffusion on the reduced visualisation hb-graph has been achieved, leading to a sorting of the hb-edges. As each hb-edge is by construction linked to one or more references, the ranks are reported to every attached references; we obtain a ranking σ_k of all the references $r \in R$ for each facet $\mathcal{F}_k$, $k \in [\![K]\!]$, with possible ties.

In order to tune the rankings, we give tunable weights to the facets $(w_{\mathcal{F}_k})_{k\in[\![K]\!]}$ such that: $\sum_{k\in[\![K]\!]} w_{\mathcal{F}_k} = 1$ and $w_{\mathcal{F}_k} \geqslant 0$ for all $k \in [\![K]\!]$.

We start by computing a weighted majority matrix M of elements obtained for each couple of references $(r_{i_1}, r_{i_2}) \in R^2$ by:

$$M(r_{i_1}, r_{i_2}) = \sum_{k\in[\![K]\!]} w_k \mathbf{1}_{\sigma_k(r_{i_1})<\sigma_k(r_{i_2})} - \sum_{k\in[\![K]\!]} w_k \mathbf{1}_{\sigma_k(r_{i_1})>\sigma_k(r_{i_2})}.$$

Algorithm 7.2 The WT-MC4 algorithm.

Prerequisite:

A set of references R and K rankings $(\sigma_k)_{k\in[\![K]\!]}$ of these references

Algorithm:

Calculate for all $(r_{i_1}, r_{i_2}) \in R^2$ the elements of M :

$$M(r_{i_1}, r_{i_2}) = \sum_{k\in[\![K]\!]} w_k \mathbf{1}_{\sigma_k(r_{i_1})<\sigma_k(r_{i_2})} - \sum_{k\in[\![K]\!]} w_k \mathbf{1}_{\sigma_k(r_{i_1})>\sigma_k(r_{i_2})}.$$

Stack = R.
L = [0 for $r \in R$] # To store the number of times the reference is visited.
Choose a current state reference: $r_{\text{current}} \in R$.
Remove r_{current} from Stack.
While stack is not empty:
    Choose a random number γ
    Choose r_{next} uniformly among all the references in R.
    If $\gamma > \gamma_0$:
        $r_{\text{current}} := r_{\text{next}}$
    else:
        If $M(r_{\text{next}}, r_{\text{current}}) > 0$
            $r_{\text{current}} := r_{\text{next}}$
        else:
            # stay in r_{current}.
    $L[r_{\text{current}}] += 1$.
    If $r_{\text{current}} \in$ Stack:
        Remove r_{current} from Stack.
LoopWhile

We propose a modified version—that we call **WT-MC4**—of the Algorithm 7.1 shown as the most efficient in [DKNS01]. This modified version includes teleportation and weights and is presented in Algorithm 7.2. The teleportation is useful for the management of ties, so that the algorithm do not loop infinitely. The threshold γ_0 is put at 0.85 which is a classical value used in the PageRank algorithm, even if we did not investigate further the relevance and efficiency of this threshold.

7.3.4. Results and evaluation

We validate the approach using random generated information spaces and apply it to the case of Arxiv querying.

7.3.4.1. Randomly generated information space

We generate each time an information space of N_{facet} facets with visualisation hb-graphs generated using the same method as in Chapter 2. The number of hb-edges for each of these facets varies and it is set to a maximum value $P_{\max}$: it corresponds also to the maximal number of references. Each reference $r \in [\![P_{\max}]\!]$ is associated randomly to one hb-edge of each facet: in this way, some hb-edges are repeated.

We give the possibility to assign a weight to each facet and explore systematically different cases that are generated using weights shared uniformly between k facets out of N_{facet}, $k \in [\![N_{\text{facet}}]\!]$, the remaining weights being put to zero.

For computation time reasons, as we generate N_{IS} information spaces—with $N_{\text{IS}} \leqslant 1000$—, we limit the experiment to $N_{\text{facet}} \leqslant 6$. Taking more facets would require more time for exploring the different weighting cases.

For each random information space generated, we compute for each of their facets the ranking of the hb-edges obtained by exchange-based diffusion. We then compute the aggregation of ranking by the Borda method and by our modified MC4, using the different weights assigned to the hb-edges, and compute the different indexes proposed in Section 7.2.6.

At the end of the N_{IS} iterations, we calculate the average of the different ranking comparison features proposed in Section 7.2.6 between the current ranking and the one obtained with the Borda method and their standard deviations. The results are presented in Table 7.1 where $N_{\text{facet}} = 3$ and $N_{\text{IS}} = 1000$. We observe in this Table that the best agreement with the Borda method is achieved with our WT-MC4 method, with equal weighting of the facets. Putting the weights on only two facets out of the three increases the disagreement—approximately 10% drop in Kendall tau correlation between equi-weight set on the three facets and two facets non-zero equi-weighting out of the three facets—with the Borda method realized on the three facets, but the gap is much less important than when putting the weight on only one facet—approximately 53% drop in Kendall tau correlation between equi-weight and one facet non-zero weighting out of the three facets. We will investigate later in this Chapter this phenomena, depending on the number of facets in the information space and the number of non-zero weighted facets. The results are similar for all the other indices with optimal values—achieved for the highest values for all indices but the scaled Spearman footrule correlation coefficient which has to be the closest to zero—obtained for the comparison of the Borda method and the modified equi-weight MC4.

In Table 7.2, obtained from the same set of information spaces than in Table 7.1, we observe that with similar facets the Kendall tau coefficient between the ranking aggregation obtained by WT-MC4 with equal weights and the ranking obtained for each facet with the exchange-based diffusion is equally low whatever the facet is, compared to the cases with weight only on one facet, where the agreement is the highest. We also observe that our method of ranking is fair, as when the facets are similarly generated, the results obtained on the different facets are similar. We did not investigate further the case of dissimilar facets in the information space.

In Table 7.3 and Figure 7.1, for $N_{\text{IS}} = 100$, we present the results of the evolution of Kendall tau coefficient between the ranking aggregation obtained by WT-MC4 with equal weights and the ranking obtained for each facet with the exchange-based diffusion when the number of facets increases from 1 to N_{facet}, with N_{facet} varying from 2 to 5. Increasing the number of facets implies more compromises between the different facets, and hence to have a lower value of the Kendall tau for the equi-weight WT-MC4 when N_{facet} increases. This behavior depends only on the number of non-zero weight facets, and not on the number of facets of the information space generated; it underlines the difficulties to arrive to a consensus.

Current	$\overline{f(\sigma_{\text{Borda}}, \sigma_{\text{Current}})}$ / $\sigma(f(\sigma_{\text{Borda}}, \sigma_{\text{Current}}))$ with $f = \ldots$										
	τ	τ_L	ν_{disagree}	ν_{matching}	ρ	O@10	O@25	O@50	J@10	J@25	J@50
Facet 0 $\overline{\lvert V\rvert} \approx 455 \quad \sigma(\lvert V\rvert) \approx 241$ $\overline{\lvert E\rvert} \approx 173 \quad \sigma(\lvert E\rvert) \approx 85$	0.389 / 0.122	0.257 / 0.093	0.135 / 0.068	0.004 / 0.002	0.541 / 0.179	0.426 / 0.195	0.597 / 0.176	0.736 / 0.160	0.144 / 0.087	0.246 / 0.108	0.359 / 0.129
Facet 1 $\overline{\lvert V\rvert} \approx 452 \quad \sigma(\lvert V\rvert) \approx 246$ $\overline{\lvert E\rvert} \approx 172 \quad \sigma(\lvert E\rvert) \approx 85$	0.385 / 0.127	0.255 / 0.098	0.134 / 0.068	0.004 / 0.002	0.544 / 0.178	0.430 / 0.198	0.597 / 0.178	0.738 / 0.156	0.145 / 0.095	0.244 / 0.109	0.360 / 0.131
Facet 2 $\overline{\lvert V\rvert} \approx 462 \quad \sigma(\lvert V\rvert) \approx 244$ $\overline{\lvert E\rvert} \approx 174 \quad \sigma(\lvert E\rvert) \approx 85$	0.389 / 0.122	0.257 / 0.094	0.136 / 0.068	0.004 / 0.003	0.539 / 0.176	0.427 / 0.197	0.593 / 0.175	0.734 / 0.159	0.145 / 0.094	0.244 / 0.107	0.361 / 0.128
References	0.428 / 0.169	0.357 / 0.260	0.073 / 0.106	0.002 / 0.002	0.436 / 0.171	0.458 / 0.194	0.541 / 0.130	0.613 / 0.088	0.295 / 0.153	0.345 / 0.119	0.364 / 0.117
WT-MC4 {'0': 0.33, '1': 0.33, '2': 0.33}	**0.650** / 0.108	**0.629** / 0.110	0.022 / 0.002	0.000 / 0	**0.253** / 0.078	**0.632** / 0.167	**0.699** / 0.124	**0.749** / 0.108	**0.452** / 0.168	**0.532** / 0.143	**0.596** / 0.137
WT-MC4 {'0': 0.5, '1': 0.5, '2': 0.0}	0.579 / 0.123	0.558 / 0.124	0.021 / 0.003	0.000 / 0	0.306 / 0.088	0.341 / 0.171	0.483 / 0.147	0.487 / 0.152	0.204 / 0.120	0.319 / 0.125	0.438 / 0.142
WT-MC4 {'0': 0.5, '1': 0.0, '2': 0.5}	0.584 / 0.123	0.564 / 0.124	0.021 / 0.003	0.000 / 0	0.302 / 0.088	0.349 / 0.168	0.482 / 0.149	0.606 / 0.138	0.208 / 0.119	0.319 / 0.127	0.439 / 0.139
WT-MC4 {'0': 0.0, '1': 0.5, '2': 0.5}	0.582 / 0.124	0.561 / 0.124	0.021 / 0.003	0.000 / 0	0.304 / 0.088	0.345 / 0.168	0.482 / 0.147	0.607 / 0.138	0.205 / 0.119	0.318 / 0.125	0.440 / 0.139
WT-MC4 {'0': 0.0, '1': 0.0, '2': 1.0}	0.301 / 0.094	0.275 / 0.095	0.027 / 0.005	0.000 / 0	0.485 / 0.060	0.212 / 0.159	0.353 / 0.156	0.489 / 0.154	0.121 / 0.102	0.219 / 0.117	0.331 / 0.138
WT-MC4 {'0': 0.0, '1': 1.0, '2': 0.0}	0.300 / 0.096	0.274 / 0.096	0.026 / 0.005	0.000 / 0	0.486 / 0.061	0.215 / 0.158	0.351 / 0.157	0.489 / 0.155	0.123 / 0.101	0.217 / 0.119	0.331 / 0.139
WT-MC4 {'0': 1.0, '1': 0.0, '2': 0.0}	0.303 / 0.096	0.277 / 0.096	0.027 / 0.005	0.000 / 0	0.484 / 0.061	0.216 / 0.157	0.353 / 0.155	0.487 / 0.152	0.123 / 0.100	0.219 / 0.117	0.329 / 0.135

Table 7.1.: Different ranking comparison features—as defined in Section 7.2.6 between Borda ranking and current ranking on 1000 generated information spaces of 3 facets with similar parameters.

Facet diffusion ranking $\Longrightarrow$	Facet 0		Facet 1		Facet 2	
vs WT-MC4 with weights $\Downarrow$	Mean	StDev	Mean	StDev	Mean	StDev
{'0': **0.33**, '1': **0.33**, '2': **0.33**}	**0.390**	0.135	**0.391**	0.133	**0.385**	0.136
{'0': **0.5**, '1': **0.5**, '2': **0.0**}	**0.505**	0.029	**0.503**	0.030	**0.033**	0.045
{'0': 0.5, '1': 0.0, '2': 0.5}	0.502	0.028	0.035	0.046	0.503	0.029
{'0': 0.0, '1': 0.5, '2': 0.5}	0.036	0.047	0.502	0.028	0.503	0.029
{'0': 0.0, '1': 0.0, '2': 1.0}	0.020	0.047	0.021	0.047	0.649	0.264
{'0': 0.0, '1': 1.0, '2': 0.0}	0.024	0.049	0.654	0.264	0.022	0.047
{'0': **1.0**, '1': **0.0**, '2': **0.0**}	**0.653**	0.262	**0.025**	0.051	**0.021**	0.049

Table 7.2.: Kendall tau of 1000 WT-MC4 ranking aggregation compared to the facet rankings (average on the 1000 information) depending on the weights put on the facets. The average features of the facets are given in Table 7.1.

# non-zero weight facets $\Rightarrow$	1		2		3		4		5	
# facets $\Downarrow$	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
2	0.585	0.251	**0.499**	0.030						
3	0.613	0.268	0.501	0.030	**0.391**	0.127				
4	0.665	0.252	0.504	0.029	0.393	0.126	**0.355**	0.100		
5	0.727	0.259	0.506	0.027	0.396	0.125	0.357	0.041	**0.321**	0.080

Table 7.3.: Average Kendall tau of the WT-MC4 ranking aggregation compared to the non-zero weight facet rankings (average on 100 information spaces) depending on the number of facets having non-zero equal weights. Parameters of facet hb-graphs similar to Table 7.1.

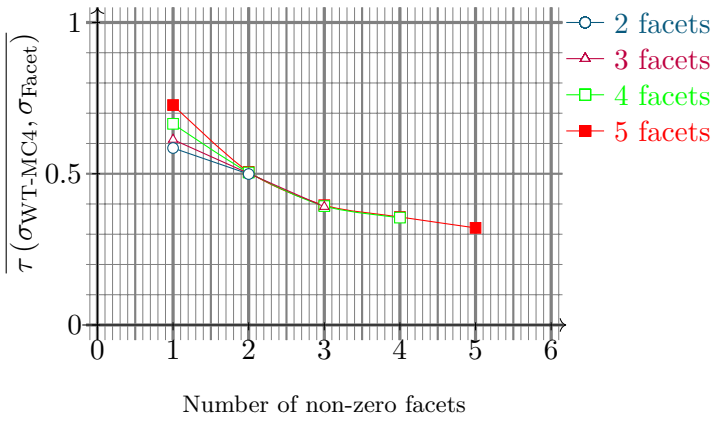


Figure 7.1.: Average Kendall tau correlation coefficient between the WT-MC4 ranking aggregation compared to the non-zero weight facet rankings depending on the number of non-zero weight facet rankings—corresponding data is shown in Table 7.2—(100 information spaces for each value of the number of facets).

# non-zero weight facets ⇒	1		2		3		4		5		6	
# facets ⇓	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
2	0.355	0.090	0.824	0.053								
3	0.295	0.096	0.582	0.109	0.665	0.104						
4	0.276	0.088	0.488	0.114	0.543	0.110	0.777	0.060				
5	0.248	0.087	0.420	0.092	0.491	0.098	0.662	0.085	0.710	0.081		
6	0.240	0.073	0.382	0.083	0.443	0.099	0.577	0.103	0.637	0.092	0.763	0.065

Table 7.4.: Average Kendall tau of the WT-MC4 ranking aggregation—with non-zero weights as indicated—compared to the Borda aggregation ranking—on all facet rankings—(average on 100 information spaces) depending on the number of facets having non-zero weights. Parameters of facet hb-graphs similar to Table 7.1.

In Table 7.4 and in Figure 7.2, we present the evolution of the strict Kendall tau rank correlation coefficient between the Borda aggregation ranking and WT-MC4, depending on the number of non-zero facets. It confirms that the variation in the disagreement between the two rankings decreases when the number of facets increases, with a behavior that is different between an odd and an even number of facets, and similarly odd and even number k of non-zero weight facets. The transition from even k to odd k leads to a lesser increase than the transition from odd k to even k. Increasing the number of facets for ranking does not necessarily improves the agreement between the two rankings. Finally, these results show how the WT-MC4 approximates the ranking obtained by Borda method with a stochastic weighted approach when the weights are equi-distributed between the non-zero weight facets.

Wondering if the results where due to a low number of generated information spaces, we have generated $N_{\text{IS}} = 1000$ information spaces. We observed exactly the same kind of results than those shown in Table 7.5 and Figure 7.3.

All these results are conform to expectation for such a weighting of facets, and, essentially, indicate that putting weights differently on facets allows the favoring of different facets, with an expected facets re-ranking that is in line with the weight—the emphasize—that is put on the different facets. Moreover, increasing the number of facets

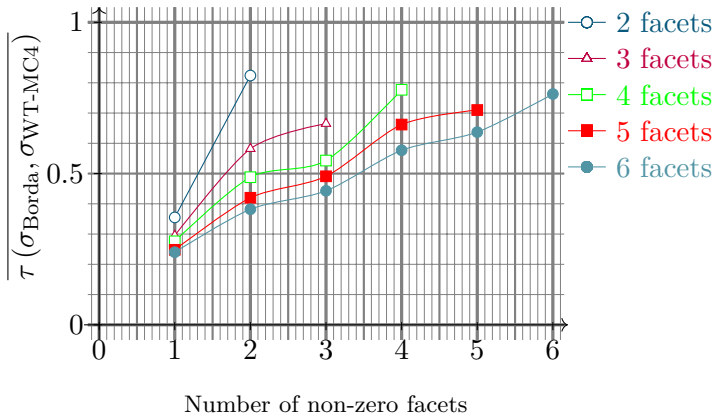


Figure 7.2.: Average Kendall tau correlation coefficient between the ranking obtained by Borda aggregation of the facet diffusion by exchange rankings and the WT-MC4 realized with equi-weight on non-zero equal weight facets—corresponding data is shown in Table 7.4—(100 information spaces for each value of the number of facets).

# non-zero weight facets ⇒	1		2		3		4		5		6	
# facets ⇓	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
2	0.357	0.096	0.821	0.050								
3	0.303	0.094	0.585	0.127	0.650	0.112						
4	0.272	0.093	0.485	0.116	0.545	0.116	0.771	0.066				
5	0.250	0.091	0.424	0.108	0.480	0.109	0.650	0.094	0.690	0.085		
6	0.239	0.086	0.393	0.096	0.455	0.101	0.592	0.089	0.628	0.090	0.755	0.058

Table 7.5.: Average Kendall tau of the WT-MC4 ranking aggregation compared to the non-zero equal weight facet rankings (average on 1000 information spaces) depending on the number of facets having non-zero weights. Parameters of facet hb-graphs similar to Table 7.1.

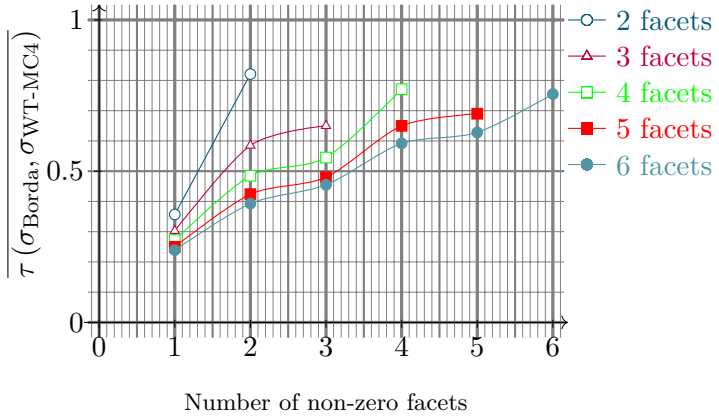


Figure 7.3.: Average Kendall tau between Borda aggregation of the rankings obtained on references via diffusion by exchange and the WT-MC4 with equi-weight on non-zero equal weight facets—it corresponds to data shown in Table 7.5—(1000 random information spaces for each value of the number of facets are generated).

makes the agreement slower to reach, but at the end, the level of agreement with the Borda ranking depends essentially on the parity of the number of facets.

7.3.4.2. Application to the Arxiv information space

The main difficulty in applying multi-diffusion to a real case is that the facet visualization hb-graphs are essentially unconnected, which limits terribly the exchange-based diffusion ranking to ensure the convergence and the coherence of the ranking. It means, strictly speaking, that the exchange-based diffusion will give results that are coherent only for the connected components. Refining the way the connected components are ranked between one another is out of the scope of this section, and is an open research question.

However, we applied the multi-diffusion to three facets of Arxiv information space: authors, keywords and subject categories. In each facet, the hb-edge ranking is achieved by using the exchange-based diffusion. In Table 7.6, we summarize the top 1000 answers returned from the 27 Arxiv queries performed: most of the time, the only facet visualization hb-graph that is connected is the one with keywords—or in the worst cases has only a few components. As in the previous section, there is no correlation between the rankings of the different facets. Since we are in fact interested in the ranking aggregation, and the way it works, we use these imperfect rankings on the overall facet visualization hb-graphs as a basis to apply the ranking aggregation.

We also consider the hb-graph of references based on their connection through hb-edges and make some diffusion on it, even if again this hb-graph is not well connected. With the rankings obtained in individual facets, we perform an aggregation ranking using the Borda method and WT-MC4 with different weights, as it has been done in the random case. We compare the different rankings obtained by WT-MC4 with the different weights to the rankings of the facets, the one of the Borda method and the one of the references, using the strict Kendall's tau coefficient.

The different ranking aggregation comparisons are given in Table 7.7: the results are based on a limited number of 29 queries performed on Arxiv including the first 100 results and 200 results; the queries performed are: 3D, printer, hypergraph, graph, "Higgs boson", particle, snow, star, Laplacian, diffusion, nanotechnology, CNN, RNN, neural network, "machine learning", "deep learning", learning, machine, "information retrieval", "information", "data mining", entropy, information, adjacency, "3D printer", physics, particle, network, neural. The results of the Borda aggregation ranking is mainly due to the author and keyword facets as it is both shown in the difference of Kendall's tau coefficient with equi-weight WT-MC4 and a:0.5/k:0.5/t:0 WT-MC4. Keyword ranking or author ranking alone still perform well facing the Borda aggregation ranking, but there is a true complementarity between the two, as the a:0.5/k:0.5/t:0 WT-MC4 Kendall's tau coefficient with Borda aggregation ranking method is improved significantly (+0.23 for authors alone and +0.16 for keywords alone). The values obtained for equi-weight WT-MC4 are higher than the one we could expect with three random facets: this can be due to the nature of hb-graphs that are behind.

Detailed results for the strict Kendall tau correlation coefficients of the different searches are given in Table 7.8 and illustrated with a three 2-D diagrams in Figure 7.4 for the facets of Authors and Keywords, Figure 7.5 for the facets of Authors and Tags and, in

		3D	printer	hypergraph	graph	boson Higgs	particle	snow	star	Laplacian	diffusion	nanotechnology	CNN	RNN
authors - keywords		-0.001	-0.063	0.102	0.086	0.083	0.129	-0.049	0.064	0.115	0.13	-0.013	0.048	0.103
authors - tags		0.055	0.051	-0.027	0.038	0.092	0.009	0.023	0.008	-0.027	-0.001	0.057	-0.044	-0.054
keywords - tags		-0.106	0.025	-0.077	-0.003	0.003	0.004	-0.051	-0.086	-0.009	-0.041	0.014	-0.038	-0.072
authors	$\|V\|$	2948	825	1529	1868	1329	2146	13468	2366	1592	2124	3854	3228	2936
	$\|E\|$	953	219	883	945	817	943	685	934	895	935	969	969	960
	# C.C.	627	193	403	656	337	787	283	489	598	779	698	655	597
keywords	$\|V\|$	4884	2616	3248	3275	2051	3972	4700	4166	2796	3962	4707	4720	4476
	$\|E\|$	1000	230	1017	997	999	999	720	999	998	1000	1000	1000	1000
	# C.C.	2	1	1	1	1	2	2	3	1	1	1	1	1
tags	$\|V\|$	161	105	343	382	17	159	121	64	461	252	91	96	101
	$\|E\|$	271	121	423	491	25	312	205	90	575	396	221	217	238
	# C.C.	10	13	5	7	1	7	11	15	7	8	6	3	3

		neural network	machine learning	deep learning	learning	machine	information retrieval	information	data mining	entropy	adjacency	3D printer	physics	network	neural
authors - keywords		0.062	0.097	0.082	0.051	0.07	0.041	0.105	0.052	0.092	0.109	-0.026	0.047	0.045	0.067
authors - tags		0.009	-0.021	-0.006	0.008	-0.007	-0.016	-0.023	-0.046	-0.041	0.03	0.082	-0.04	0.084	0.006
keywords - tags		-0.014	-0.051	-0.038	-0.028	-0.054	0.005	-0.053	-0.006	-0.07	0.012	-0.069	-0.075	0.045	-0.017
authors	$\|V\|$	2486	2594	2892	2491	2392	2357	1830	2542	1589	2053	2970	2143	2307	2533
	$\|E\|$	963	979	961	967	961	914	900	945	895	926	953	1530	951	967
	# C.C.	795	761	728	668	783	629	692	717	696	737	679	1373	670	796
keywords	$\|V\|$	3759	3561	3968	3490	3977	3935	3815	4600	3317	3958	5389	3570	3907	3910
	$\|E\|$	999	999	999	999	997	998	992	996	999	1000	1000	1832	999	1000
	# C.C.	2	2	1	1	2	2	3	1	1	1	2	8	1	1
tags	$\|V\|$	169	154	119	114	216	162	197	187	247	406	183	214	164	173
	$\|E\|$	367	288	281	220	398	301	380	368	379	537	288	553	352	369
	# C.C.	9	4	6	2	8	5	12	9	5	20	10	2	5	8

Table 7.6.: Strict Kendall tau correlation coefficient between two facets exchange-based diffusion rankings and some additional statistics on facets when querying Arxiv, on first top 1000 answers returned by the Arxiv API.

	Top 100 results					Top 200 results					Top 1000 results				
	$\|V\|$	$\|E\|$	#C.C.	τ_{100}	$\sigma(\tau_{100})$	$\|V\|$	$\|E\|$	#C.C.	τ_{200}	$\sigma(\tau_{200})$	$\|V\|$	$\|E\|$	#C.C.	τ_{1000}	$\sigma(\tau_{1000})$
authors (a)	241.9	97.8	89.1	0.437	0.095	484.7	118.0	167.1	0.440	0.088	2718.2	922.9	660.2	0.470	0.055
keywords (k)	864.5	100.9	1.4	0.537	0.064	1384.1	199.9	1.4	0.541	0.057	3878.9	991.4	1.7	0.549	0.049
tags (t)	47.3	51.2	7.4	0.179	0.095	73.1	90.0	8.2	0.166	0.086	187.3	321.0	7.4	0.095	0.077
references	249.9	100.9	x	0.096	0.162	481.6	199.9	x	0.130	0.159	2235.3	992.9	x	0.220	0.149
WT-MC4 a: 0.33, k: 0.33, t: 0.33	x	x	x	0.734	0.069	x	x	x	0.706	0.065	x	x	x	0.660	0.071
WT-MC4 a: 0.5, k: 0.5, t: 0	x	x	x	0.687	0.060	x	x	x	0.699	0.050	x	x	x	0.741	0.041
WT-MC4 a: 0.5, k: 0, t: 0.5	x	x	x	0.430	0.037	x	x	x	0.421	0.053	x	x	x	0.392	0.056
WT-MC4 a: 0, k: 0.5, t: 0.5	x	x	x	0.545	0.063	x	x	x	0.526	0.050	x	x	x	0.475	0.043
WT-MC4 a: 0, k: 0, t: 1	x	x	x	0.179	0.095	x	x	x	0.160	0.086	x	x	x	0.088	0.074
WT-MC4 a: 0, k: 1, t: 0	x	x	x	0.533	0.064	x	x	x	0.538	0.056	x	x	x	0.546	0.048
WT-MC4 a: 1, k: 0, t: 0	x	x	x	0.425	0.099	x	x	x	0.426	0.078	x	x	x	0.443	0.064

Table 7.7.: Strict Kendall tau correlation coefficient between Borda aggregation ranking and the rankings obtained either directly by diffusion on the corresponding facet or by WT-MC4 with different weights, when querying the Arxiv database (on 29 queries) for the first 100, 200 and 1000 results returned by the API.

	3D	printer	hypergraph	graph	boson Higgs	particle	snow	star	Laplacian	diffusion	nanotechnology	CNN	RNN
authors (a)	0.472	0.378	0.504	0.49	0.476	0.482	0.445	0.498	0.456	0.498	0.473	0.502	0.535
keywords (k)	0.505	0.476	0.553	0.534	0.606	0.609	0.485	0.563	0.544	0.566	0.503	0.534	0.553
tags (t)	0.052	0.292	0.066	0.181	0.055	0.107	0.082	-0.036	0.213	0.11	0.109	-0.002	-0.025
references	0.292	-0.028	0.104	0.039	0.548	0.265	0.171	0.572	-0.022	0.142	0.292	0.405	0.389
WT-MC4 a: 0.33, k: 0.33, t: 0.33	0.577	0.738	0.695	0.748	0.56	0.665	0.619	0.559	0.806	0.728	0.582	0.551	0.648
WT-MC4 a: 0.5, k: 0.5, t: 0	0.776	0.695	0.745	0.73	0.781	0.767	0.763	0.805	0.656	0.707	0.78	0.777	0.793
WT-MC4 a: 0.5, k: 0, t: 0.5	0.421	0.436	0.406	0.431	0.448	0.403	0.35	0.372	0.443	0.373	0.405	0.426	0.356
WT-MC4 a: 0, k: 0.5, t: 0.5	0.479	0.546	0.493	0.491	0.595	0.49	0.442	0.467	0.501	0.45	0.445	0.477	0.378
WT-MC4 a: 0, k: 0, t: 1	0.044	0.293	0.037	0.125	0.044	0.107	0.076	-0.033	0.199	0.108	0.108	-0.006	-0.031
WT-MC4 a: 0, k: 1, t: 0	0.502	0.474	0.551	0.532	0.605	0.604	0.483	0.561	0.54	0.561	0.499	0.532	0.551
WT-MC4 a: 1, k: 0, t: 0	0.467	0.382	0.485	0.414	0.386	0.464	0.441	0.477	0.412	0.467	0.461	0.478	0.522

	neural network	machine learning	deep learning	learning	machine	information retrieval	information	data mining	entropy	adjacency	3D printer	physics	network	neural
authors (a)	0.479	0.511	0.513	0.503	0.472	0.472	0.473	0.48	0.45	0.467	0.453	0.236	0.491	0.488
keywords (k)	0.542	0.565	0.548	0.535	0.543	0.542	0.572	0.532	0.579	0.547	0.495	0.736	0.519	0.538
tags (t)	0.121	0.023	0.048	0.034	0.099	0.089	0.09	0.088	0.066	0.233	0.11	0.037	0.204	0.112
references	0.178	0.311	0.269	0.344	0.195	0.206	0.178	0.174	0.179	0.03	0.232	0.198	0.075	0.202
WT-MC4 a: 0.33, k: 0.33, t: 0.33	0.709	0.6	0.667	0.591	0.671	0.65	0.662	0.72	0.679	0.783	0.594	0.603	0.706	0.709
WT-MC4 a: 0.5, k: 0.5, t: 0	0.736	0.761	0.764	0.773	0.723	0.752	0.736	0.737	0.707	0.686	0.767	0.636	0.734	0.728
WT-MC4 a: 0.5, k: 0, t: 0.5	0.386	0.427	0.38	0.398	0.388	0.387	0.391	0.374	0.347	0.437	0.411	0.157	0.446	0.373
WT-MC4 a: 0, k: 0.5, t: 0.5	0.454	0.483	0.433	0.448	0.464	0.461	0.506	0.413	0.463	0.526	0.492	0.512	0.49	0.436
WT-MC4 a: 0, k: 0, t: 1	0.119	0.016	0.05	0.033	0.098	0.083	0.086	0.087	0.066	0.214	0.108	0.041	0.201	0.107
WT-MC4 a: 0, k: 1, t: 0	0.539	0.562	0.546	0.533	0.54	0.54	0.568	0.53	0.576	0.544	0.492	0.73	0.516	0.537
WT-MC4 a: 1, k: 0, t: 0	0.454	0.486	0.5	0.483	0.446	0.439	0.403	0.457	0.392	0.462	0.445	0.178	0.482	0.477

Table 7.8.: Kendall tau correlation coefficient between Borda aggregation ranking and the rankings obtained either directly by diffusion on the corresponding facet or by WT-MC4 with different weights, when querying the Arxiv database (27 queries).

Figure 7.6 for the facets of Keywords and Tags; it enhances the fluctuations depending on the queries on the rankings depending on the different facets. The facets of authors and keywords often contribute a lot in the final ranking obtained in WT-MC4 and corroborate with Borda aggregation method. We observe that the tags have more importance in queries covering different subjects, i.e. when the categorization by tags starts to be discriminative of different topics.

7.4. Further comments

Rank aggregation in an information space requires to address different difficulties, including ties and the connectedness of the generated hb-graphs. We have proposed in this study a way of aggregating the different facets by using a random surfing with teleportation in the information space, using a weighted criteria to handle the behavior of the random surfer. This process is finally a stochastic barycentric Borda. Applying it to real cases requires to find some ways of handling the different connected components in any real case applications, that might be very different in nature depending on the facets. A way of handling this would certainly be to normalize the values of the diffusion according to the size of the components the vertices are in. This remains an open question to implement it and evaluate it. Another approach could be to consider a modified exchange-based diffusion process that would take into account the links to the

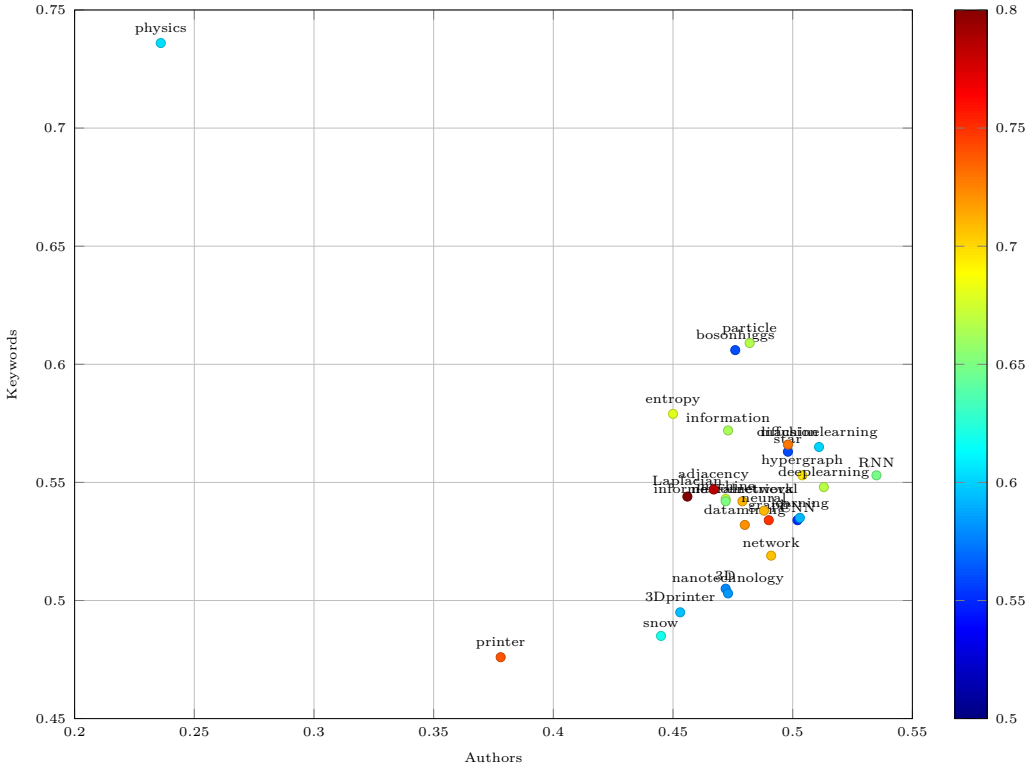


Figure 7.4.: Comparison of different queries Kendall tau correlation coefficients between Borda aggregation ranking and the rankings obtained either directly by diffusion on the corresponding facet (Authors and Keywords) or by WT-MC4 with equi-weights, when querying the Arxiv database (27 queries).

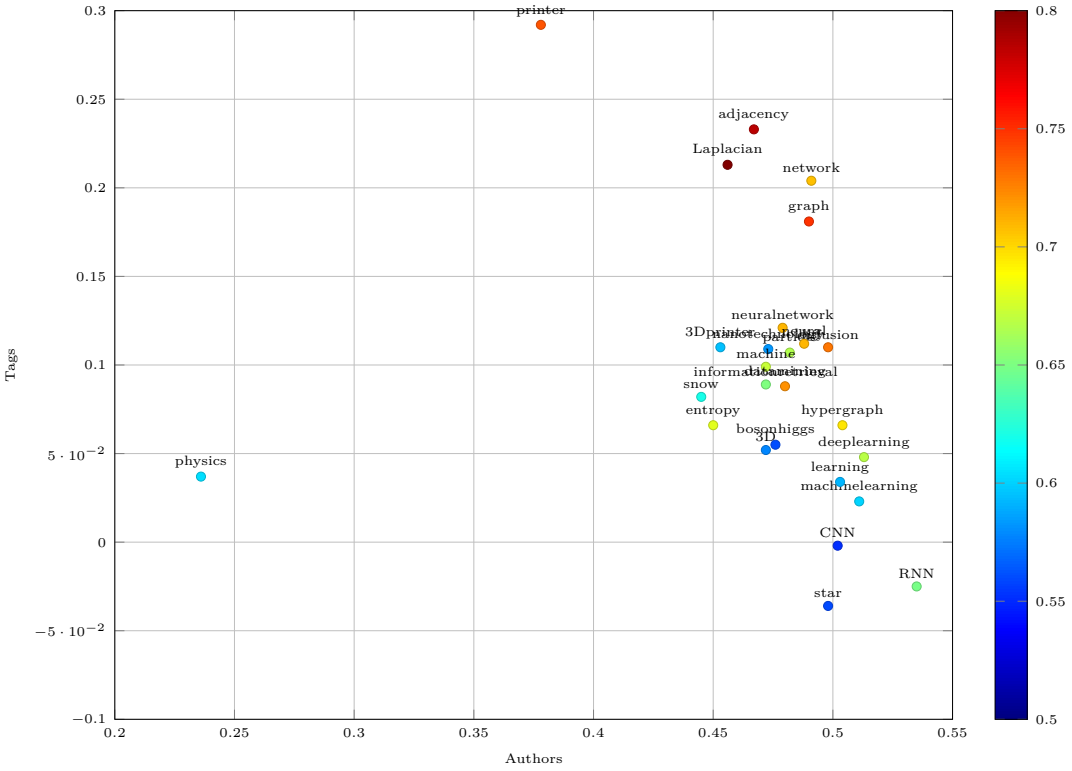


Figure 7.5.: Comparison of different queries Kendall tau correlation coefficients between Borda aggregation ranking and the rankings obtained either directly by diffusion on the corresponding facet or by WT-MC4 with equi-weights, when querying the Arxiv database (27 queries).

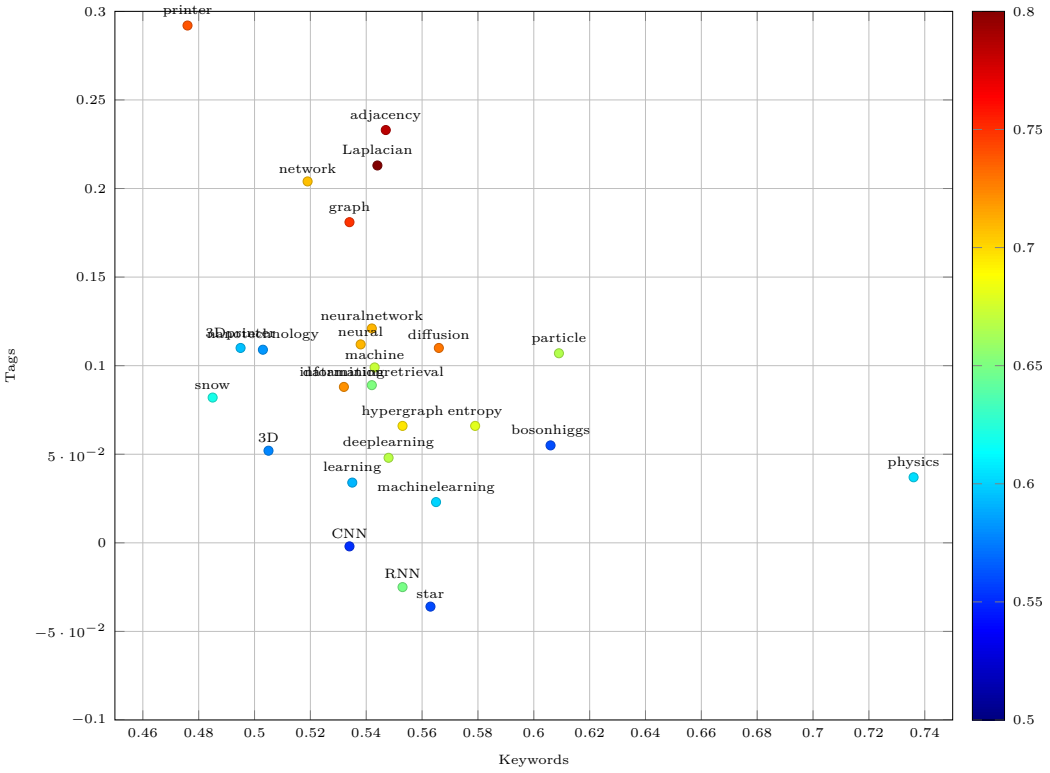


Figure 7.6.: Comparison of different queries Kendall tau correlation coefficients between Borda aggregation ranking and the rankings obtained either directly by diffusion on the corresponding facet or by WT-MC4 with equi-weights, when querying the Arxiv database (27 queries).

references. A third way could be to introduce a low weighted hyperedge that connects all the vertices in one hyperedge, reflecting that the elements belong to the same search. The method we propose as any algorithm of this kind can be parallelized.

Conclusion

From a coarsening problem to visualize large hypergraphs in order to handle properly their layout, the quest to model the diffusion on such structures has led us to redefining adjacency in hypergraphs. Modeling e-adjacency requires tensors. We have focused on preserving the interpretability in terms of hypergraph uniformisation and polynomial homogenization, which has led us to introduce hb-graphs, and particularly natural hb-graphs, to handle vertex redundancy.

Hb-graphs have proven to be effective to handle efficiently refined information in comparison with hypergraphs and graphs. They have been an opportunity to introduce an exchange-based diffusion that takes into account multiplicities of vertices and that enables the ranking not only of vertices, but also of hb-edges.

Using natural hb-graphs, we proposed two additional e-adjacency tensors, and particularly the straightforward one, which has some nice spectral properties. Nonetheless, using exchange-based diffusion, we have shown that the m-uniformisation processes involved have an influence on the diffusion over the structure itself. In this context, we have evaluated the different proposed tensors presented in this Thesis as well as in the literature.

Concluding by introducing a new Laplacian tensor for general hb-graphs, we have opened the door to new theoretical developments to study tensor-based diffusion.

Concomitantly, hb-graphs fit for the modeling of co-occurrence networks. Their use in multi-faceted information spaces leads us to propose a hb-graph framework, and the DataHbEdron, enabling full visual querying of a textual dataset.

Supporting redundancy, natural hb-graphs enable a dense space, that fills the space of hypergraphs. They constitute a bridge between graph structures, multiset structures and their set support, tensor approaches and polynomial views. A lot of their properties have still to be discovered or studied, such as the L-polynomial.

This research has lead us to explore different worlds, often precluded of difficult challenges, but has been a charming journey. We hope to continue this exploration in forth-coming research: there are a lot of remaining open research questions that can appear as a research statement for future work. We gather here some of the research questions raised in this Thesis that require further investigations.

- ***On hb-graphs themselves:*** hb-graphs are a powerful structure to handle refined information on co-occurrences. We still have to investigate some of their mathematical properties. This includes the following questions:
  - *Are general hb-graphs a separate category?*
    $\rightsquigarrow$ We conjecture that this question has a positive answer, and that natural hb-graphs are a sub-category.

  - *What are extensions of hb-graphs to negative multiplicities?*
    $\rightsquigarrow$ It could be very valuable to extend hb-graphs to negative multiplicities for handling cases related to opinion mining.
  - *How to handle multi-variable multiplicities on hb-graphs?*
    $\rightsquigarrow$ This could be a possibility to handle multiple features on a same hb-graphs, for instance multiplicity and weights, or in image processing.
  - *What is a coloring of a hb-graph?*
    $\rightsquigarrow$ Hypergraph coloring has applications in divide and conquer approaches; hb-graph coloring, and particularly of natural hb-graphs could lead to refinements in this strategies.
  - *What kind of clustering algorithm to use with hb-graphs?*
    $\rightsquigarrow$ Some work exists on structures that are not properly using hyperedge-based weighting of vertices and that are in practice similar to hb-graphs. These works have shown that taking into accounts multiplicities at the hb-edge level enhances clustering. Investigating further clustering on hb-graphs could lead to refined clustering algorithms that account for the higher order relationships.

- ***On multi-faceted information spaces and diffusion in facets:*** different challenges at every level exists: it includes an upstream work on the datasets themselves, on the way the data are fed into the information space, on the evaluation of the visualization interface, as well on the extraction of the important information from the networks themselves.
  - *What is the impact on the user to interact with multi-faceted visualisation of an information space to solve tasks?*
    $\rightsquigarrow$ A full evaluation including A/B testing is required to evaluate the Data-HbEdron compared to traditional verbatim interfaces for searching in an information space.
  - *How to improve the data feeding of the information space?*
    $\rightsquigarrow$ The proposed framework is versatile enough to support various kind of data. The implementation of facets is done manually at the moment. Automatic facet discovery could be one way of investigation to improve the feeding of the information space. This subject encompasses also the problem of data linkage and the merging of heterogeneous sources, and automatic exploration throughout the different datasets. All these problems are linked to the FAIR-ability of data. FAIR—for Findable, Accessible, Interoperable and Reusable— are principles that were edict in 2016 in [WDA+16] for improving infrastructures that supports the reuse of scholarly data.
  - *How to rank information in a disconnected hb-graph? How to rank an information space that is disconnected on some facets?*
    $\rightsquigarrow$ Different strategies can be foreseen, one being to connect data with a differentiated link that would interconnect them by the fact they belong to the same information space or are members of the same facet.
  - *How to enhance diffusion in other ways in the information space?*
    $\rightsquigarrow$ Alternative strategies to the one taken for ranking aggregation can be

foreseen based on diffusion. Some of them interconnect the different facets via links between them.

  - *Which sort of features to use as abstract information function?*
    $\rightsquigarrow$ The influence of biases still have to be tested on the rank aggregation and on real datasets.

- ***On e-adjacency tensor and tensor-based diffusion:*** this aspect concerns additional theoretical and experimental developments on the proposed e-adjacency tensor and layered Laplacian.
  - *What is the disturbance on diffusion of the m-uniformisation process in simulated and real cases?*
    $\rightsquigarrow$ Evaluating the real impact of m-uniformisation processes is still an open work; it includes its testing on different real datasets, and finding measures to evaluate it.
  - *How do we define the algebraic / analytic connectivity in hb-graphs? What are the links between the hb-graph layered Laplacian eigenvalues and the hb-graph features such as the algebraic / analytic connectivity?*
    $\rightsquigarrow$ Algebraic connectivity in graphs corresponds to the second eigenvalue of the Laplacian matrix; extension to hypergraphs have been proposed for even uniform hypergraphs as the second smallest Z-eigenvalue of the Laplacian in [HQ12]. To extend algebraic connectivity, [Qi14] uses the concept of analytic connectivity for a uniform hypergraph via an optimization formulation. Extending this concept to hb-graphs may link to additional insights on the particular structures, like it has been done for graphs with hypergraphs.
  - *How does the diffusion process based on this layered Laplacian evolve with time? How can we use this layered Laplacian for clustering?*
    $\rightsquigarrow$ Finding how the layered Laplacian helps to understand the underlying diffusion process is fundamental for having insights into the capture of higher order relationships. Coupled with the idea of the spectral clustering in graphs, the layered Laplacian could help clustering the hb-graph via spectral techniques.
  - *Can we evaluate the perturbation on the exchange-based diffusion from the L-polynomial before uniformisation?*
    $\rightsquigarrow$ As the layered Laplacian is constructed using a polynomial homogenization, this homogenization corresponds to a m-uniformisation of the hb-graph and might have impacts on the diffusion process on the structure.
  - *What is a spectral analysis of hb-graphs?*
    $\rightsquigarrow$ This is a vast topic, that could help to refine results in the particular structures that are hypergraphs for hb-graphs, as hypergraphs already help to refine graph spectral analysis. We already investigate some results in our Thesis on the built e-adjacency tensor for general hb-graphs.

- ***Further questions*** to be addressed that open up the subject, to different fields from visualisation, via ontologies and problems of data linkage and FAIR principles:
  - *How to enhance meaningful visualizations for heterogeneous datasets, mixing textual and numerical values, while keeping the links?*

  $\rightsquigarrow$ This is a key issue, as categorical data are often displayed as bar charts while textual data fit for hb-graph representation. Using our mechanisms in the hb-graph framework enhances new possibilities of visualisation and interactivity, that still have to be investigated.

- *How to give semantic to clusters in textual datasets?*
  $\rightsquigarrow$ Finding the semantic of vertices gathered in communities is fundamental in order to give insights on the visualization of very large hb-graphs, as they need to be presented using communities due to their complexity.
- *In relation with the DataHbEdron, several sources of data can be used: how to ensure the coherence of what is going to be displayed?*
  $\rightsquigarrow$ Behind, there is still the problem of interoperability and FAIR-ability of the data (even if it is not necessary scholar data); data cleansing and data linking might be necessary to obtain sufficiently meaningful and coherent visualizations.

It remains an open question on how diffusion over such structures should be handled, if ever possible. Diffusion is a clear concept on graphs. Generalizing this concept to (m-)uniform structures with the finite element approach in order to obtain the Laplacian remains valid thanks to the hb-edge / hyperedge uniformity. But it is still unclear if this concept can be generalized to non-uniform structures like general hypergraphs, and a posteriori to structures like general hb-graphs. A revision of the diffusion concept to multi-diffusion—multi in the sense of multi-directed—is seen as a requirement for further developments.

Meyrin (Switzerland), March 2020, the 23rd.

Bibliography

[ABG06] Marianne Akian, Ravindra Bapat, and Stéphane Gaubert. Max-plus algebra. *Handbook of linear algebra*, 39, 2006, https://doi.org/10.1201/9781420010572.

[ACYL17] Le An, Xiaojing Chen, Songfan Yang, and Xuelong Li. Person re-identification by multi-hypergraph fusion. *IEEE transactions on neural networks and learning systems*, 28(11): 2763–2774, 2017, https://doi.org/10.1109/TNNLS.2016.2602082.

[ADLGP17] A. Agocs, D. Dardanis, J.-M. Le Goff, and D. Proios. Interactive graph query language for multidimensional data in Collaboration Spotting visual analytics framework. *ArXiv e-prints*, December 2017, https://arxiv.org/abs/1712.04202.

[AG08] Renzo Angles and Claudio Gutierrez. Survey of Graph Database Models. *ACM Comput. Surv.*, 40(1): 1:1–1:39, February 2008, http://doi.acm.org/10.1145/1322432.1322433.

[Alb91] Joseph Albert. Algebraic Properties of Bag Data Types. In *VLDB*, volume 91, pages 211–219. Citeseer, 1991, http://dl.acm.org/citation.cfm?id=645917.672310.

[BA16] Marina Blanton and Everaldo Aguiar. Private and oblivious set and multiset operations. *International Journal of Information Security*, 15(5): 493–518, October 2016, https://doi.org/10.1007/s10207-015-0301-1.

[BAD13] Abdelghani Bellaachia and Mohammed Al-Dhelaan. Random walks in hypergraph. In *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods, Venice Italy*, pages 187–194, 2013.

[BBCH03] Airat Bekmetjev, Graham Brightwell, Andrzej Czygrinow, and Glenn Hurlbert. Thresholds for families of multisets, with an application to graph pebbling. *Discrete Mathematics*, 269(1-3): 21–34, July 2003, https://doi.org/10.1016/S0012-365X(02)00745-8.

[BCM17] Anirban Banerjee, Arnab Char, and Bibhash Mondal. Spectra of general hypergraphs. *Linear Algebra and its Applications*, 518: 14–30, April 2017, https://doi.org/10.1016/j.laa.2016.12.022.

[BDE09] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review*, 51(1): 34–81, February 2009, https://doi.org/10.1137/060657704.

[Ber67] Claude Berge. Graphes et hypergraphes. 1970. *Dunod, Paris*, 1967.

[Ber73] Claude Berge. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

[BFR06] J.-P. Banâtre, P. Fradet, and Y. Radenac. Generalised multisets for chemical programming. *Mathematical Structures in Computer Science*, 16(04): 557, August 2006, https://doi.org/10.1017/S0960129506005317.

[BFR07] Jean-Pierre Banâtre, Pascal Fradet, and Yann Radenac. Programming self-organizing systems with the higher-order chemical language. *IJUC*, 3(3): 161–177, 2007.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10): P10008, October 2008, https://doi.org/10.1088/1742-5468/2008/10/P10008.

[Bli88] Wayne D. Blizard. Multiset theory. *Notre Dame Journal of Formal Logic*, 30(1): 36–66, December 1988, https://doi.org/10.1305/ndjfl/1093634995.

[BLM93] Jean-Pierre Banâtre and Daniel Le Métayer. Programming by multiset transformation. *Communications of the ACM*, 36(1): 98–111, January 1993, https://doi.org/10.1145/151233.151242.

[BMSW08] Ronald Brown, Ifor Morris, J Shrimpton, and Christopher D Wensley. Graphs of morphisms of graphs. *the electronic journal of combinatorics*, 15(1): 1, 2008.

[BN03] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6): 1373–1396, June 2003, https://doi.org/10.1162/089976603321780317.

[Bre13] Alain Bretto. *Hypergraph Theory*. Mathematical Engineering. Springer International Publishing, Heidelberg, 2013.

[BS17] Mojtaba Nouri Bygi and Ileana Streinu. Efficient pebble game algorithms engineered for protein rigidity applications. In *2017 IEEE 7th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pages 1–6, Orlando, FL, October 2017. IEEE, https://doi.org/10.1109/ICCABS.2017.8114301.

[CAO+18] Nicholas Cummins, Shahin Amiriparian, Sandra Ottl, Maurice Gerczuk, Maximilian Schmitt, and Bjorn Schuller. Multimodal Bag-of-Words for Cross Domains Sentiment Analysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4954–4958, Calgary, AB, April 2018. IEEE, https://doi.org/10.1109/ICASSP.2018.8462660.

[CD12] Joshua Cooper and Aaron Dutle. Spectra of uniform hypergraphs. *Linear Algebra and its Applications*, 436(9): 3268–3292, May 2012, https://doi.org/10.1016/j.laa.2011.11.018.

[CDF+04] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[CF88] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 539–549, White Plains, NY, USA, 1988. IEEE, https://doi.org/10.1109/SFCS.1988.21970.

[CGLM08] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric Tensors and Symmetric Tensor Rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3): 1254–1279, January 2008, https://doi.org/10.1137/060661569.

[Chu93] F Chung. The Laplacian of a hypergraph. *Expanding graphs (DIMACS series)*, pages 21–36, 1993.

[CK97] S.Jeromy Carrière and Rick Kazman. WebQuery: searching and visualizing the Web through connectivity. *Computer Networks and ISDN Systems*, 29(8-13): 1257–1267, September 1997, https://doi.org/10.1016/S0169-7552(97)00062-7.

[CLTZ18] T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral Properties of Hypergraph Laplacian and Approximation Algorithms. *Journal of the ACM*, 65(3): 1–48, March 2018, https://doi.org/10.1145/3178123.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6): 377–387, June 1970, https://doi.org/10.1145/362384.362685.

[Cor04] Gilberto Corso. Families and clustering in a natural numbers network. *Physical Review E*, 69(3): 036106, March 2004, https://doi.org/10.1103/PhysRevE.69.036106.

[Cou65] Philippe Courrege. Sur la forme intégro-différentielle des opérateurs de C^∞_k dans C satisfaisant au principe du maximum. *Séminaire Brelot-Choquet-Deny. Théorie du Potentiel*, 10(1): 1–38, 1965.

[CQU15] Pierre Comon, Yang Qi, and Konstantin Usevich. A Polynomial Formulation for Joint Decomposition of Symmetric Tensors of Different Orders. In *Latent Variable Analysis and Signal Separation*, volume 9237, pages 22–30. Springer International Publishing, Cham, 2015.

[CR19] Uthsav Chitra and Benjamin J Raphael. Random Walks on Hypergraphs with Edge-Dependent Vertex Weights. *arXiv preprint arXiv:1905.08287*, 2019.

[CSdR13] Aleksandr Chuklin, Pavel Serdyukov, and Maarten de Rijke. Click model-based information retrieval metrics. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, page 493, Dublin, Ireland, 2013. ACM Press, https://doi.org/10.1145/2484028.2484071.

[CTTV15] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Forward and backward bisimulations for chemical reaction networks. *arXiv preprint arXiv:1507.00163*, 2015.

[CTWZ19] T.-H. Hubert Chan, Zhihao Gavin Tang, Xiaowei Wu, and Chenzi Zhang. Diffusion operator and spectral analysis for directed hypergraph Laplacian. *Theoretical Computer Science*, 784: 46–64, September 2019, https://doi.org/10.1016/j.tcs.2019.03.032.

[CW02] L. Chen and P. P. Wang. Fuzzy relation equations (I): the general and specialized solving algorithms. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 6(6): 428–435, September 2002, https://doi.org/10.1007/s00500-001-0157-3.

[dB81] Jean C de Borda. Mémoire sur les élections au scrutin. 1781.

[DB13] Michael Donoser and Horst Bischof. Diffusion Processes for Retrieval Revisited. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1320–1327, Portland, OR, USA, June 2013. IEEE, https://doi.org/10.1109/CVPR.2013.174.

[DB14] Aurélien Ducournau and Alain Bretto. Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Computer Vision and Image Understanding*, 120: 91–102, March 2014, https://doi.org/10.1016/j.cviu.2013.10.012.

[DKNS01] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 613–622, Hong Kong, Hong Kong, 2001. ACM Press, https://doi.org/10.1145/371920.372165.

[DL18] Cangfeng Ding and Kan Li. Centrality Ranking via Topologically Biased Random Walks in Multiplex Networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Rio de Janeiro, July 2018. IEEE, https://doi.org/10.1109/IJCNN.2018.8489403.

[DM11] Matthias Dehmer and Abbe Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1): 57–78, January 2011, https://linkinghub.elsevier.com/retrieve/pii/S0020025510004147.

[DP99] Jiurgen DAËËOW and Gheorghe PaAUN. On the power of membrane computing. *Journal of Universal Computer Science*, 5(2): 33–49, 1999.

[dP15] Élie de Panafieu. Phase transition of random non-uniform hypergraphs. *Journal of Discrete Algorithms*, 31: 26–39, March 2015, https://doi.org/10.1016/j.jda.2015.01.009.

[DPN08] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Tampa, FL, USA, December 2008. IEEE, http://ieeexplore.ieee.org/document/4761366/.

[DPP16] Megan Dewar, David Pike, and John Proos. Connectivity in Hypergraphs. *arXiv preprint arXiv:1611.07087*, 2016.

[DRRD12] M. Dork, Nathalie Henry Riche, G. Ramos, and S. Dumais. PivotPaths: Strolling through Faceted Information Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(12): 2709–2718, December 2012, https://doi.org/10.1109/TVCG.2012.252.

[DW80] W. Dörfler and D. A. Waller. A category-theoretical approach to hypergraphs. *Archiv der Mathematik*, 34(1): 185–192, December 1980, https://doi.org/10.1007/BF01224952.

[DWL19] Cunxiang Duan, Ligong Wang, and Xihe Li. Some Properties of the Signless Laplacian and Normalized Laplacian Tensors of General Hypergraphs. *Taiwanese Journal of Mathematics*, July 2019, https://doi.org/10.11650/tjm/190606.

[EG95] Thomas Eiter and Georg Gottlob. Identifying the Minimal Transversals of a Hypergraph and Related Problems. *SIAM Journal on Computing*, 24(6): 1278–1304, December 1995, https://doi.org/10.1137/S0097539793250299.

[ERV05] Ernesto Estrada and Juan A Rodriguez-Velazquez. Complex networks as hypergraphs. *arXiv preprint physics/0505137*, 2005.

[Fag83] Ronald Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM*, 30(3): 514–550, July 1983, https://doi.org/10.1145/2402.322390.

[FCS12] K M Frahm, A D Chepelianskii, and D L Shepelyansky. PageRank of integers. *Journal of Physics A: Mathematical and Theoretical*, 45(40): 405101, October 2012, https://doi.org/10.1088/1751-8113/45/40/405101.

[FMU82] Ronald Fagin, Alberto O. Mendelzon, and Jeffrey D. Ullman. A simplied universal relation assumption and its properties. *ACM Transactions on Database Systems*, 7(3): 343–360, September 1982, http://portal.acm.org/citation.cfm?doid=319732.319735.

[Fur86] G. W. Furnas. *Generalized fisheye views*, volume 17. April 1986.

[FW93] Mark I. Freidlin and Alexander D. Wentzell. Diffusion Processes on Graphs and the Averaging Principle. *The Annals of Probability*, 21(4): 2215–2245, October 1993, https://doi.org/10.1214/aop/1176989018.

[FYZ+19] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33: 3558–3565, July 2019, https://doi.org/10.1609/aaai.v33i01.33013558.

[GD17] Debarghya Ghoshdastidar and Ambedkar Dukkipati. Uniform hypergraph partitioning: Provable tensor methods and sampling techniques. *Journal of Machine Learning Research*, 18(50): 1–41, 2017.

[Geo17] Konstantinos Georgatos. Multi-agent Belief Revision Using Multisets. In *Logic, Rationality, and Interaction*, volume 10455, pages 466–479. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[GJ12] KP Girish and John Sunil Jacob. On multiset topologies. *Theory and Applications of Mathematics & Computer Science*, 2(1): 37–52, 2012.

[GLG+13] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE*

Transactions on Visualization and Computer Graphics, 19(12): 2277–2286, December 2013, https://doi.org/10.1109/TVCG.2013.173.

[GMM+13] Ziyu Guan, Gengxin Miao, Russell McLoughlin, Xifeng Yan, and Deng Cai. Co-Occurrence-Based Diffusion for Expert Search on the Web. *IEEE Transactions on Knowledge and Data Engineering*, 25(5): 1001–1014, May 2013, https://doi.org/10.1109/TKDE.2012.49.

[Gun03] Jeremy Gunawardena. Chemical reaction network theory for in-silico biologists. *Notes available for download at http://vcp. med. harvard. edu/papers/crnt. pdf*, 2003.

[GXTL10] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1): 113–129, February 2010, https://doi.org/10.1007/s10044-008-0141-y.

[GZCN09] Gourab Ghoshal, Vinko Zlatić, Guido Caldarelli, and M. E. J. Newman. Random hypergraphs and their applications. *Physical Review E*, 79(6): 066118, June 2009, https://doi.org/10.1103/PhysRevE.79.066118.

[Hal02] Eran Halperin. Improved Approximation Algorithms for the Vertex Cover Problem in Graphs and Hypergraphs. *SIAM Journal on Computing*, 31(5): 1608–1623, January 2002, https://doi.org/10.1137/S0097539700381097.

[Har54] Zellig S. Harris. Distributional Structure. *WORD*, 10(2-3): 146–162, August 1954, https://doi.org/10.1080/00437956.1954.11659520.

[HF11] Cheng-Feng Hu and Shu-Cherng Fang. Set covering-based surrogate approach for solving sup- $\mathcal{T}$ equation constrained optimization problems. *Fuzzy Optimization and Decision Making*, 10(2): 125–152, June 2011, https://doi.org/10.1007/s10700-011-9099-0.

[HQ12] Shenglong Hu and Liqun Qi. Algebraic connectivity of an even uniform hypergraph. *Journal of Combinatorial Optimization*, 24(4): 564–579, November 2012, http://link.springer.com/10.1007/s10878-011-9407-1.

[HQ15] Shenglong Hu and Liqun Qi. The Laplacian of a uniform hypergraph. *Journal of Combinatorial Optimization*, 29(2): 331–366, February 2015, https://doi.org/10.1007/s10878-013-9596-x.

[Hu13] Shenglong Hu. *Spectral hypergraph theory*. PhD Thesis, The Hong Kong Polytechnic University, 2013.

[HYLW09] Qiang-Sheng Hua, Dongxiao Yu, Francis C. M. Lau, and Yuexuan Wang. Exact Algorithms for Set Multicover and Multiset Multicover Problems. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, volume 5878, pages 34–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[IT15] Ahmed Isah and Y. Tella. The Concept of Multiset Category. *British Journal of Mathematics & Computer Science*, 9(5): 427–437, January 2015, https://doi.org/10.9734/bjmcs/2015/18415.

[JCCB13] Jian Zhao, Christopher Collins, Fanny Chevalier, and Ravin Balakrishnan. Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12): 2080–2089, December 2013, https://doi.org/10.1109/TVCG.2013.167.

[JVHB14] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS ONE*, 9(6): e98679, June 2014, https://dx.plos.org/10.1371/journal.pone.0098679.

[KAF+08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics: Definition, Process, and Challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, volume 4950, pages 154–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Kar72] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Springer US, Boston, MA, 1972.

[Kat13] Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

[Kem59] John G Kemeny. Mathematics without numbers. *Daedalus*, 88(4): 577–591, 1959.

[Ken38] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2): 81–93, June 1938, https://doi.org/10.1093/biomet/30.1-2.81.

[Ken48] Maurice George Kendall. Rank correlation methods. 1948.

[KKC17] Han Kyul Kim, Hyunjoong Kim, and Sungzoon Cho. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266: 336–352, November 2017, https://doi.org/10.1016/j.neucom.2017.05.046.

[KL02] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.

[Knu68] Donald Knuth. The art of computer programming 1: Fundamental algorithms 2: Seminumerical algorithms 3: Sorting and searching 4: Generating all trees. History of Combinatorial Generation. *MA: Addison-Wesley*, 30, 1968.

[KPP+19] Bogumił Kamiński, Valérie Poulin, Paweł Prałat, Przemysław Szufel, and François Théberge. Clustering via hypergraph modularity. *PLOS ONE*, 14(11): e0224307, November 2019, https://doi.org/10.1371/journal.pone.0224307.

[LAYK+18] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S Muthukrishnan, Vishwa Vinay, and Zheng Wen. Offline evaluation of ranking policies with

click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1685–1694. ACM, 2018.

[LCL11] Jungmin Lee, Minsu Cho, and Kyoung Mu Lee. Hyper-graph matching via reweighted random walks. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1633–1640. IEEE, 2011.

[Lee13] John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013.

[Lim13] Lek-Heng Lim. Tensors and hypermatrices. *Handbook of Linear Algebra, 2nd Ed., CRC Press, Boca Raton, FL*, pages 231–260, 2013.

[Lin10] Shili Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5): 555–570, 2010.

[LL05] John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6(Jan): 129–163, 2005.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, 2014. tex.organization: Springer.

[Lo93] László Lovász and others. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is eighty*, 2(1): 1–46, 1993.

[Loe92] Daniel Loeb. Sets with a negative number of elements. *Advances in Mathematics*, 91(1): 64–74, 1992.

[LP11] Linyuan Lu and Xing Peng. High-Ordered Random Walks and Generalized Laplacians on Hypergraphs. In *WAW*, pages 14–25. Springer, 2011.

[LQY13] Guoyin Li, Liqun Qi, and Gaohang Yu. The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory. *Numerical Linear Algebra with Applications*, 20(6): 1001–1029, 2013.

[LSP16] Matthew R Lakin, Darko Stefanovic, and Andrew Phillips. Modular verification of chemical reaction network encodings via serializability analysis. *Theoretical computer science*, 632: 21–42, 2016.

[Maj87] Zofia Majcher. Alternating cycles and realizations of a degree sequence. *Commentationes Mathematicae Universitatis Carolinae*, 28(3): 467–480, 1987.

[Mäk90] Erkki Mäkinen. How to draw a hypergraph. *International Journal of Computer Mathematics*, 34(3-4): 177–185, 1990, https://doi.org/10.1080/00207169008803875.

[Mar05] AV Markovskii. On the relation between equations with max-product composition and the covering problem. *Fuzzy Sets and Systems*, 153(2): 261–273, 2005.

[MEP+14] Robert Campbell McColl, David Ediger, Jason Poovey, Dan Campbell, and David A. Bader. A Performance Evaluation of Open

Source Graph Databases. PPAA '14, pages 11–18. ACM, 2014, https://doi.org/10.1145/2567634.2567638. event-place: Orlando, Florida, USA.

[Mer95] Russell Merris. A survey of graph Laplacians. *Linear and Multilinear Algebra*, 39(1-2): 19–31, 1995.

[MHVG15] Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in Byzantine systems. *Distributed Computing*, 28(6): 423–441, 2015.

[MKL11] Hao Ma, Irwin King, and Michael R Lyu. Mining web graphs for recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 24(6): 1051–1064, 2011.

[MN12] Tom Michoel and Bruno Nachtergaele. Alignment and integration of complex networks by hypergraph-based spectral clustering. *Physical Review E*, 86(5): 056111, November 2012, https://doi.org/10.1103/PhysRevE.86.056111.

[MPL17] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics Rep.*, 2017.

[MSCM16] Venkatesh N. Murthy, Avinash Sharma, Visesh Chari, and R. Manmatha. Image Annotation using Multi-scale Hypergraph Heat Diffusion Framework. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval - ICMR '16*, pages 299–303, New York, New York, USA, 2016. ACM Press, https://doi.org/10.1145/2911996.2912055.

[MSWL18] Shuming Ma, Xu Sun, Yizhong Wang, and Junyang Lin. Bag-of-words as target for neural machine translation. *arXiv preprint arXiv:1805.04871*, 2018.

[MWW+17] Shervin Minaee, Siyun Wang, Yao Wang, Sohae Chung, Xiuyuan Wang, Els Fieremans, Steven Flanagan, Joseph Rath, and Yvonne W. Lui. Identifying mild traumatic brain injury patients from MR images using bag of visual words. In *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–5, Philadelphia, PA, December 2017. IEEE, https://doi.org/10.1109/SPMB.2017.8257054.

[MYLK08] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 233–242. ACM, 2008.

[New01a] M. E. J. Newman. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64(1): 016131, June 2001, https://doi.org/10.1103/PhysRevE.64.016131.

[New01b] M. E. J. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1): 016132, June 2001, https://link.aps.org/doi/10.1103/PhysRevE.64.016132.

[Nik00] Mila Nikolova. Local Strong Homogeneity of a Regularized Estimator. *SIAM Journal on Applied Mathematics*, 61(2): 633–658, January 2000, https://doi.org/10.1137/S0036139997327794.

[NJY+10] Nan Cao, Jimeng Sun, Yu-Ru Lin, D Gotz, Shixia Liu, and Huamin Qu. FacetAtlas: Multifaceted Visualization for Rich Text Corpora. *IEEE Transactions on Visualization and Computer Graphics*, 16(6): 1172–1181, November 2010, https://doi.org/10.1109/TVCG.2010.154.

[No85] Jean Antoine Nicolas and others. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Par M. le marquis de Condorcet....* de l'Imprimerie Royale, 1785.

[OGMM20] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Tuning ranking in co-occurrence networks with general biased exchange-based diffusion on hyper-bag-graphs. 2020. arXiv: 2003.07323 [cs.SI].

[OLGMM17a] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Adjacency and Tensor Representation in General Hypergraphs Part 1: e-adjacency Tensor Uniformisation Using Homogeneous Polynomials. *arXiv preprint arXiv:1712.08189*, 2017.

[OLGMM17b] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Networks of Collaborations: Hypergraph Modeling and Visualisation. *arXiv preprint arXiv:1707.00115*, 2017.

[OLGMM18a] Xavier Ouvrard, Jean-Marie Le Goff, and S. Marchand-Maillet. On Adjacency and e-Adjacency in General Hypergraphs: Towards a New e-Adjacency Tensor. *Electronic Notes in Discrete Mathematics*, 70: 71–76, December 2018, https://doi.org/10.1016/j.endm.2018.11.012.

[OLGMM18b] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Adjacency and Tensor Representation in General Hypergraphs. Part 2: Multisets, Hb-graphs and Related e-adjacency Tensors. *arXiv preprint arXiv:1805.11952*, 2018.

[OLGMM18c] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Diffusion by Exchanges in HB-Graphs: Highlighting Complex Relationships. *CBMI Proceedings*, 2018, https://doi.org/10.1109/CBMI.2018.8516525.

[OLGMM18d] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Hypergraph Modeling and Visualisation of Complex Co-occurence Networks. *Electronic Notes in Discrete Mathematics*, 70: 65–70, December 2018, https://doi.org/10.1016/j.endm.2018.11.011.

[OLGMM19a] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Diffusion by Exchanges in HB-Graphs: Highlighting Complex Relationships Extended version. *Arxiv:1809.00190v2*, 2019.

[OLGMM19b] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. The HyperBagGraph DataEdron: An Enriched Browsing Experience of Multimedia Datasets. *arXiv preprint arXiv:1905.11695*, 2019.

[OLGMM19c] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. On Hb-graphs and their Application to General Hypergraph e-adjacency Tensor. *MCCCC32 Special Volume of the Journal of Combinatorial Mathematics and Combinatorial Computing, to be published*, 2019.

[OLGMM20] Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. The HyperBagGraph DataEdron: An Enriched Browsing Experience of Datasets. *LNCS, 46th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2020)*, 2020.

[Ouv20] Xavier Ouvrard. Hypergraphs: an introduction and review. *arXiv preprint arXiv:2002.05014*, 2020.

[Pău11] Gheorghe Păun. *Membrane Computing*, volume 6501 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. Citation Key Alias: paun2012membrane.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[PK08] Ruzica Piskac and Viktor Kuncak. Decision Procedures for Multisets with Cardinality Constraints. In *Verification, Model Checking, and Abstract Interpretation*, volume 4905, pages 218–232. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Po00] Phillip Poplin and others. The Semiring of Multisets. 2000.

[PS15] Lynnette Purda and David Skillicorn. Accounting Variables, Deception, and a Bag of Words: Assessing the Tools of Fraud Detection. *Contemporary Accounting Research*, 32(3): 1193–1223, September 2015, https://doi.org/10.1111/1911-3846.12089.

[PSS05] S.U. Pillai, T. Suel, and Seunghun Cha. The Perron-Frobenius theorem: some of its applications. *IEEE Signal Processing Magazine*, 22(2): 62–75, March 2005, https://doi.org/10.1109/MSP.2005.1406483.

[Pu13] Li Pu. Relational Learning with Hypergraphs. 2013.

[PVAdST19] Daniel Carlos Guimaraes Pedronette, Lucas Pascotti Valem, Jurandy Almeida, and Ricardo da S. Torres. Multimedia Retrieval Through Unsupervised Hypergraph-Based Manifold Ranking. *IEEE Transactions on Image Processing*, 28(12): 5824–5838, December 2019, https://doi.org/10.1109/TIP.2019.2920526.

[PWWQ16] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150: 109–125, September 2016, https://doi.org/10.1016/j.cviu.2016.03.013.

[PZ14] Kelly J. Pearson and Tan Zhang. On Spectral Hypergraph Theory of the Adjacency Tensor. *Graphs and Combinatorics*, 30(5): 1233–1248, September 2014, https://doi.org/10.1007/s00373-013-1340-x.

[PZ15] Kelly J. Pearson and Tan Zhang. The Laplacian tensor of a multi-hypergraph. *Discrete Mathematics*, 338(6): 972–982, June 2015, https://doi.org/10.1016/j.disc.2015.01.021.

[Qi14] Liqun Qi. H^+-eigenvalues of Laplacian and signless Laplacian tensors. *Communications in Mathematical Sciences*, 12(6): 1045–1064, 2014, https://doi.org/10.4310/CMS.2014.v12.n6.a3.

[QL17] Liqun Qi and Ziyan Luo. *Tensor Analysis: Spectral Theory and Special Tensors*. Society for Industrial and Applied Mathematics, Philadelphia, PA, April 2017.

[Rad15] Aurelian Radoaca. Simple Venn Diagrams for Multisets. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2015 17th International Symposium on*, pages 181–184. IEEE, 2015.

[Ran45] Shiyali Ramamrita Ranganathan. *Elements of library classification: based on lectures delivered at the University of Bombay in December 1944*. NK Publishing House, Poona, 1945.

[Ran62] Shiyali Ramamrita Ranganathan. *Elements of library classification*. 1962.

[RKS17] MR Gauthama Raman, Kannan Kirthivasan, and VS Shankar Sriram. Development of rough set–hypergraph technique for key feature identification in intrusion detection systems. *Computers & Electrical Engineering*, 59: 189–200, 2017.

[Rod03] J.A. Rodríguez. On the Laplacian Spectrum and Walk-regular Hypergraphs. *Linear and Multilinear Algebra*, 51(3): 285–297, September 2003, https://doi.org/10.1080/0308108031000084374.

[RV93] Sridhar Rajagopalan and Vijay V Vazirani. Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1993.

[Sha13] Jia-Yu Shao. A general product of tensors with applications. *Linear Algebra and its Applications*, 439(8): 2350–2366, October 2013, https://doi.org/10.1016/j.laa.2013.07.010.

[SIYS07] D Singh, A Ibrahim, T Yohanna, and J Singh. An overview of the applications of multisets. *Novi Sad Journal of Mathematics*, 37(3): 73–92, 2007.

[SJ12] Ravi Shekhar and C.V. Jawahar. Word Image Retrieval Using Bag of Visual Words. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 297–301, Gold Coast, Queenslands, TBD, Australia, March 2012. IEEE, https://doi.org/10.1109/DAS.2012.96.

[SJP+16] Maximilian Schmitt, Christoph Janott, Vedhas Pandit, Kun Qian, Clemens Heiser, Werner Hemmert, and Björn Schuller. A bag-of-audio-words approach for snore sounds' excitation localisation. In *Speech Communication; 12. ITG Symposium*, pages 1–5. VDE, 2016.

[SK12] Barna Saha and Samir Khuller. Set Cover Revisited: Hypergraph Cover with Hard Capacities. In *Automata, Languages, and Programming*, volume 7391, pages 762–773. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Spe87] C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 100(3/4): 441, 1987, https://doi.org/10.2307/1422689.

[SPS85] Jeanette Schmidt-Pruzan and Eli Shamir. Component structure in the evolution of random hypergraphs. *Combinatorica*, 5(1): 81–94, March 1985, https://doi.org/10.1007/BF02579445.

[ST09] Ileana Streinu and Louis Theran. Sparse hypergraphs and pebble game algorithms. *European Journal of Combinatorics*, 30(8): 1944–1964, November 2009, https://doi.org/10.1016/j.ejc.2008.12.018.

[Ste12] John G. Stell. Relations on Hypergraphs. In *Relational and Algebraic Methods in Computer Science*, volume 7560, pages 326–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[SVK16] Elias C. Stavropoulos, Vassilios S. Verykios, and Vasileios Kagklis. A transversal hypergraph approach for the frequent itemset hiding problem. *Knowledge and Information Systems*, 47(3): 625–645, June 2016, https://doi.org/10.1007/s10115-015-0862-3.

[SWG+18] Fernanda B. Silva, Rafael de O. Werneck, Siome Goldenstein, Salvatore Tabbone, and Ricardo da S. Torres. Graph-based bag-of-words for classification. *Pattern Recognition*, 74: 266–285, February 2018, https://doi.org/10.1016/j.patcog.2017.09.018.

[Syr01] Apostolos Syropoulos. Mathematics of Multisets. In *Multiset Processing*, volume 2235, pages 347–358. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[Syr03] Apostolos Syropoulos. Categorical models of multisets. *Romanian Journal of Information Science and Technology*, 6(3-4): 393–400, 2003.

[SZ03] Sivic and Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2, Nice, France, 2003. IEEE, https://doi.org/10.1109/ICCV.2003.1238663.

[SZ09] Frans Schalekamp and Anke van Zuylen. Rank Aggregation: Together We're Strong. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 38–51. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 2009.

[SZB19] Lizhu Sun, Jiang Zhou, and Changjiang Bu. Spectral properties of general hypergraphs. *Linear Algebra and its Applications*, 561: 187–203, January 2019, https://doi.org/10.1016/j.laa.2018.09.029.

[Tar14] Paul Tarau. Towards a generic view of primality through multiset decompositions of natural numbers. *Theoretical Computer Science*, 537: 105–124, June 2014, https://doi.org/10.1016/j.tcs.2014.04.025.

[TCR10] Carla Taramasco, Jean-Philippe Cointet, and Camille Roth. Academic team formation as evolving hypergraphs. *Scientometrics*, 85(3): 721–740, December 2010, https://doi.org/10.1007/s11192-010-0226-4.

[TDKF17] Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard. Learning Heat Diffusion Graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3): 484–499, September 2017, https://doi.org/10.1109/TSIPN.2017.2731164.

[TN11] Hung-Khoon Tan and Chong-Wah Ngo. Fusing heterogeneous modalities for video and image re-ranking. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval - ICMR '11*, pages 1–8, Trento, Italy, 2011. ACM Press, https://doi.org/10.1145/1991996.1992011.

[To98] Michel Truchon and others. *An extension of the Condorcet criterion and Kemeny orders*. Citeseer, 1998.

[Tsa12] Chih-Fong Tsai. Bag-of-Words Representation in Image Annotation: A Review. *ISRN Artificial Intelligence*, 2012: 1–19, 2012, https://doi.org/10.5402/2012/376804.

[vHP09] F. van Ham and A. Perer. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6): 953–960, November 2009, https://doi.org/10.1109/TVCG.2009.108.

[Vol09] Vitaly I Voloshin. *Introduction to graph and hypergraph theory*. Nova Science Publ., 2009.

[WDA+16] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, and others. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[WZS16] Leiquan Wang, Zhicheng Zhao, and Fei Su. Efficient multi-modal hypergraph learning for social image classification with complex label correlations. *Neurocomputing*, 171: 242–251, January 2016, https://doi.org/10.1016/j.neucom.2015.06.064.

[XDYF16] Zihang Xu, Junping Du, Lingfei Ye, and Dan Fan. Multi-feature indexing for image retrieval based on hypergraph. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 494–500, Beijing, China, August 2016. IEEE, https://doi.org/10.1109/CCIS.2016.7790309.

[XHLJ18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Yag86] Ronald R. Yager. ON THE THEORY OF BAGS. *International Journal of General Systems*, 13(1): 23–37, November 1986, https://doi.org/10.1080/03081078608934952.

[YL78] H. P. Young and A. Levenglick. A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics*, 35(2): 285–300, September 1978, http://epubs.siam.org/doi/10.1137/0135023.

[ZGC10] Vinko Zlatić, Andrea Gabrielli, and Guido Caldarelli. Topologically biased random walk and community finding in networks. *Physical Review E*, 82(6): 066109, December 2010, https://doi.org/10.1103/PhysRevE.82.066109.

[ZHS07] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, pages 1601–1608, 2007.

[ZSC99] J.Y. Zien, M.D.F. Schlag, and P.K. Chan. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(9): 1389–1399, September 1999, https://doi.org/10.1109/43.784130.

[ZTH14] Li-Jun Zhao, Ping Tang, and Lian-Zhi Huo. Land-Use Scene Classification Using a Concentric Circle-Structured Multiscale Bag-of-Visual-Words Model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12): 4620–4631, December 2014, https://doi.org/10.1109/JSTARS.2014.2339842.

[ZWHC06] Tao Zhou, Bing-Hong Wang, P.M. Hui, and K.P. Chan. Topological properties of integer networks. *Physica A: Statistical Mechanics and its Applications*, 367: 613–618, July 2006, https://linkinghub.elsevier.com/retrieve/pii/S0378437105011969.

Index

Appendices

A. List of talks and conferences

Date	Conference	Title	Kind
		Computer science	
05.2017	GPU-Days Wigner Budapest 2017	Challenges in visualizing large graphs and hypergraphs.	Talk
09.2018	CBMI 2018	Exchange-Based Diffusion in Hb-Graphs: Highlighting Complex Relationships.	Talk & article
10.2018	CUSO Linguistic & CS	Hb-graphs and their application to textual datasets.	Talk
01.2019	CUSO CS	HyperBagGraph DataEdron: Modeling and Visualisation of Complex Co-occurence Networks.	Poster
01.2019	AMLD	The DataEdron of HyperBag-Graphs: An Enriched Browsing Experience of Scientific Publications.	Poster
05.2019	IDIAP Valais/Wallis AI Workshop 5th edition Interpreting machine learning	The HyperBagGraph DataEdron: An Enriched Browsing Experience of Scientific Publication Databases.	Talk
01.2020	SOFSEM 2020 Cyprus	The HyperBagGraph DataEdron: An Enriched Browsing Experience of Multimedia Datasets.	Talk & article
		Mathematics	
10.2017	APMEP	Graphes et hypergraphes @ CERN	Talk
09.2018	2nd IMA Derby	On Adjacency and e-Adjacency in General Hypergraphs: Towards a New e-Adjacency Tensor.	Talk & short article
09.2018	2nd IMA Derby	Hypergraph Modeling and Visualisation of Complex Co-occurrence Networks.	Poster & short article
10.2019	MCCC32 Duluth	On Hb-graphs and their Application to General Hypergraph e-adjacency Tensor.	Abstract & talk & Journal
07.2019	CTW 2019 Twente	Multi-diffusion in Hb-Graphs.	Abstract & talk
07.2019	SIAM/ICIAM 2019	The HyperBagGraph DataEdron: An Enriched Browsing Experience of Scientific Publication Databases.	Poster & Talk
09.2019	SEME / AMIES Toulouse	Analyse sémantique du langage naturelle.	Talk & Rapport
01.2020	Low rank models SIAM/CUSO 2020	Hyper-bag-graphs and general hyper(-bag-)graph e-adjacency tensors	Poster
		Physics	
04.2018	CERN Doctoral poster session	Hypergraph Modeling and Visualisation of Complex Collaboration Networks.	Poster
05.2019	CCEGN	Hb-graphs and their Applications.	Poster & informal talk
11.2019	AIME Budapest	Hb-graphs and their Applications.	Talk
		Other	
11.2019	SES Pune	The HyperBagGraph DataEdron: An Enriched Browsing Experience of Scientific Publication Databases.	Poster

Table A.1.: List of talks and posters related to this Thesis—as of 07.12.2019.

B. Mathematical background

This Chapter is based on [OLGMM17a], [OLGMM17b], [OLGMM18b], [OLGMM18c], [OLGMM19a], [OLGMM19c] and [Ouv20]. It provides all the necessary background concepts used in this Thesis.

B.1. Hypergraphs

[Ber67] introduces hypergraphs as a means to generalize graphs. Graphs only support pairwise relationships. Hypergraphs preserve the multi-adic relationships and, therefore, become a natural modeling of collaboration networks and various other situations. They already allow a huge step in modeling, but some limitations remain that will be discussed further in this Thesis that has pushed us to introduce an extension of hypergraphs. In this Section, we aim at giving a short synthesis on hypergraphs based on [Ouv20] and introduce the needed notations used throughout this Thesis. The interested reader can refer to [Ouv20] for a full introduction on hypergraphs and their applications.

B.1.1. Generalities

As given in [Ber73], a **hypergraph** $\mathcal{H} = (V, E)$ on a finite **set of vertices** (or nodes) $V \triangleq \{v_i : i \in [\![n]\!]\}$[1] is defined as a family of hyperedges $E \triangleq (e_j)_{j \in [\![p]\!]}$ where each **hyperedge** is a non-empty subset of V and such that $\bigcup_{j \in [\![p]\!]} e_j = V$.

It means that in a hypergraph, a hyperedge links one or more vertices. In [Vol09], the definition of hypergraphs includes also hyperedges that are empty sets as hyperedges are defined as a family of subsets of a finite vertex set and it is not necessary that the union covers the vertex set. Both the vertex set and the family of hyperedges can be empty; it they are at the same time, the hypergraph is then designated as the **empty hypergraph**. This definition of hypergraph opens their use in various collaboration networks. It is the one we choose in this Thesis.

In [Bre13], an intermediate definition is taken, relaxing only the covering of the vertex set by the union of the hyperedges enabling only isolated vertices in hypergraphs.

Other interesting definitions of hypergraphs, based on binary relations, exist that are given in [Ste12]; a full comparison is achieved in [Ouv20].

When needed, elements of V will be written $(v_i)_{i \in [\![n]\!]}$ and those of E will be written as $(e_j)_{j \in [\![p]\!]}$. Abusively, e_j will also designate the subset $\iota(e_j)$ of V.

[1] We write for $n \in \mathbb{N}^* : [\![n]\!] = \{i : i \in \mathbb{N}^* \wedge i \leqslant n\}$

Two hyperedges $e_{j_1} \in E$ and $e_{j_2} \in E$, with $j_1, j_2 \in [\![p]\!]$ and $j_1 \neq j_2$ such that $e_{j_1} = e_{j_2}$ are said **repeated hyperedges**.

A hypergraph is said with **no repeated hyperedge**, if it has no repeated hyperedges.

Following [CF88], where the hyperedge family is viewed as a multiset of hyperedges, a hypergraph with repeated hyperedges is called a **multi-hypergraph**.

A hypergraph is said **simple**, if for any two hyperedges $e_{j_1} \in E$ and $e_{j_2} \in E$:

$$e_{j_1} \subseteq e_{j_2} \Rightarrow j_1 = j_2.$$

Hence, a simple hypergraph has no repeated hyperedges.

A hyperedge $e \in E$ such that: $|e| = 1$ is called a **loop**.

A hypergraph is said **linear** if it is simple and such that every pair of hyperedges shares at most one vertex.

A **sub-hypergraph** $\mathcal{K}$ of a hypergraph $\mathcal{H}$ is the hypergraph formed by a subset W of the vertices of $\mathcal{H}$ and the hyperedges of $\mathcal{H}$ that are subsets of W.

A **partial hypergraph** $\mathcal{H}'$ generated by a subset $E' \subseteq E$ of the hyperedges is a hypergraph containing exactly these hyperedges and whose vertex set contains at least all the vertices incident to this hyperedge subset.

The **star of a vertex** $v \in V$ of a hypergraph $\mathcal{H}$, written $H(v)$, is the family of hyperedges containing v.

The **dual** of a hypergraph $\mathcal{H}$ is the hypergraph $\mathcal{H}^*$ whose vertices corresponds to the hyperedges of $\mathcal{H}$ and the hyperedges of $\mathcal{H}^*$ are the vertices of $\mathcal{H}$, with the incident relation that links each vertex to its star.

The **neighborhood of a vertex** $v \in V$ is the set $\Gamma(v)$ of vertices that belongs to the hyperedges this vertex is belonging.

B.1.2. Weighted hypergraph

In [ZHS07], the definition of a weighted hypergraph is given, based on the definition of [Ber73] of a hypergraph.

$\mathcal{H}_{w_e} = (V, E, w_e)$ is a **weighted hypergraph**, if (V, E) is a hypergraph and $w_e : E \to \mathbb{R}$ is a function that associates to each hyperedge $e \in E$ a weight $w_e(e)$.

We can refine this definition to handle weights on individual vertices, by using a second function $w_v : V \to \mathbb{R}$ that associates to each vertex $v \in V$ a weight $w_v(v)$. But putting weights that are hyperedge dependent cannot be achieved with hypergraphs as it would imply to move to a new algebra, as we will see with the introduction of hb-graphs.

B.1.3. Hypergraph features

Hypergraph features are very similar to those of graphs with some arrangements to account for their differences in structure.

The **order** $o_{\mathcal{H}}$ of a hypergraph $\mathcal{H}$ is defined as $o_{\mathcal{H}} \triangleq |V|$.

The **rank** $r_{\mathcal{H}}$ of a hypergraph $\mathcal{H}$ is the maximum of the cardinalities of the hyperedges:

$$r_{\mathcal{H}} \triangleq \max_{e \in E} |e| ,$$

while the **anti-rank** $s_{\mathcal{H}}$ corresponds to the minimum:

$$s_{\mathcal{H}} \triangleq \min_{e \in E} |e| .$$

The **degree of a vertex** $v_i \in V$, written $\mathbf{deg}(v_i) = d_i$, corresponds to the number of hyperedges that this vertex participates in. Hence:

$$\deg(v_i) \triangleq |H(v_i)| .$$

It is also designated as **hyper-degree** in some articles.

B.1.4. Paths and related notions

A **path** $v_{i_0} e_{j_1} v_{i_1} \dots e_{j_s} v_{i_s}$ in a hypergraph $\mathcal{H}$ from a vertex u to a vertex v is a vertex / hyperedge alternation with s hyperedges e_{j_k} such that: $\forall k \in [\![s]\!], j_k \in [\![p]\!]$ and $s+1$ vertices v_{i_k} with $\forall k \in \{0\} \cup [\![s]\!], i_k \in [\![n]\!]$ and such that $v_{i_0} = u$, $v_{i_s} = v$, $u \in e_{j_1}$ and $v \in e_{j_s}$ and that for all $k \in [\![s-1]\!]$, $v_{i_k} \in e_{j_k} \cap e_{j_{k+1}}$.

The **length of a path** from u to v is the number of hyperedges it traverses; given a path $\mathscr{P}$, we write $l(\mathscr{P})$ its length. It holds that if $\mathscr{P} = v_{i_0} e_{j_1} v_{i_1} \dots e_{j_s} v_{i_s}$, we have: $l(\mathscr{P}) = s$.

The **hypergraph distance** $d(u, v)$ between two vertices u and v of a hypergraph is the length of the shortest path between u and v, if there exists, that can be found in the hypergraph. In the case where there is no path between the two vertices, they are said to be **disconnected**, and we set: $d(u, v) = +\infty$. A hypergraph is said to be **connected** if there exists a path between every pair of vertices of the hypergraph, and **disconnected** otherwise.

A **connected component** of a hypergraph is a maximal subset of the vertex set for which there exists a path between any two vertices of this maximal subset in the hypergraph.

B.1.5. Multi-graph, graph, 2-section

A hypergraph with rank at most 2 is called a **multi-graph**. A simple multi-graph without loop is a **graph**.

For the moment, we keep the original concept of adjacency as it is implicitly given in [Bre13]; we mention it here as **2-adjacency** since it is a pairwise adjacency.

Two distinct vertices $v_{i_1} \in V$ and $v_{i_2} \in V$ are said **2-adjacent** if there exists $e \in E$ such that $v_{i_1} \in e$ and $v_{i_2} \in e$.

The graph $[\mathcal{H}]_2 \triangleq \left(V_{[2]}, E_{[2]}\right)$ obtained from a hypergraph $\mathcal{H} = (V, E)$ by considering: $V_{[2]} \triangleq V$ and such that if v_{i_1} and v_{i_2} are 2-adjacent in $\mathcal{H}$, $\{v_{i_1}, v_{i_2}\} \in E_{[2]}$ is called the **2-section of the hypergraph** $\mathcal{H}$.

The graph $[\mathcal{H}]_{\mathcal{I}} \triangleq (V_{\mathcal{I}}, E_{\mathcal{I}})$ obtained from a hypergraph $\mathcal{H} = (V, E)$ by considering: $V_{\mathcal{I}} \triangleq V^*$ and, such that, if $e_{j_1} \in E$ and $e_{j_2} \in E$—with $j_1 \neq j_2$—are intersecting hyperedges in $\mathcal{H}$, then $\{e_{j_1}, e_{j_2}\} \in E_{\mathcal{I}}$, is called the **intersection graph** of the hypergraph $\mathcal{H}$.

Let $k \in \mathbb{N}^*$. A hypergraph is said to be ***k*-uniform** if all its hyperedges have the same cardinality k.

A **directed hypergraph** $\mathcal{H} = (V, E)$ on a finite set of n vertices (or vertices) $V = \{v_i : i \in [\![n]\!]\}$ is defined as a family of p **hyperedges** $E = (e_j)_{j \in [\![p]\!]}$ where each hyperedge e_j contains exactly two non-empty subsets of V, one which is called the **source**—written $e_{s\,j}$—and the other one which is the **target**—written $e_{t\,j}$. A hypergraph that is not directed is said to be an **undirected hypergraph**.

The **incidence graph** $[\mathcal{H}]_L = (V_L, E_L)$—or Levi graph—of a hypergraph $\mathcal{H} = (V, E)$ is the bipartite graph of vertex set $V_L = V \cup V_E$ where V_E is the set of vertices v_e corresponding to each hyperedge e of E and where $(v, v_e) \in E_L$ if $v \in V$ and $v \in e$.

It is worth mentioning that the incidence graph confuses the hypergraph with its dual:

$$[\mathcal{H}]_L = [\mathcal{H}^*]_L \,.$$

B.1.6. Sum of hypergraphs

Let $\mathcal{H}_1 = (V_1, E_1)$ and $\mathcal{H}_2 = (V_2, E_2)$ be two hypergraphs. The **sum** of these two hypergraphs is the hypergraph written $\mathcal{H}_1 + \mathcal{H}_2$ defined as:

$$\mathcal{H}_1 + \mathcal{H}_2 \triangleq (V_1 \cup V_2, E_1 \cup E_2) \,.$$

This sum is said **direct** if $E_1 \cap E_2 = \emptyset$. In this case, the sum is written $\mathcal{H}_1 \oplus \mathcal{H}_2$.

B.1.7. Matrices associated to hypergraphs

B.1.7.1. Incidence matrix

The **incidence matrix**:

$$H \triangleq [h_{ij}]_{\substack{i \in [\![n]\!] \\ j \in [\![p]\!]}}$$

of a hypergraph is the matrix having rows indexed by the corresponding indices of vertices of $\mathcal{H}$ and columns by the hyperedge indices, and where the coefficient $h_{ij} \triangleq 1$ when $v_i \in e_j$, and $h_{ij} \triangleq 0$ when $v_i \notin e_j$.

B.1.7.2. Adjacency matrix

We focus in this paragraph on the pairwise adjacency as defined in [Ber73] and [Bre13]. In the latter, the **adjacency matrix of a hypergraph** $\mathcal{H}$ is defined as the square

matrix $A \triangleq [a_{ij}]$ having rows and columns indexed by indices corresponding to the indices of vertices of $\mathcal{H}$ and where for all $i, j \in [\![p]\!]$, $i \neq j$:

$$a_{ij} \triangleq |\{e \in E \, : \, v_i \in e \wedge v_j \in e\}|$$

and for all $i \in [\![p]\!]$:

$$a_{ii} \triangleq 0.$$

It holds, following [ERV05]:

$$A = HH^{\intercal} - D_V$$

where $D_V \triangleq \text{diag}\left((d_i)_{i \in [\![n]\!]}\right)$ is the diagonal matrix containing vertex degrees.

The **adjacency matrix of a weighted hypergraph** $\mathcal{H}_w$ is defined in [ZHS07] as the matrix A_w of size $n \times n$ defined as:

$$A_w \triangleq HWH^{\intercal} - D_w$$

where:

$$W \triangleq \text{diag}\left((w_j)_{j \in [\![p]\!]}\right)$$

is a diagonal matrix of size $p \times p$ containing the weights $(w_j)_{j \in [\![p]\!]}$ of the respective hyperedges $(e_j)_{j \in [\![p]\!]}$ and D_w is the diagonal matrix of size $n \times n$:

$$D_w \triangleq \text{diag}\left((d_w(v_i))_{i \in [\![n]\!]}\right)$$

and where for all $i \in [\![n]\!]$:

$$d_w(v_i) \triangleq \sum_{j \in \{k : k \in [\![p]\!] \wedge v_i \in e_k\}} w_j$$

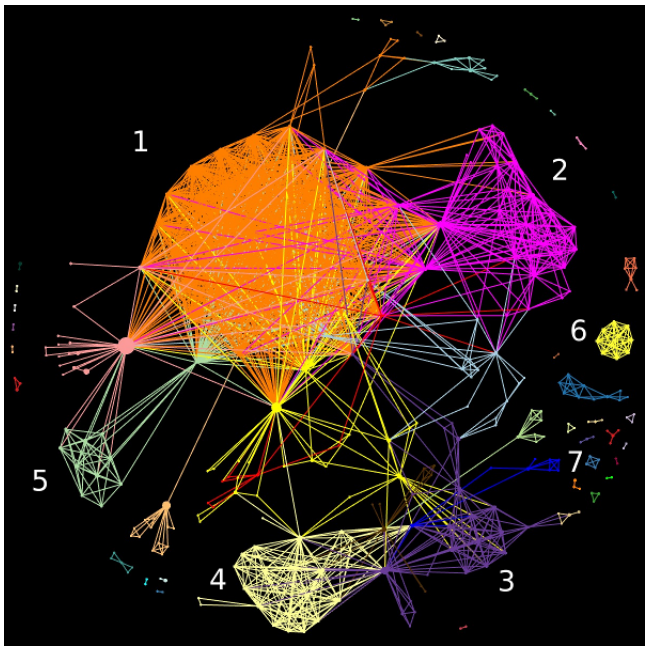
is the **weighted degree** of the vertex $v_i \in V$.

We will refine the concept of adjacency in general hypergraphs in the next Chapter.

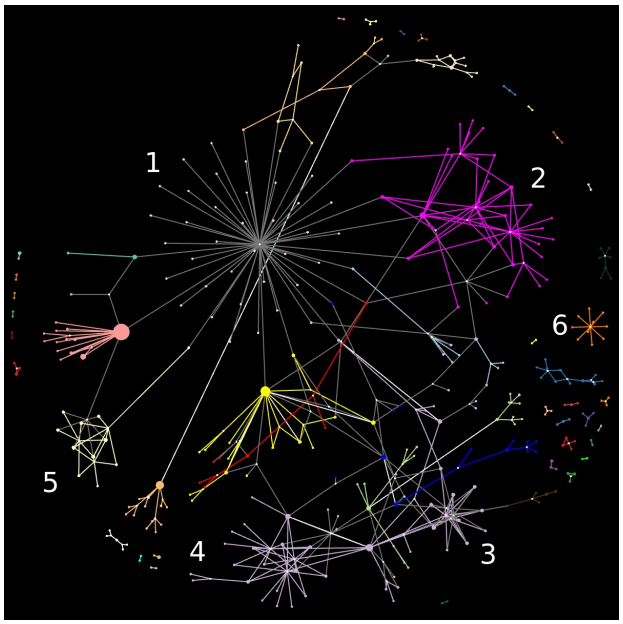
B.1.8. Hypergraph visualisation

In [Mäk90], hypergraph visualizations are classified in two categories called the "subset standard" and the "edge standard". These two types of representations reflect the two facets of hypergraphs. The **subset standard** reflects that hyperedges are subsets of the vertex set: the vertices of a hyperedge are drawn as points and hyperedges as closed envelopes. The **edge standard** reflects that vertices of a hyperedge maintain together a multi-adic relationship: two main representations are the **clique representation**, which corresponds to the 2-section graph of the hypergraph and the **extra-node representation**, which corresponds to the incidence graph—also called Levi graph—of the hypergraph.

To have full details on these representations, and others like the Zykov representation, the reader can refer to [Ouv20]. We just tackle here the problem of large hypergraph visualization, which is central in some ways, to the foundations of this Thesis.



Sub-figure B.1 (a): Clique representation: The coordinates of the nodes are calculated by ForceAtlas2 on the extra-node view and then transferred to this view.



Sub-figure B.1 (b): Extra-node representation: The coordinates of the nodes are calculated by ForceAtlas2 for this representation.

Figure B.1.: Hypergraph of organizations: Sub-figures (a) and (b) refer to the search: title:((bgo AND cryst*) OR (bgo AND calor*)) abstract:((bgo AND cryst*) OR (bgo AND calor*)) from [OLGMM17b].

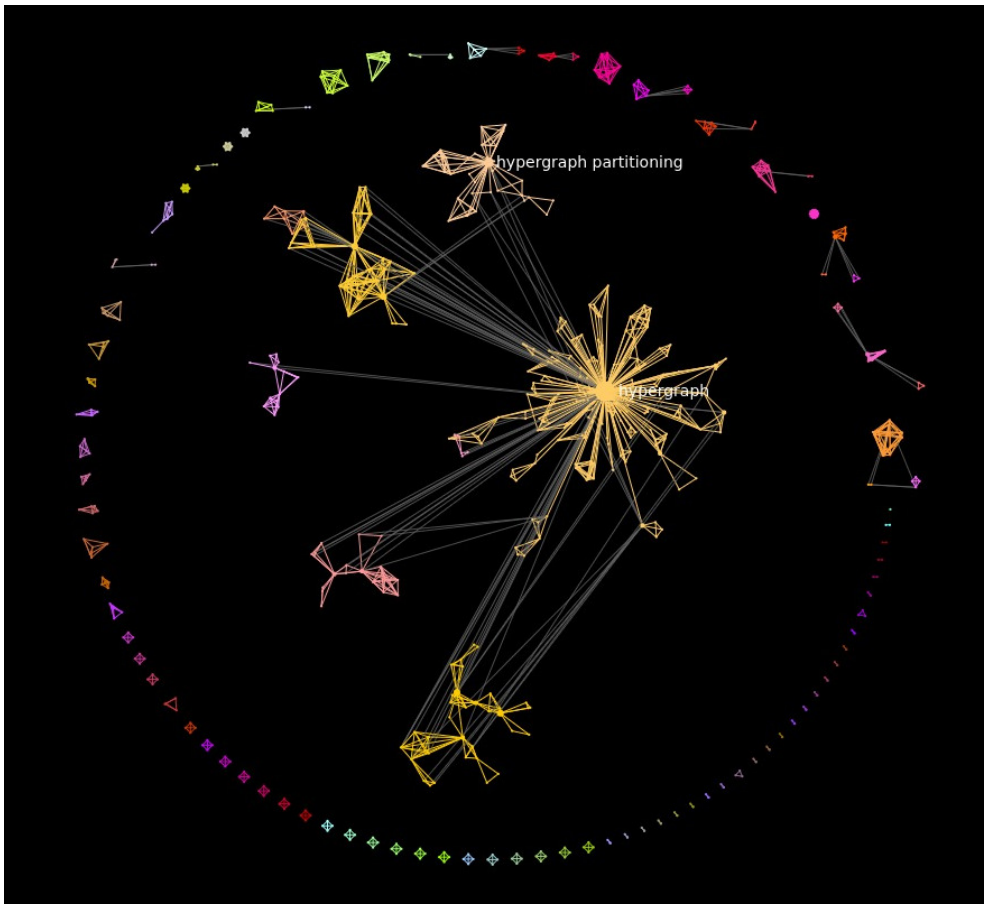


Figure B.2.: Keyword collaborations from search: "TITLE: hypergraph".

#publications = 200, #nodes = 707, #edges = 1655, #clusters = 102, #isolated clusters = 67.

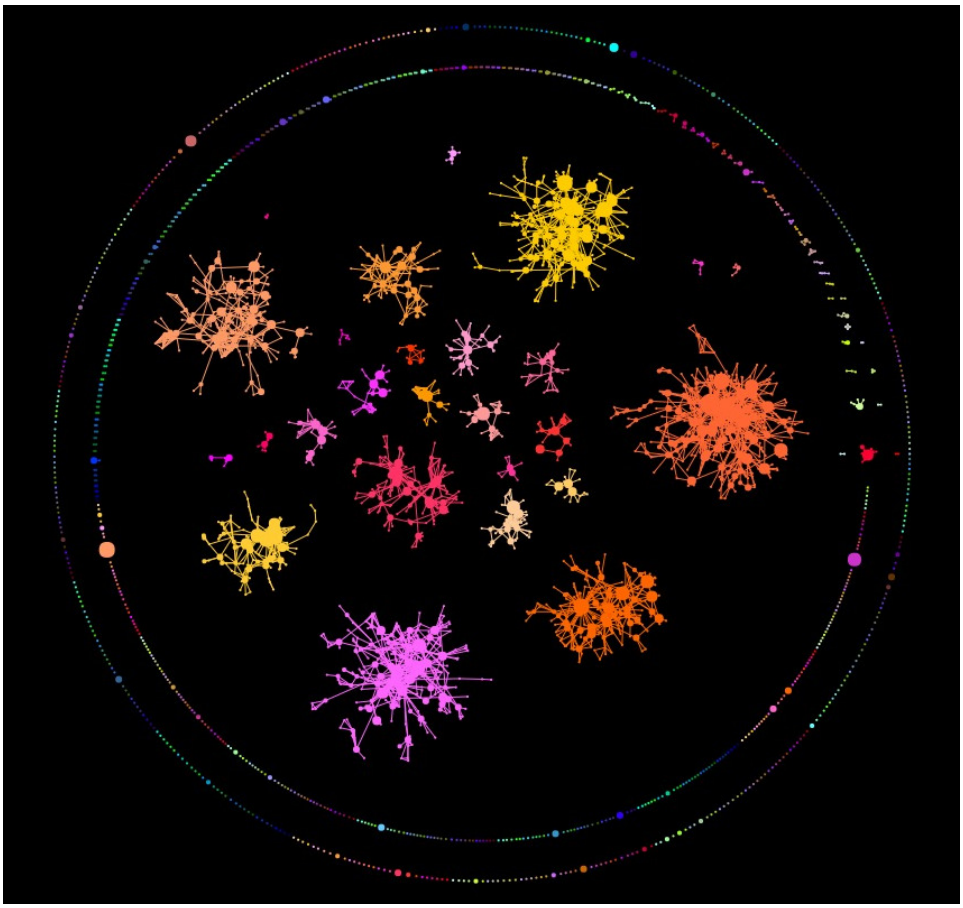


Figure B.3.: Organization collaborations from search: “TITLE:graph”.

#publications = 3969, #patents=893
#nodes = 2932, #edges = 4731, #clusters = 951, #isolated clusters = 914.

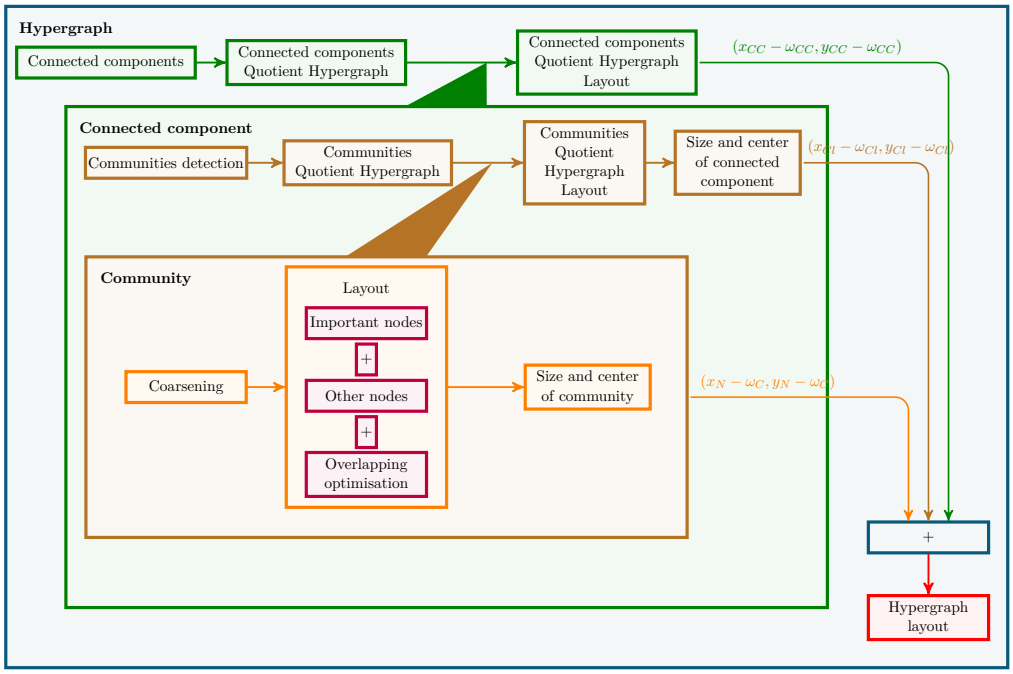


Figure B.4.: Principle of the calculation of coordinates for large hypergraphs.

Large hypergraphs require strategies to be properly visualized. We have shown in [OLGMM17b] that switching from the 2-section representation to the incidence representation reduces considerably the overall cognitive load, as the number of edges to be represented is effectively highly diminished in real networks; moreover, true collaborations are seen more distinguishably. An example is given in Figure B.1, that shows how the representation of the same hypergraph in two different modes impact the cognitive load of the information that is represented.

Complex co-occurrence networks can be modeled in a first instance by modeling them by hypergraphs as we have presented it in [OLGMM18d], where the hyperedges represent co-occurrences attached to a given reference; we will refine this approach in Chapter 6 using hb-graphs. When scaling up, such representations of large hypergraphs require different strategies in order to improve the global layout and the information displaying. These strategies include the segmentation of the hypergraph into connected components: they are ordered firstly by the number of references attached to the co-occurrences constituting them and secondly with respect to the number of vertices within co-occurrences they contain. The layout of each connected component is computed separately, by considering the communities based on the Louvain algorithm presented in [BGLL08] and using a force directed algorithm, named ForceAtlas2, presented in [JVHB14], to ensure the layout of the communities and of the vertices inside the communities. To achieve a proper layout, vertices of importance for the community are placed first and the remaining vertices are then placed around those fixed important vertices, with an optimization of the layout. The principle of the calculus is given in Figure B.4 and the results is shown in Figure B.2 and Figure B.3.

Optimizing the layout of a large hypergraph is a challenging task. It requires to capture

important vertices of the hypergraph, i.e. the ones we do not want to loose by overwhelming or overlapping in the representation. This requirement has been the starting point of our quest for a diffusion tensor.

B.2. Multisets

This section is extracted from a section of [OLGMM19c].

B.2.1. Generalities

The concept of multisets is known from the ancient times. [Knu68] mentioned that N.G. de Bruijn coined the word "***multiset***" to designate structures that were previously called "list, bunch, bag, heap, sample, weighted set, collection" in the literature. In [Yag86], the theory of bags and fuzzy bags is developed. In [Alb91], definitions and algebraic properties on bags are given, with the aim of having a convenient modeling data structure. In [Bli88], the concept of multiset is traced back to the very origin of numbers, when numbers were represented by repeating occurrences of symbols.

In [Bli88], the authors start by giving the following naive concept of multiset before building the theory MST of multisets:

"The naive concept of multiset that we now formalize has the following properties: (i) a multiset is a collection of elements in which certain elements may occur more than once; (ii) occurrences of a particular element in a multiset are indistinguishable; (iii) each occurrence of an element in a multiset contributes to the cardinality of the multiset; (iv) the number of occurrences of a particular element in a multiset is a (finite) positive integer; (v) the number of distinguishable (distinct) elements in a multiset needs to be finite; and (vi) a multiset is completely determined if we know which elements belong to it and the number of times each element belongs to it."

Nonetheless, the requirement (v) on the finitude is not necessary and, relying on [SIYS07], we will give the definitions on multisets without this constraint and this will hold in the remaining of this Thesis.

We consider a countable set A of distinct objects and a subset $\mathbb{W} \subseteq \mathbb{R}^+$. We consider m an application from A to $\mathbb{W}$. Then $\mathfrak{A}_m \triangleq (A, m)$[2] is called a **multiset**—or **mset** or **bag**— on A. A is called the **ground** or the **universe** of the multiset $\mathfrak{A}_m$, m is called the **multiplicity function** of the multiset $\mathfrak{A}_m$. $\mathfrak{A}^\star_m \triangleq \{x \in A \colon m(x) \neq 0\}$ is called the **support** of $\mathfrak{A}_m$. The elements of the support of a mset are called its **generators**. A multiset where $\mathbb{W} \subseteq \mathbb{N}$ is called a **natural multiset**.

We write $\mathfrak{M}(A)$ the set of all multisets of universe A. Some extensions of multisets exist where the multiplicity function can have its range in $\mathbb{Z}$—called hybrid set in [Loe92]. Some other extensions exist like fuzzy multisets ([Syr01]). Most definitions when not restricted to natural multisets can extend to $\mathbb{W} \subseteq \mathbb{R}$.

[2] We systematically use fraktur font to designate multisets or mathematical objects that involves multisets. For the reader that is not familiar to fraktur font, see Appendix F to have used letters and their correspondence in the text font.

Several notations of msets exist. One common notation which we will use here is: $\mathfrak{A}_m = \{x_i^{m_i} : i \in [\![n]\!] \wedge x_i \in A\}$ with $A = \{x_i : i \in [\![n]\!]\}$ the universe of the mset $\mathfrak{A}_m$ and the notation $\forall i \in [\![n]\!] : m_i = m(x_i)$. When the universe is made clear by the context, we can abusively write: $\mathfrak{A}_m = \{x_i^{m_i} : i \in [\![n]\!] \wedge x_i \in A \wedge m_i \neq 0\}$.

An other useful notation for a natural multiset is the one similar to an unordered list:

$$\mathfrak{A}_m = \left\{\!\!\left\{ \underbrace{x_1, \ldots, x_1}_{m_1 \text{ times}}, \ldots, \underbrace{x_n, \ldots, x_n}_{m_n \text{ times}} \right\}\!\!\right\}.$$

In this last representation, we do not write the elements that are not in the support, also the universe has to be clearly stated.

The **m-cardinality** $\#_m\mathfrak{A}_m$ of a mset $\mathfrak{A}_m$ corresponds to the sum of the multiplicities of the elements of its universe:

$$\#_m\mathfrak{A}_m \triangleq \sum_{x \in A} m(x)$$

while the **cardinality** $\#\mathfrak{A}_m$ of a mset $\mathfrak{A}_m$ is the number of elements of its support:

$$\#\mathfrak{A}_m = |\mathfrak{A}_m^\star|.$$

There exists only one multiset of universe A with an empty support that is called the **empty multiset** of universe A and written $\emptyset_A$.

Different operations can be defined on the set of all multisets of the same universe. We consider up to the end of this Section two msets $\mathfrak{A} = (U, m_\mathfrak{A})$ and $\mathfrak{B} = (U, m_\mathfrak{B})$ on the same universe U. We define different operations such as inclusion, equality, union, intersection, sum,...

The **inclusion** of $\mathfrak{A}$ in $\mathfrak{B}$—written $\mathfrak{A} \subseteq \mathfrak{B}$—holds if for all $x \in U$:

$$m_\mathfrak{A}(x) \leqslant m_\mathfrak{B}(x).$$

$\mathfrak{A}$ is then called a **submset** of $\mathfrak{B}$ and said to be **included** in $\mathfrak{B}$.

If $\mathfrak{A} \subseteq \mathfrak{B}$ and $\mathfrak{B} \subseteq \mathfrak{A}$, then $\mathfrak{A}$ and $\mathfrak{B}$ are said to be **equal**.

The **union** of $\mathfrak{A}$ and $\mathfrak{B}$ is the mset $\mathfrak{C} \triangleq \mathfrak{A} \cup \mathfrak{B}$ of universe U and of multiplicity function $m_\mathfrak{C}$ such that for all $x \in U$:

$$m_\mathfrak{C}(x) \triangleq \max\left(m_\mathfrak{A}(x), m_\mathfrak{B}(x)\right).$$

The **intersection** of $\mathfrak{A}$ and $\mathfrak{B}$ is the mset $\mathfrak{D} \triangleq \mathfrak{A} \cap \mathfrak{B}$ of universe U and of multiplicity function $m_\mathfrak{D}$ such that for all $x \in U$:

$$m_\mathfrak{D}(x) \triangleq \min\left(m_\mathfrak{A}(x), m_\mathfrak{B}(x)\right).$$

The **sum** of $\mathfrak{A}$ and $\mathfrak{B}$ is the mset $\mathfrak{S} \triangleq \mathfrak{A} \uplus \mathfrak{B}$ of universe U and of multiplicity function $m_\mathfrak{S}$ such that for all $x \in U$:

$$m_\mathfrak{S}(x) \triangleq m_\mathfrak{A}(x) + m_\mathfrak{B}(x).$$

It is worth reminding that $\cup$, $\cap$ and $\uplus$ are commutative and associative laws on msets of the same universe. The empty mset of same universe corresponds to the identity for these operations. $\uplus$ is distributive for $\cup$ and $\cap$. $\cup$ and $\cap$ are distributive one for the other. $\cup$ and $\cap$ are idempotent.

The **difference** of two msets is the mset $\mathfrak{T} \triangleq \mathfrak{A} - \mathfrak{B}$ of universe U and of multiplicity function $m_{\mathfrak{T}}$ such that for all $x \in U$:

$$m_{\mathfrak{T}}(x) \triangleq m_{\mathfrak{A}}(x) - m_{\mathfrak{A} \cap \mathfrak{B}}(x).$$

A set can be viewed as a natural mset where the multiplicities are in $\{1\}$, i.e. a natural multiset where the support corresponds to the universe and such that the multiplicity of the elements is always 1.

We can note that the classical property for sets: $(\mathfrak{A} - \mathfrak{B}) \cap \mathfrak{B} = \emptyset_U$ does not hold for multisets—see [SIYS07] for an example.

The **complementation** of $\mathfrak{A}$ referring to a family $\mathfrak{R}$ of multisets $(\mathfrak{A}_i)_{i \in I}$ of universe U is the multiset $\mathfrak{A}^c$ of universe U and of multiplicity function $m_{\mathfrak{A}^c}$ such that for all $x \in U$:

$$m_{\mathfrak{A}^c}(x) \triangleq \max_{i \in I} \left(m_{\mathfrak{A}_i}(x)\right) - m_{\mathfrak{A}}(x).$$

Finally, the **power set** of a multiset $\mathfrak{A}_m$—written $\mathcal{P}(\mathfrak{A}_m)$—is defined as the set of all submsets of $\mathfrak{A}_m$.

We define the **fusion** of two msets $\mathfrak{A}$ of universe U and $\mathfrak{B}$ of universe V as the multiset $\mathring{\mathfrak{F}} \triangleq \mathfrak{A} \oplus \mathfrak{B}$ of universe $U \cup V$ and of multiplicity function $m_{\mathring{\mathfrak{F}}}$ such that for all $x \in U \cup V$:

$$m_{\mathring{\mathfrak{F}}}(x) \triangleq \begin{cases} m_{\mathfrak{A}}(x) & \text{if } x \in U \backslash V \\ m_{\mathfrak{A}}(x) + m_{\mathfrak{B}}(x) & \text{if } x \in U \cap V \\ m_{\mathfrak{B}}(x) & \text{if } x \in V \backslash U. \end{cases}$$

An interesting alternative approach to define multisets is the one given in [Syr01] where a natural multiset $\mathfrak{A}_m = (A, m)$ is viewed as a couple $\langle A_0, \rho \rangle$, where A_0 is the set of instances of elements of A, that includes copies of elements, and ρ is an equivalency relation ρ over A_0 :

$$\forall x \in A_0, \forall x' \in A_0 : \ \ x \rho x' \Leftrightarrow \exists! c \in A : \ x \rho c \wedge x' \rho c.$$

Two elements of A_0 such that: $x \rho x'$ are said **copies** of one another. The unique $c \in A$ is called the original element. x and x' are said copies of c.

Also, A_0/ρ is isomorphic to A and:

$$\forall \overline{x} \in A_0/\rho, \exists! c \in A : |\{x : x \in \overline{x}\}| = m(c) \wedge \forall x \in \overline{x} : x \rho c.$$

The set A_0 is then called a **copy-set** of the multiset $\mathfrak{A}_m$.

We can remark that a copy-set for a given multiset is not unique. Sets of equivalency classes of two couples $< A_0, \rho >$ and $< A_0', \rho' >$ of a given multiset are isomorphic.

B.2.2. Algebraic representation of a natural multiset

We propose in this Section two algebraic representations of a natural multiset $\mathfrak{A}_m = (A, m)$ of countable universe $A = \{x_i : i \in [\![n]\!]\}$ and multiplicity function m.

Vector representation of a natural multiset

A multiset $\mathfrak{A}_m$ can be conveniently represented by a vector of length the cardinality of the universe and where its components represent the multiplicity of the corresponding element. We suppose that the elements of A are given in a fixed order—it is always possible to index these elements by a subset of the positive integer set. Hence $\overrightarrow{\mathfrak{A}_m} \triangleq (m(x_i))_{x_i \in A}$ is called a **vector representation of the multiset** $\mathfrak{A}_m$. This representation requires $|A|$ space and has $|A| - |\mathfrak{A}_m^\star|$ null elements.

The sum of the elements of $\overrightarrow{\mathfrak{A}_m}$ is $\#_m \mathfrak{A}_m$.

This representation will be useful later when considering family of multisets in order to build the incident matrix of a hb-graph.

An alternative representation is obtained using a symmetric hypermatrix. This approach is needed to reach our goal of constructing an e-adjacency tensor for general hb-graphs and hypergraphs.

Hypermatrix representations of a natural multiset

The **unnormalized hypermatrix representation of the natural multiset** $\mathfrak{A}_m = (A, m)$ is the symmetric hypermatrix $\boldsymbol{A_u} \triangleq (a_{u,i_1 \dots i_r})_{i_1,\dots,i_r \in [\![n]\!]}$ of order $r = \#_m \mathfrak{A}_m$ and dimension n such that $a_{u,i_1 \dots i_r} = 1$ if $\forall j \in [\![r]\!] : i_j \in [\![n]\!] \wedge x_{i_j} \in \mathfrak{A}_m^\star$. The other elements are null.

Hence the number of non-zero elements in $\boldsymbol{A_u}$ is $\dfrac{r!}{\prod\limits_{x \in \mathfrak{A}_m^\star} m(x)}$ out of the n^r elements of the representation.

The sum of the elements of $\boldsymbol{A_u}$ is then: $\dfrac{r!}{\prod\limits_{x \in \mathfrak{A}_m^\star} m(x)}$.

To normalize $\boldsymbol{A_u}$, we enforce the sum of the elements of the hypermatrix to be the m-rank of the multiset it encodes.

The **normalized hypermatrix representation of the multiset** $\mathfrak{A}_m$ is the symmetric hypermatrix $\boldsymbol{A} \triangleq (a_{i_1 \dots i_r})_{i_1,\dots,i_r \in [\![n]\!]}$ of order $r = \#_m \mathfrak{A}_m$ and dimension n such that $a_{i_1 \dots i_r} = \dfrac{\prod\limits_{x \in \mathfrak{A}_m^\star} m(x)}{(r-1)!}$ if $\forall j \in [\![r]\!] : i_j \in [\![n]\!] \wedge x_{i_j} \in \mathfrak{A}_m^\star$. The other elements are null.

B.2.3. Morphisms, category and multisets

In [Syr01], the author introduces two equivalent categories of multisets, called **MSet** and **Bags**: **MSet** reflects the multiplicity approach while **Bags** reflects the copy approach

and reserved to what is called pure multisets in [Syr03], i.e. what we have called natural multisets.

MSet objects are pairs (A, P) constituted of a set A and $P : A \to \mathbf{Set}$ a presheaf[3] on A, that describes a multiset, as $P(a)$ with $a \in A$ is a set of cardinality equals to the multiplicity of a in the multiset. An arrow between two objects of **MSet** is a pair constituted of a function and a natural transformation, i.e. a family of functions.

The definition of **Bags** is based on a function $p : X \to A$ where X represents the set of all copies of elements of A. The objects of **Bags** are pairs (A, p) and an arrow between two objects (A, p) and (B, q), with $p : X \to A$ and $q : Y \to B$ of **Bags** is a pair (f, g) such that $f : A \to B$ and $g : X \to Y$ and such that $q \circ g = f \circ p$.

In [Syr03], the author introduces a third category called **Mul** and linked it to a new definition of general multisets. A multiset is defined in this reference as a pair (A, ρ) where A is a set and ρ is an equivalence relation on A. A morphism of two multisets (A, ρ) and (A', ρ') is defined as a function $f : A \to A'$ such that for all x_1 and x_2 in A, if $x_1 \rho x_2$, then $f(x_1)\, \rho' f(x_2)$. In **Mul**, the objects correspond to the multisets and the arrows to the morphisms of the multisets. This last definition is the one retained in [IT15], where the author characterizes the different kind of morphisms encountered in **Mul**: it includes the monomorphisms that corresponds exactly to the injective morphisms, the epimorphisms corresponding to the surjective functions, and shows that any split monomorphism is a monomorphism and that any split epimorphism is an epimorphism. Moreover, the composition of arrows keeps monomorphisms, epimorphisms and bimorphisms. In **Mul**, every isomorphism is a bimorphism but the converse does not hold, which would have made **Mul** a balanced category.

B.2.4. Topologies on multisets

Defining topologies on multisets requires to handle the problem of the complementation on multisets. In [GJ12], the authors consider $[A]^{\infty}$ the set of all multisets of universe A and $[A]^{w}$ the set of all multisets where elements have a multiplicity in $[\![w]\!] \cup \{0\}$. All multisets are then considered in $[A]^{w}$ and the complementation of a multiset $\mathfrak{A} \in [A]^{w}$ is done using this family.

The sum of two multisets $\mathfrak{A}$ and $\mathfrak{B}$ in $[A]^{w}$ is redefined as the mset $\mathfrak{S}^{w} \triangleq \mathfrak{A} \uplus \mathfrak{B}$ of universe A and of multiplicity function $m_{\mathfrak{S}^{w}}$ such that for all $a \in A$:

$$m_{\mathfrak{S}^{w}}(a) \triangleq \min\{w, m_{\mathfrak{A}}(a) + m_{\mathfrak{B}}(a)\}.$$

The authors consider different kind of submsets of a mset $\mathfrak{A}$ in $[A]^{w}$: the whole submset where the elements of the support of the submset have same multiplicity than in $\mathfrak{A}$ and the full submset where the elements of the submset have a multiplicity less or equal to the one of $\mathfrak{A}$. For each kind corresponds a power mset, either whole—written $\mathrm{PW}(\mathfrak{A})$—or full—denoted $\mathrm{PF}(\mathfrak{A})$—, which are both sets of all whole or full submsets of a given mset. These two power msets are ordinary sets. The cardinality of $\mathrm{PW}(\mathfrak{A})$

[3] A **presheaf** F on a category $\mathcal{C}$ is a contravariant functor: $F : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$. When $\mathcal{C}$ is a discrete category, i.e. a category having only objects and no arrows, $\mathcal{C}^{\mathrm{op}} = \mathcal{C}$, which is the case for a set A considered as a category.

is 2^n where $n = \#\mathfrak{A}^\star$ and the cardinality of $\mathrm{PF}(\mathfrak{A})$ is $\prod_{a \in \mathfrak{A}^\star} m(a)$. In both cases, Cantor power set theorem—that stipulates for sets that the cardinality of a set—is strictly less than the cardinality of its power set fails in the case of msets. In order to preserve this theorem for multisets, the authors introduce a power mset $\mathfrak{P}(\mathfrak{A})$, that takes into account the repetitions of msets. They then consider different operation on a collection of msets drawn from $[A]^w$ including Cartesian product of two msets and a mset relation on a mset.

Finally, the authors introduce a multiset topology $\tau \subseteq \mathfrak{P}^\star(\mathfrak{A})$ of a multiset $\mathfrak{A} \in [A]^w$ with the classical axioms of a topology, called $\mathfrak{A}$-topology: these axioms includes the fact that the m-set union of any sub-collection of τ is in τ and the same for the mset intersection of any finite sub-collection, and the fact that $\mathfrak{A}$ and $\emptyset_A$ are in τ. Classical topology is then considered including closure, interior and limit point as well as continuous multiset functions.

The key point in this work has been to introduce the power mset of a multiset.

B.2.5. Applications of multisets

This section is partly based on [OLGMM19a].

Multisets, under the appellation bag, appear in different domains such as text modeling, image description and audio [SJP+16]. In text representation, bag of words have been first introduced in [Har54]: bags are lists of words with repetitions, i.e. multisets of words on a universe. Many applications occur with different approaches. Bags of words have been used for instance in fraud detection [PS15]. More recently, bag of words have been used successfully for translation by neural nets as a target for the translation, since a sentence can be translated in many different ways [MSWL18]. In [CAO+18], multi-modal bag of words have been used for cross domains sentiment analysis.

Bags of visual words is the transcription to image of textual bags of words; in bags of visual words, a visual vocabulary based on image features is built to enabling the description of images as bags of these features. Since their introduction in [SZ03], many applications have been realized: in visual categorization [CDF+04], in image classification and filtering [DPN08], in image annotation [Tsa12], in action recognition [PWWQ16], in land-use scene classification [ZTH14], in identifying mild traumatic brain injuries [MWW+17] and in word image retrieval [SJ12].

Bags of concepts are an extension of bags of words to successive concepts in a text [KKC17]. A recent extension of these concepts is given in [SWG+18] where bag of graphs are introduced to encode in graphs the local structure of a digital object: bags of graphs are available as bags of singleton graphs and bags of visual graphs. Using the hb-graphs as we propose in this Thesis will allow to extend this approach, by taking advantage of multi-adic relationships, and, also, of the multiplicity of vertices specific to each hb-edge.

Additional applications to mathematics can be found in [SIYS07]: it includes the integer prime factorization, the representation of monic polynomials by the mset of its roots and correspondence between polynomial operations and mset operations, the zeros and poles of meromorphic functions. Other non-straightforward applications in mathematics are

considered such as the semi-ring of multisets developed in [Po00] making connections between eigenvalues and eigenvectors of max-plus algebra[4] and the non-negative real number systems.

Different approaches exist in computing, either theoretical or practical, that uses multisets. For instance, in [BLM93] the authors present a formalism where programs are presented in terms of multiset transformations, giving birth to the GAMMA programs, the first version of chemical programming[5]; other approaches exist such as the one using generalised multisets in [BFR06] or higher-order chemical language in [BFR07]. Multisets are used in membrane computing, a part of computer science that seeks to find computational models that mimic biological systems as introduced by Păun[6] in 1998, to describe objects in super-cells in [DP99].

In [PK08], the authors use decision procedures for multisets using cardinality constraints. Collection of multisets are found in software verification and interactive theorem proving. They emphasize the useful operations and relations on multisets, particularly formulas on the number of occurrences of each element in a multiset, using quantifier-free Presburger arithmetic (QFPA for short) which is a first order decidable theory that uses an arithmetic containing only addition and recurrence statement. They propose an algorithm for reducing multiset formulas to sum normal form, introducing constraints on multisets; the satisfiability is achieved using an extension of QFPA to unbounded expressions.

More recently in [BA16], the authors introduce multiset operations for privacy-preserving set operations among multiple players, such as computing database intersection of different organizations.

B.3. Tensors and hypermatrices

B.3.1. An introduction on tensors

A rigorous introduction of tensors is achieved in [Lee13]: we rely on it in this introduction.

First, we introduce the notion of co-vector.

We consider a finite dimension vector space V over $\mathbb{R}$. A **co-vector** is a linear form i.e. an application $\omega : V \to \mathbb{R}$ which is linear[7]. The set of all co-vectors of V constitutes the **dual space** $V^\star$ of V.

For a given basis $(e_i)_{i\in[\![n]\!]}$ of V, the co-vectors $\left(\epsilon^j\right)_{j\in[\![n]\!]}$ of $V^\star$ are defined by:

$$\epsilon^j\left(e_i\right) \triangleq \delta_{ij},$$

where: $\delta_{ij} \triangleq \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$ is the Kronecker delta symbol. The co-vectors $\left(\epsilon^j\right)_{j\in[\![n]\!]}$ constitutes a basis of $V^\star$, called the **dual basis** of $(e_i)_{i\in[\![n]\!]}$.

[4] Cf for instance [ABG06] for a definition of max-plus algebra

[5] For a cool introduction, see: http://gigasquidsoftware.com/chemical-computing/index.html

[6] Cf [Pău11] for a reference book on membrane computing

[7] An application $f : V \to W$, where V and W are two vector spaces, is linear if for all $v, v' \in V$ and λ, μ, $f\left(\lambda v + \mu v'\right) = \lambda f\left(v\right) + \mu f\left(v'\right)$.

We define the **tensor product of two co-vectors** ω and ω' of $V^\star$ as the map:

$$\omega \otimes \omega' : V \times V \to \mathbb{R},$$

such that for $(v_1, v_2) \in V \times V$:

$$\omega \otimes \omega' (v_1, v_2) \triangleq \omega (v_1) \cdot \omega' (v_2) .$$

The tensor product of two co-vectors can be generalized to any two multilinear forms.

Considering k vector spaces $(V_j)_{j \in [\![k]\!]}$, a **multilinear form** is an application $F : V_1 \times ... \times V_k \to \mathbb{R}$ which is linear in each of its variables. We write $\mathcal{L}(V_1, ..., V_k; \mathbb{R})$ the space of the multilinear forms on the vector space $V_1 \times ... \times V_k$.

The **tensor product of two multilinear forms** $F : V_1 \times ... \times V_k \to \mathbb{R}$ and $G : W_1 \times ... \times W_l \to \mathbb{R}$ is defined as the map: $F \otimes G : V_1 \times ... \times V_k \times W_1 \times ... \times W_l \to \mathbb{R}$ such that for all $(v_1, ..., v_k) \in V_1 \times ... \times V_k$ and $(w_1, ..., w_l) \in W_1 \times ... \times W_l$, it holds:

$$F \otimes G (v_1, ..., v_k, w_1, ..., w_l) \triangleq F (v_1, ..., v_k) \cdot G (w_1, ..., w_l) .$$

$F \otimes G$ is a multilinear form of $k + l$ variables and lives in $\mathcal{L}(V_1, ..., V_k, W_1, ..., W_l; \mathbb{R})$.

Hence, considering k vector spaces $(V_j)_{j \in [\![k]\!]}$ and k linear forms $\omega^j \in V_j^\star$, with $j \in [\![k]\!]$, $\bigotimes_{j \in [\![k]\!]} \omega_j$ is a multilinear form of $\mathcal{L}(V_1, ..., V_k; \mathbb{R})$ and for $(v_1, ..., v_k) \in V_1 \times ... \times V_k$:

$$\bigotimes_{j \in [\![k]\!]} \omega^j (v_1, ..., v_k) \triangleq \prod_{j \in [\![k]\!]} \omega^j (v_j) .$$

It follows that considering a basis $\left(e_{(j)\, i_j}\right)_{i_j \in [\![n_j]\!]}$ for each of the vector space V_j of dimension n_j with $j \in [\![k]\!]$ and, considering the corresponding dual spaces $\left(\epsilon_{(j)}^{i_j}\right)_{i_j \in [\![n_j]\!]}$, it holds that:

$$\left(\bigotimes_{j \in [\![k]\!]} \epsilon_{(j)}^{i_j} \right)_{i_1 \in [\![n_1]\!], ..., i_k \in [\![n_k]\!]}$$

is a basis of $\mathcal{L}(V_1, ..., V_k; \mathbb{R})$.

This means that any multilinear form $F \in \mathcal{L}(V_1, ..., V_k; \mathbb{R})$ can be decomposed in:

$$F = \sum_{i_1 \in [\![n_1]\!], ..., i_k \in [\![n_k]\!]} F_{i_1 ... i_k} \bigotimes_{j \in [\![k]\!]} \epsilon_{(j)}^{i_j}.$$

With the same idea, and without developing all the arguments that can be found in [Lee13], the same approach can be taken to build a new vector space $V_1 \otimes ... \otimes V_k$ in which elements are of the type $v_1 \otimes ... \otimes v_k$ such that they are linear in each of the v_i, $i \in [\![k]\!]$. It is shown that $V_1^\star \otimes ... \otimes V_k^\star$ is isomorphic to $\mathcal{L}(V_1, ..., V_k; \mathbb{R})$ and that by duality: $V_1 \otimes ... \otimes V_k$ is isomorphic to $\mathcal{L}(V_1^\star, ..., V_k^\star; \mathbb{R})$.

With $V_1 = ... = V_k = V$ a finite dimensional space, a **covariant cubical tensor on V of order (or rank)** k is an element of $\underbrace{V^\star \otimes ... \otimes V^\star}_{k \text{times}}$ i.e. a multilinear form of k

elements of V. We write: $\mathcal{L}_k^0(V)$ the space of the covariant cubical tensor on V of rank k. An element of $V \otimes ... \otimes V$ is called a **contravariant cubical tensor on V of rank k**. We write: $\mathcal{L}_0^k(V)$ the space of the contravariant cubical tensor on V of rank k. More generally an element of $\underbrace{V \otimes ... \otimes V}_{k\text{times}} \otimes \underbrace{V^\star \otimes ... \otimes V^\star}_{l\text{times}}$ is called a mixed tensor of type (k,l). The corresponding space is $\mathcal{L}_k^l(V)$.

Only elements of $\mathcal{L}_k^0(V)$ admits the restricted approach that is presented in the next section.

B.3.2. Restricted approach of tensors as hypermatrices

As we are interested only in cubical tensors[8] that are abusively identified to hypermatrices, we give a straightforward presentation of tensors and their relative hypermatrices.

We write $\mathcal{L}_k^0(\mathbb{R}^n)$ the space of cubical tensors of rank k and dimension n with values in $\mathbb{R}$, and $\mathcal{A}_k$ one of its elements.

The **Segre outerproduct** $\otimes$ of $\boldsymbol{a} = [a_i] \in \mathbb{R}^l$ and $\boldsymbol{b} = [b_j] \in \mathbb{R}^m$ is defined as:

$$\boldsymbol{a} \otimes \boldsymbol{b} = [a_i b_j]_{\substack{i \in [\![l]\!] \\ j \in [\![m]\!]}} \in \mathbb{R}^{l \times m}.$$

More generally, as given in [CGLM08], the **outerproduct** of k vectors $u_{(i)} \in \mathbb{R}^{n_i}$, with $i \in [\![k]\!]$ and where $\forall i \in [\![k]\!]$, $n_i \in \mathbb{N}^*$, is defined as:

$$\bigotimes_{i=1}^{k} \boldsymbol{u_{(i)}} \triangleq \left[\prod_{i=1}^{k} u_{(i)j_i} \right]_{j_1 \in [\![n_1]\!], .., j_k \in [\![n_k]\!]} \in \mathbb{R}^{n_1 \times ... \times n_k}.$$

Here, we consider $n_i = n$ for all $i \in [k]$. Writing $\boldsymbol{e_1}$, ..., $\boldsymbol{e_n}$ the canonical basis of $\mathbb{R}^n$, $(\boldsymbol{e_{i_1}} \otimes ... \otimes \boldsymbol{e_{i_k}})_{i_1,...,i_k \in [\![n]\!]}$ is a basis of $\mathcal{L}_k^0(\mathbb{R}^n)$.

Then the tensor $\mathcal{A}_k \in \mathcal{L}_k^0(\mathbb{R}^n)$ is associated to a hypermatrix $\boldsymbol{A_k} = \left(a_{(k)\, i_1 ... i_k}\right)_{i_1,...,i_k \in [\![n]\!]}$ by writing $\mathcal{A}_k$ as:

$$\mathcal{A}_k = \sum_{i_1,...,i_k \in [\![n]\!]} a_{(k)\, i_1 ... i_k} \boldsymbol{e_{i_1}} \otimes ... \otimes \boldsymbol{e_{i_k}}.$$

$\boldsymbol{A_k}$ is called the **canonical hypermatrix representation** (CHR for short) of $\mathcal{A}_k$.

In [Lim13], a **multilinear matrix multiplication** is defined considering $\boldsymbol{A} \in \mathbb{R}^{n_1 \times ... \times n_d}$ and $X_j = \left[x_{(j)kl}\right] \in \mathbb{R}^{m_j \times n_j}$ for $j \in [\![d]\!]$ as:

$$\boldsymbol{A'} \triangleq (X_1, ..., X_d).\boldsymbol{A}$$

where $\boldsymbol{A'}$ is the hypermatrix of $\mathbb{K}^{m_1 \times ... \times m_d}$ of coefficients:

$$a'_{j_1 ... j_d} \triangleq \sum_{k_1 \in [\![n_1]\!], ..., k_d \in [\![n_d]\!]} x_{(1)j_1k_1} ... x_{(d)j_dk_d} a_{k_1 ... k_d}$$

for $j_i \in [\![m_i]\!]$ with $i \in [\![d]\!]$.

[8] We often abusively use the word tensor for its hypermatrix representation in a canonical basis in the verbatim of this Thesis, following the abuse made in [QL17]. Nonetheless, when writing it mathematically we make the difference between the tensor $\mathcal{A}$ of dimension n and rank r and its canonical hypermatrix representation (CHR for short), written $\boldsymbol{A} = (a_{i_1 ... i_r})$ as defined in this Section.

B.3.3. On classification of hypermatrices

This Section is extracted from [OLGMM19c].

Most of the definitions and results of this Section are taken directly from [QL17], which is the first book on tensor spectral analysis.

We consider a tensor $\mathcal{A}$ and its CHR of order m and dimension n:

$$\boldsymbol{A} = (a_{i_1 \dots i_m})_{i_1,\dots,i_m \in [\![n]\!]}.$$

The set of such hypermatrices is written $T_{m,n}$. The subset of $T_{m,n}$ of hypermatrices with only non-negative[9] coefficients is written $N_{m,n}$.

A hypermatrix $\boldsymbol{A} = (a_{i_1 \dots i_m})_{i_1,\dots,i_m \in [\![n]\!]} \in T_{m,n}$ is said **reducible** if there exists a nonempty proper subset $J \subsetneq [\![n]\!]$ such that:

$$\forall i_1 \in J, \forall (i_2, ..., i_m) \in ([\![n]\!] \setminus J)^{m-1} : a_{i_1 \dots i_m} = 0.$$

A hypermatrix that is not reducible is said **irreducible**. The notion of reducibility has to be seen as a possible way of reducing the dimensionality of the problem.

Irreducible non-negative hypermatrices have plenty of nice properties, particularly the Perron-Frobenius theorem for irreducible non-negative hypermatrices which is one of the two declinations of the extension of the Perron-Frobenius theorem for irreducible non-negative matrices.

The Perron-Frobenius theorem for irreducible non-negative matrices states that the eigenvector associated to the spectral radius of a non-negative matrix is non-negative, and, if, moreover, this matrix is irreducible, then this eigenvector is positive[10] and its eigenvalue is the unique one associated with a non-negative eigenvector.

The Perron-Frobenius theorem has a lot of applications in probability with stochastic matrices and is the basis for algorithms such as PageRank, ensuring that the convergence is feasible [PSS05].

But hypermatrices manipulated in hypergraph theory are reducible. In this case, this first extension of the theorem of Perron-Frobenius cannot be used.

Weak irreducibility of non-negative hypermatrices has been introduced to help to solve this problem. To define weak irreducibility, an **associated graph** $\mathcal{G}(\mathcal{A})$ is built out of the non-negative hyper-cubic CHR $\boldsymbol{A}$ which represents $\mathcal{A}$ by considering as vertex set $[\![n]\!]$ and building the edges by considering an edge from i to j if there exists $a_{ii_2 \dots i_m} \neq 0$ such that $j \in \{i_2, ..., i_m\}$.

A directed graph is **strongly connected** if for any ordered pair of vertices (i, j) of the graph, there exists a directed path from i to j.

A tensor $\mathcal{A}$ is said **weakly irreducible** if its associated graph is strongly connected. A tensor that is not weakly irreducible is said **weakly reducible**.

We consider now a tensor $\mathcal{A}$ of non-negative CHR $\boldsymbol{A} = (a_{i_1 \dots i_m}) \in N_{m,n}$.

[9] I.e. either positive or equals to zero.
[10] I.e. all the coefficients are strictly greater than zero.

The **representative vector of** $\mathcal{A}$ is the vector $u(\mathcal{A})$ of coordinates:

$$u_i(\mathcal{A}) \triangleq \sum_{i_2,\dots,i_m \in [\![n]\!]} a_{ii_2\dots i_n}.$$

$\mathcal{A}$ is called a **strictly non-negative tensor** if its representative vector is positive.

Let now J be a proper nonempty subset of $[\![n]\!]$. The tensor $\mathcal{A}_J$ of CHR $\boldsymbol{A}_J \triangleq (\alpha_{i_1\dots i_m}) \in N_{m,|J|}$ such that: $\alpha_{i_1\dots i_m} \triangleq a_{i_1\dots i_m}$ if $i_1, \dots, i_m \in J$ is called the **principal sub-tensor of** $\mathcal{A}$ **associated to** J.

$\mathcal{A}$ is said a **non-trivially non-negative tensor** if there exists a principal sub-tensor of $\mathcal{A}$ that is a strictly non-negative tensor.

The following proposition will be helpful in the spectral analysis of hb-graphs.

Proposition B.1. *A non-negative tensor has a positive eigenvalue if and only if it is non-trivially non-negative.*

As a consequence, a non-trivially non-negative tensor has its spectral radius positive.

In [QL17], a procedure is given to check easily if a non-negative tensor is non-trivially negative or not.

B.3.4. Eigenvalues

The definitions and results of this Section are based on [QL17]. Proofs can be found in this reference.

Let $T_{m,n}$ be the set of all real hypermatrices of order m and dimension n and $S_{m,n}$ the subset of $T_{m,n}$ of symmetric hypermatrices, i.e. hypermatrices such that are invariant by any permutations of the indices.

Let $\boldsymbol{A} = (a_{i_1\dots i_m}) \in T_{m,n}$. Let $\boldsymbol{I} \in T_{m,n}$ designates the identity tensor.

Definition B.1. *A number $\lambda \in \mathbb{C}$ is an **eigenvalue** of $\boldsymbol{A}$ if there exists a nonzero vector $x \in \mathbb{C}^n$ such that:*

$$\forall i \in [\![1,n]\!], \left(\boldsymbol{A}x^{m-1}\right)_i = \lambda x_i^{m-1}. \tag{B.1}$$

*In this case, x is called an **eigenvector** of $\boldsymbol{A}$ associated with the eigenvalue λ and (x,λ) is called an **eigenpair** of $\boldsymbol{A}$.*

*The set of all eigenvalues of $\boldsymbol{A}$ is called the **spectrum** of $\boldsymbol{A}$. The largest modulus of all eigenvalues is called the **spectra radius of** $\boldsymbol{A}$, denoted as $\rho(\boldsymbol{A})$.*

Remark B.3.1. *Writing $x^{[m-1]} = \left(x_i^{m-1}\right)_{i\in[\![n]\!]}$, B.1 can be transformed into:*

$$\boldsymbol{A}x^{m-1} = \lambda x^{[m-1]}.$$

Proposition B.2. *Let α and β be two real numbers.*

If (λ, x) is an eigenpair of $\boldsymbol{A}$, then $(\alpha\lambda + \beta, x)$ is an eigenpair of $\alpha\boldsymbol{A} + \beta\boldsymbol{I}$.

Definition B.2. *A **H-eigenvalue** is an eigenvalue λ of $\boldsymbol{A}$ that has a real eigenvector x associated to it. x is called in this case an **H-eigenvector**.*

Proposition B.3. *A H-eigenvalue is real.*

A real eigenvalue is not necessarily a H-eigenvalue.

The following theorem holds for symmetric tensors:

Theorem B.1. *Let $\boldsymbol{A} \in S_{m,n}$. If m is even, then $\boldsymbol{A}$ always have H-eigenvalues and, $\boldsymbol{A}$ is positive definite (resp. semi-definite) if and only if its smallest H-eigenvalue $\lambda_{H_{\min}}(\boldsymbol{A})$ is positive (resp. non-negative).*

Theorem B.2. *Let $\boldsymbol{A} \in T_{m,n}$ be a non-negative tensor. Then $\boldsymbol{A}$ has at least one H-eigenvalue and $\lambda_{H_{\max}}(\boldsymbol{A}) = \rho(\boldsymbol{A})$. Furthermore $\lambda_{H_{\max}}(\boldsymbol{A})$ has a non-negative H-eigenvector.*

Definition B.3. *A tensor is called an **essentially non-negative tensor** if all its off-diagonal entries are non-negative.*

*A tensor is called a **Z-tensor** if its off-diagonal entries are non-positive.*

Theorem B.3. *Essentially non-negative tensors and Z-tensors always have H-eigenvalues.*

Definition B.4. *Let $\boldsymbol{A} = (a_{i_1 \ldots i_m}) \in T_{m,n}$.*

The diagonal elements of $\boldsymbol{A}$ are the elements $a_{i\ldots i}$ for $i \in [\![n]\!]$.

The off-diagonal elements of $\boldsymbol{A}$ are the other elements.

An important result is the following:

Proposition B.4. *Let $\boldsymbol{A} \in T_{m,n}$. Then the eigenvalues of $\boldsymbol{A}$ belong to the union of n disks in $\mathbb{C}$. These n disks have the diagonal entries of $\boldsymbol{A}$ as their centers and the sums of the absolute values of the off-diagonal entries as their radii.*

Remark B.3.2. *The proof shows that if (λ, x) is an eigenpair of $\boldsymbol{A} = (a_{i_1 \ldots i_m})$, it holds for $i \in [\![m]\!]$ such that: $|x_i| = \max\{|x_j| : j \in [\![n]\!]\}$:*

$$|\lambda - a_{i\ldots i}| \leqslant \sum_{\substack{i_2,\ldots,i_m \in [\![n]\!] \\ \delta_{ii_2\ldots i_m} = 0}} |a_{ii_2\ldots i_m}|. \tag{B.2}$$

Corollary B.1. *If $\boldsymbol{A}$ is a non-negative tensor of $T_{m,n}$ with an equal row sum r, then r is the spectral radius of $\boldsymbol{A}$.*

Other types of eigenvalues also exist. It is worth mentioning E-eigenvalues and Z-eigenvalues since they are part of the related work of this Thesis.

Given a tensor $\boldsymbol{A} = (a_{i_1 \dots i_m})$ of $T_{m,n}$, $\lambda \in \mathbb{C}$ is an **E-eigenvalue** of $\boldsymbol{A}$ if there exists a vector $x \in \mathbb{C}^n$, called an **E-eigenvector**, such that: $Ax^{m-1} = \lambda x$ with: $x^\top x = 1$.

A **Z-eigenvalue** is an E-eigenvalue associated to a real E-eigenvector. A Z-eigenvalue is always real—but a real E-eigenvalue is not necessary a Z-eigenvalue.

Any symmetric tensor has Z-eigenvalues.

C. Complements Chapter 1: An introduction to hb-graphs

C.1. An example using hypergraphs for modeling co-occurrence networks and showing their limitations

Considering the publication co-occurrence network example, including three facets—organization, publication itself and keywords—as shown in Figure .

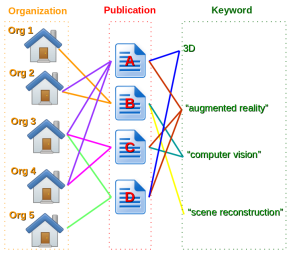


Figure C.1.: Example of a publication co-occurrence network with three facets

If co-occurrences of organizations are retrieved choosing as reference the publications ids, the co-occurrences shown in Table C.1 are obtained. In this case, a family of subsets of the vertex set of organizations is obtained; each of this subset corresponds to a hyperedge of the hypergraph modeling the co-occurrences.

Pub A	Org 2, Org3, Org 4
Pub B	Org 1, Org 2
Pub C	Org 3, Org 4
Pub D	Org2, Org 3, Org 5

Table C.1.: Co-occurrences of organizations built using Publication as reference

If now, co-occurrences of organizations are built using as reference the keywords, the co-occurrences retrieved include multiplicities that are depending on the co-occurrence. In this case, we obtain a family of multisets of organizations of same universe, which is the set of organizations the dataset holds, as shown in Table C.2. Hypergraphs in this case are not able to handle it.

scene reconstruction	$\{\{\text{Org } 1^{1}, \text{Org } 2^{1}\}\}$
computer vision	$\{\{\text{Org } 1^{1}, \text{Org } 2^{1}, \text{Org } 3^{1}, \text{Org } 4^{1}\}\}$
augmented reality	$\{\{\text{Org } 2^{1}, \text{Org } 3^{3}, \text{Org } 4^{2}, \text{Org } 5^{1}\}\}$
3D	$\{\{\text{Org } 2^{1}, \text{Org } 3^{2}, \text{Org } 4^{1}, \text{Org } 5^{1}\}\}$

Table C.2.: Co-occurrences of organizations built using Keywords as reference

C.2. Homomorphisms of natural hb-graphs

In this appendix, we specify homomorphisms of natural hb-graphs and define the category of hb-graphs. We extend in this Section the results given for hypergraphs in [DW80]—as exposed in [Ouv20]—to hb-graphs.

Let the universe multiset functor: $P : \mathbf{Set} \Rightarrow \mathbf{Mul}$ be defined by:

- for its object part, it associates to an object V of **Set** an object $P(V)$ which is the collection (class) of all multisets on V i.e. the elements of $\mathfrak{M}(V)$.
- for its arrow part, by: for $h \in \hom_{\text{Set}}(V_1, V_2)$: $P(h)$ is the morphism $P(h)$ of $\hom_{\text{Mul}}(P(V_1), P(V_2))$ such that for $M_1 \in P(V_1)$, with $M_1 = \phi((X_1, \rho_1))$, $P(h)(M_1)$ with $(X_2, \rho_2) = \phi^{-1}(P(h)(M_1))$ is such that:

$$\forall x, x' \in X_1 : x \rho_1 x' \Rightarrow y \rho_2 y'$$

  where y and y' are the elements of X_2 corresponding to copies of elements of $P(h)(M_1) = M_2$.

We consider **HbGraph** composed of a class of objects constituted of all hb-graphs. A hb-graph $\mathfrak{H}$ is given in this class by using a vertex set V, a hb-edge list $\mathfrak{E}$ and a mapping function $\iota : \mathfrak{E} \to \mathfrak{M}(V)$ where $\mathfrak{M}(V)$ is the set of all multisets of universe V which is obtained as the image of $V \in \mathbf{Set}$ through the functor P. We have for each $e \in \mathfrak{E} : \iota(e)$ corresponds to a multiset belonging to $\mathfrak{M}(V)$.

We consider two hb-graphs: $\mathfrak{H}_1 = (V_1, \mathfrak{E}_1, \iota_1)$ and $\mathfrak{H}_2 = (V_2, \mathfrak{E}_2, \iota_2)$.

A hb-graph homomorphism $h : \mathfrak{H}_1 \to \mathfrak{H}_2$ is defined as a pair of functions $h = (h_V, h_E)$ with $h_V : \mathfrak{M}(V_1) \to \mathfrak{M}(V_2)$ and $h_E : \mathfrak{E}_1 \to \mathfrak{E}_2$ such that the diagram:

$$\begin{array}{ccc} \mathfrak{E}_1 & \xrightarrow{\iota_1} & \mathfrak{M}(V_1) \setminus \{\emptyset\} \\ \downarrow{\scriptstyle h_E} & & \downarrow{\scriptstyle P(h_V)} \\ \mathfrak{E}_2 & \xrightarrow{\iota_2} & \mathfrak{M}(V_2) \setminus \{\emptyset\} \end{array}$$

commutes.

Let $h_1 \in \text{Mor}_{\mathbf{HbGraph}}(\mathfrak{H}_1, \mathfrak{H}_2)$ and $h_2 \in \text{Mor}_{\mathbf{HbGraph}}(\mathfrak{H}_2, \mathfrak{H}_3)$ be two hb-graph homomorphisms. The composition of these two hb-graphs, written $h = h_2 \circ h_1$, is the hb-graph

homomorphism: $h = (h_V, h_E)$ such that: $h_V = h_{V,2} \circ h_{V,1}$ and $h_E = h_{E,2} \circ h_{E,1}$. This composition is associative.

For every object $\mathfrak{H}$, the identity hb-graph morphism $1_{\mathfrak{H}}$ keeps the hb-edges identical and keeps each hb-edge in the same state.

Hence, considering objects as hb-graphs and arrows as hb-graph homomorphisms, **HbGraph** is a category.

The following lemmata occur:

Lemma C.1. ***Hyp*** *is a full sub-category of* ***HbGraph****.*

Proof. We consider the inclusion functor: $J : \mathbf{Hyp} \Rightarrow \mathbf{HbGraph}$ defined as:

- for the object part: $\mathcal{H} \to J(\mathcal{H})$ where $J(\mathcal{H})$ is the hb-graph of universe the vertex set of $\mathcal{H}$ and where the hb-edges are linked to multisets with multiplicity no more than one;
- for the arrow part: at each hypergraph homomorphism corresponds a unique hb-graph homomorphism that behaves similarly on vertices and hb-edges than the one on hypergraph.

□

Lemma C.2. *Duality of hb-graphs is a covariant functor* $d : \boldsymbol{HbGraph} \Rightarrow \boldsymbol{HbGraph}$

Proof. The object part of the functor associates to a hb-graph its dual: for $\mathfrak{H} = (V, \mathfrak{E}, \iota) \in \mathbf{HbGraph}$, $d(\mathfrak{H}) = \tilde{\mathfrak{H}}$ with $\tilde{\mathfrak{H}} = (\mathfrak{E}, \mathfrak{V}, \tilde{\iota})$ where: $\tilde{\iota} : V \to \mathfrak{M}(\mathfrak{E})$ such that: $\tilde{\iota}(v)$ corresponds to the multiset described by: $\left\{\left\{\mathfrak{e}^{m_{\mathfrak{e}}(v)} : v \in \iota(\mathfrak{e})\right\}\right\}$.

The arrow part of the functor associates to one hb-graph homomorphism $h \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_1, \mathfrak{H}_2)$ the hb-graph homomorphism corresponding to the duals: $d(h) : d(\mathfrak{H}_1) \to d(\mathfrak{H}_2)$ such that $d(h)(x) = h(x)$ for $x \in V \cup E$.

If we take $\mathfrak{H} = \mathfrak{H}_1 = \mathfrak{H}_2$ and $h = \mathrm{Id}_{\mathfrak{H}}$, we have $d(\mathrm{Id}_{\mathfrak{H}}) = \mathrm{Id}_{d(\mathfrak{H})}$.

Taking $h_1 \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_1, \mathfrak{H}_2)$ and $h_2 \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_2, \mathfrak{H}_3)$, we have:

$$d(h_1 \circ h_2) = h_1 \circ h_2 = d(h_1) \circ d(h_2).$$

□

We write **TGraph** the category of all graphs as defined in [BMSW08].

Lemma C.3. *The incidence multipartite graph of a hb-graph induces a covariant functor* $M : \boldsymbol{HbGraph} \to \boldsymbol{TGraph}$*.*

Proof. We have explained in Section 1.4.6.2 how we can associate a unique extra-vertex multipartite graph $\mathcal{G}_{\mathfrak{H}}$ to a hb-graph $\mathfrak{H}$.

We consider for M :

- for the object part: $M : \mathfrak{H} \to M(\mathfrak{H})$ where $M(\mathfrak{H}) = \mathcal{G}_{\mathfrak{H}}$.

- for the arrow part: given $h \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_1, \mathfrak{H}_2)$, we consider $M(h) : \mathcal{G}_{\mathfrak{H}_1} \to \mathcal{G}_{\mathfrak{H}_2}$ such that: $v \to h(v)$ and $e \to h(e)$ such that any copies of a vertex in a hb-edge has its image that is the copy of the image of the copies in the hb-edge image.

If we take $\mathfrak{H} = \mathfrak{H}_1 = \mathfrak{H}_2$ and $h = \mathrm{Id}_{\mathfrak{H}}$, copies of vertices will remain the same by $M(\mathrm{Id}_{\mathfrak{H}})$ as well as hb-edges and $M(\mathrm{Id}_{\mathfrak{H}}) = \mathrm{Id}_{M(\mathfrak{H})}$.

Moreover, taking $h_1 \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_1, \mathfrak{H}_2)$ and $h_2 \in \mathrm{Hom}_{\mathbf{HbGraph}}(\mathfrak{H}_2, \mathfrak{H}_3)$, we have:

$$M(h_2 \circ h_1) = M(h_2) \circ M(h_1).$$

□

We conjecture that general hb-graphs constitute a new category, but this remains an open question.

D. Complements Chapter 2: Diffusion in hb-graphs

D.1. Facets and Biased diffusion

This chapter is extending the approach presented in Chapter 2 and in [OLGMM19a]. It has been published after this Thesis Defense in [OGMM20].

Prerequisites: Section 1.3 and, preferably Chapter 2.

D.1.1. Motivation

Facets are not all of same nature in an information space; depending on the information they carry, the ranking of the information hold by their visualisation hb-graph has to be performed on different features, and the importance stressed on the lower, higher or medium values depending on the application. Hence, the necessity of extending to a more general approach the exchange-based diffusion that is already coupled to a biased random walk given in [OLGMM19a] and presented in Chapter 2.

In this Section, we want to tackle the following Research Question:

Research question D.1. *How to extend the exchange-based diffusion on hb-graphs to support a general biased-feature-exchange-based diffusion?*

We start by reviewing what has been achieved so far on biased diffusion, and propose a framework to achieve such a kind of diffusion.

D.1.2. Related work

In [DM11], the authors introduce an abstract information function $f : V \to \mathbb{R}^+$ which is associated to a probability for each vertex $v_i \in V$:

$$p^f(v_i) = \frac{f(v_i)}{\sum_{j \in [\![|V|]\!]} f(v_j)}.$$

In [ZGC10], a bias is introduced in the transition probability of a random walk in order to explore communities in a network. The bias is either related to a vertex property x_i, such as the degree, or to an edge property y_j, such as the edge multiplicity or the shortest path betweenness. For a vertex, the new transition probability between vertex v_i and v_j is given by:

$$T_{ij}(x, \beta) = \frac{a_{ij} e^{\beta x_i}}{\sum_l a_{lj} e^{\beta x_l}},$$

where $A = (a_{ij})_{i,j\in[\![n]\!]}$ is the adjacency matrix of the graph and β is a parameter.

A same kind of bias, can be used related to the edges and can be combined to the former to have the overall transition probability from one vertex to an other.

D.1.3. Biased diffusion in hb-graphs

We consider a weighted hb-graph $\mathfrak{H} = (V, \mathfrak{E}, w_e)$ with $|V| = n$ and $|\mathfrak{E}| = p$; we write H the incidence matrix of the hb-graph.

D.1.3.1. Abstract information functions and bias

We consider a hb-edge **based vertex abstract information function:**

$$f_V : V \times \mathfrak{E} \to \mathbb{R}^+.$$

The exchange-diffusion that we have presented in Chapter 2 is a particular example of biased diffusion, where the biases are given in Table D.1. An unbiased diffusion would be to have a vertex abstract function and a hb-edge vertex function that is put to 1 for every vertices and hb-edges, i.e. equi-probability for every vertices and every hb-edges.

Hb-edge based vertex abstract information function	$f_V(v_i, \mathfrak{e}_j) = m_j(v_i)\, w(\mathfrak{e}_j)$
Vertex abstract information function	$F_V(v_i) = d_{w,v_i}$
Vertex bias function	$g_V(x) = x$
Vertex overall bias	$G_V(v_i) = d_{w,v_i}$
Vertex-based hb-edge abstract information function	$f_E(\mathfrak{e}_j, v_i) = m_j(v_i)\, w(\mathfrak{e}_j)$
Hb-edge abstract information function	$F_E(\mathfrak{e}_j) = w(\mathfrak{e}_j)\, \#_m \mathfrak{e}_j$
Hb-edge bias function	$g_E(x) = x$
Hb-edge overall bias	$G_E(\mathfrak{e}_j) = w(\mathfrak{e}_j)\, \#_m \mathfrak{e}_j$

Table D.1.: Features used in the exchange-based diffusion of Chapter 2.

We define the **vertex abstract information function**: $F_V : V \to \mathbb{R}^+$ such that:

$$F_V(v_i) \triangleq \sum_{j\in[\![p]\!]} f_V(v_i, \mathfrak{e}_j).$$

We define the **probability corresponding to the hb-edge based vertex abstract information** as:

$$p^{f_V}(\mathfrak{e}_j|v_i) \triangleq \frac{f_V(v_i, \mathfrak{e}_j)}{F_V(v_i)}.$$

If we now consider a **vertex bias function**: $g_V : \mathbb{R}^+ \to \mathbb{R}^+$ applied to $f_V(v_i, \mathfrak{e}_j)$, we can define a **biased probability on the transition from vertices to hb-edges** as:

$$\widetilde{p_V}(\mathfrak{e}_j|v_i) \triangleq \frac{g_V(f_V(v_i, \mathfrak{e}_j))}{G_V(v_i)},$$

where $G_V(v_i)$, the **vertex overall bias**, is defined as:

$$G_V(v_i) \triangleq \sum_{j \in [\![p]\!]} g_V(f_V(v_i, \mathfrak{e}_j)).$$

Typical choices for g_V are: $g_V(x) = x^\alpha$ or $g_V(x) = e^{\alpha x}$. When $\alpha > 0$, higher values of f_V are encouraged, and on the contrary, when $\alpha < 0$ smaller values of f_V are encouraged.

Similarly, we introduce the **vertex-based hb-edge abstract information function**: $f_E : \mathfrak{E} \times V \to \mathbb{R}^+$.

We define the **hb-edge abstract information function**: $F_E : V \to \mathbb{R}^+$ as:

$$F_E(\mathfrak{e}_j) \triangleq \sum_{i \in [\![n]\!]} f_E(\mathfrak{e}_j, v_i).$$

We define the **probability corresponding to the vertex-based hb-edge abstract information** as:

$$p^{f_E}(v_i|\mathfrak{e}_j) \triangleq \frac{f_E(\mathfrak{e}_j, v_i)}{F_E(\mathfrak{e}_j)}.$$

If we now consider a vertex bias function: $g_E : \mathbb{R}^+ \to \mathbb{R}^+$ applied to $f_E(\mathfrak{e}_j, v_i)$, we can define a **biased probability on the transition from hb-edges to vertices** as:

$$\widetilde{p_E}(v_i|\mathfrak{e}_j) \triangleq \frac{g_E(f_E(\mathfrak{e}_j, v_i))}{G_E(\mathfrak{e}_j)},$$

where $G_E(\mathfrak{e}_j)$, the hb-edge overall bias is defined as:

$$G_E(\mathfrak{e}_j) = \sum_{i \in [\![n]\!]} g_E(f_E(\mathfrak{e}_j, v_i)).$$

Typical choices for g_E are: $g_E(x) = x^\alpha$ or $g_E(x) = e^{\alpha x}$. When $\alpha > 0$, higher values of f_E are encouraged, and on the contrary, when $\alpha < 0$ smaller values of f_E are encouraged.

D.1.3.2. Biased diffusion by exchange

We now consider a two-phase step diffusion by exchange, with a similar approach than in Chapter 2; we just describe here what differs from this approach.

The vertices hold an information value at time t given by: $\alpha_t : V \to [0;1]$.

The hb-edges hold an information value at time t given by: $\epsilon_t : \mathfrak{E} \to \mathbb{R}$.

During the first phase between time t and time $t + \frac{1}{2}$, the contribution to the value $\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j)$ from the vertex v_i is now given by:

$$\delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j|v_i) = \widetilde{p_V}(\mathfrak{e}_j|v_i)\,\alpha_t(v_i)$$

and

$$\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) = \sum_{i=1}^{n} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i).$$

We still have:

$$\alpha_{t+\frac{1}{2}}(v_i) = \alpha_t(v_i) - \sum_{j=1}^{p} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i).$$

Claim D.1 (No information on vertices at $t+\frac{1}{2}$)**.** *It holds:*

$$\forall i \in [\![n]\!] : \alpha_{t+\frac{1}{2}}(v_i) = 0.$$

Proof. For all $i \in [\![n]\!]$:

$$\begin{aligned}
\alpha_{t+\frac{1}{2}}(v_i) &= \alpha_t(v_i) - \sum_{j\in[\![p]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \\
&= \alpha_t(v_i) - \sum_{j\in[\![p]\!]} \frac{g_V(f_V(v_i,\mathfrak{e}_j))}{G_V(v_i)}\alpha_t(v_i) \\
&= \alpha_t(v_i) - \alpha_t(v_i)\frac{\sum_{j\in[\![p]\!]} g_V(f_V(v_i,\mathfrak{e}_j))}{G_V(v_i)} \\
&= 0.
\end{aligned}$$

□

Claim D.2 (Conservation of the information of the hb-graph at $t+\frac{1}{2}$)**.** *It holds:*

$$\sum_{v_i\in V} \alpha_{t+\frac{1}{2}}(v_i) + \sum_{\mathfrak{e}_j\in\mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) = 1.$$

Proof. We have:

$$\begin{aligned}
\sum_{v_i\in V} \alpha_{t+\frac{1}{2}}(v_i) + \sum_{\mathfrak{e}_j\in\mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) &= \sum_{\mathfrak{e}_j\in\mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \sum_{\mathfrak{e}_j\in\mathfrak{E}} \sum_{i\in[\![n]\!]} \delta\epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j \mid v_i) \\
&= \sum_{\mathfrak{e}_j\in\mathfrak{E}} \sum_{i\in[\![n]\!]} \frac{g_V(f_V(v_i,\mathfrak{e}_j))}{G_V(v_i)}\alpha_t(v_i) \\
&= \sum_{i\in[\![n]\!]} \alpha_t(v_i)\frac{\sum_{\mathfrak{e}_j\in\mathfrak{E}} g_V(f_V(v_i,\mathfrak{e}_j))}{G_V(v_i)} \\
&= \sum_{i\in[\![n]\!]} \alpha_t(v_i) \\
&= 1.
\end{aligned}$$

□

We introduce the **vertex overall bias matrix**:

$$G_V \triangleq \operatorname{diag}\left((G_V(v_i))_{i\in[\![n]\!]}\right)$$

and the **biased vertex-feature matrix**:

$$B_V \triangleq \left[g_V(f_V(v_i,\mathfrak{e}_j))\right]_{\substack{i\in[\![n]\!]\\ j\in[\![p]\!]}}.$$

It holds:

$$P_{\mathfrak{E},t+\frac{1}{2}} = P_{V,t} G_V^{-1} B_V. \tag{D.1}$$

During the second phase that starts at time $t+\frac{1}{2}$, the values held by the hb-edges are transferred to the vertices.

The contribution to $\alpha_{t+1}(v_i)$ given by a hb-edge $\mathfrak{e}_j$ is proportional to $\epsilon_{t+\frac{1}{2}}$ in a factor corresponding to the biased probability $\widetilde{p_E}(v_i|\mathfrak{e}_j)$:

$$\delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) = \widetilde{p_E}(v_i|\mathfrak{e}_j)\, \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j).$$

Hence, we have:

$$\alpha_{t+1}(v_i) = \sum_{j=1}^{p} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j)$$

and:

$$\epsilon_{t+1}(\mathfrak{e}_j) = \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i=1}^{n} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j).$$

Claim D.3 (The hb-edges have no value at $t+1$)**.** *It holds:*

$$\epsilon_{t+1}(\mathfrak{e}_j) = 0.$$

Proof. For all $i \in [\![p]\!]$:

$$\begin{aligned}
\epsilon_{t+1}(\mathfrak{e}_j) &= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i\in[\![n]\!]} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \\
&= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) - \sum_{i\in[\![n]\!]} \widetilde{p_E}(v_i|\mathfrak{e}_j)\, \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \left(1 - \frac{\sum_{i\in[\![n]\!]} g_E(f_E(\mathfrak{e}_j, v_i))}{G_E(\mathfrak{e}_j)}\right) \\
&= 0.
\end{aligned}$$

□

Claim D.4 (Conservation of the information of the hb-graph at $t+1$)**.** *It holds:*

$$\sum_{v_i\in V} \alpha_{t+1}(v_i) + \sum_{\mathfrak{e}_j\in\mathfrak{E}} \epsilon_{t+1}(\mathfrak{e}_j) = 1.$$

Proof.

$$\begin{aligned}
\sum_{v_i \in V} \alpha_{t+1}(v_i) + \sum_{\mathfrak{e}_j \in \mathfrak{E}} \epsilon_{t+1}(\mathfrak{e}_j) &= \sum_{v_i \in V} \alpha_{t+1}(v_i) \\
&= \sum_{v_i \in V} \sum_{\mathfrak{e}_j \in \mathfrak{E}} \delta\alpha_{t+1}(v_i \mid \mathfrak{e}_j) \\
&= \sum_{v_i \in V} \sum_{\mathfrak{e}_j \in \mathfrak{E}} \widetilde{p_E}(v_i|\mathfrak{e}_j)\, \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= \sum_{\mathfrak{e}_j \in \mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \frac{\sum\limits_{v_i \in V} g_E(f_E(\mathfrak{e}_j, v_i))}{G_E(\mathfrak{e}_\mathfrak{j})} \\
&= \sum_{\mathfrak{e}_j \in \mathfrak{E}} \epsilon_{t+\frac{1}{2}}(\mathfrak{e}_j) \\
&= 1.
\end{aligned}$$

□

We now introduce $G_{\mathfrak{E}} \triangleq \operatorname{diag}\left((G_E(\mathfrak{e}_j))_{j \in [\![p]\!]}\right)$ the diagonal matrix of size $p \times p$ and the **biased hb-edge-feature matrix**:

$$B_E \triangleq \left[g_E(f_E(\mathfrak{e}_j, v_i))\right]_{\substack{j \in [\![p]\!] \\ i \in [\![n]\!]}},$$

it comes:

$$P_{\mathfrak{E},t+\frac{1}{2}} G_{\mathfrak{E}}^{-1} B_E = P_{V,t+1}. \tag{D.2}$$

Regrouping (D.1) and (D.2):

$$P_{V,t+1} = P_{V,t} G_V^{-1} B_V G_{\mathfrak{E}}^{-1} B_E. \tag{D.3}$$

It is valuable to keep a trace of the intermediate state:

$$P_{\mathfrak{E},t+\frac{1}{2}} = P_{V,t} G_V^{-1} B_V$$

as it records the information on hb-edges.

Writing $T = G_V^{-1} B_V G_{\mathfrak{E}}^{-1} B_E$, it follows from D.3:

$$P_{V,t+1} = P_{V,t} T. \tag{D.4}$$

Claim D.5 (Stochastic transition matrix)**.** *T is a square row stochastic matrix of dimension n.*

Proof. Let consider:

$$A = (a_{ij})_{\substack{i \in [\![n]\!] \\ j \in [\![p]\!]}} = G_V^{-1} B_V \in M_{n,p}$$

and

$$B = (b_{jk})_{\substack{j \in [\![p]\!] \\ k \in [\![n]\!]}} = G_{\mathfrak{E}}^{-1} B_E \in M_{p,n}.$$

A and B are non-negative rectangular matrices. Moreover:

- $a_{ij} = \widetilde{p_V}(\mathfrak{e}_j|v_i)$ and it holds:

$$\begin{aligned}\sum_{j\in[\![p]\!]} a_{ij} &= \sum_{j\in[\![p]\!]} \widetilde{p_V}(\mathfrak{e}_j|v_i)\\ &= \sum_{j\in[\![p]\!]} \frac{g_V(f_V(v_i,\mathfrak{e}_j))}{G_V(v_i)}\\ &= 1;\end{aligned}$$

- $b_{jk} = \widetilde{p_E}(v_k|\mathfrak{e}_j)$ and it holds:

$$\begin{aligned}\sum_{k\in[\![n]\!]} b_{jk} &= \sum_{k\in[\![n]\!]} \widetilde{p_E}(v_k|\mathfrak{e}_j)\\ &= \frac{\sum\limits_{k\in[\![n]\!]} g_E(f_E(\mathfrak{e}_j,v_k))}{G_E(\mathfrak{e}_\mathrm{j})}\\ &= 1.\end{aligned}$$

We have:

$$P_{V,t+1} = P_{V,t}AB,$$

where:

$$AB = \left(\sum_{j\in[\![p]\!]} a_{ij}b_{jk}\right)_{\substack{i\in[\![n]\!]\\ k\in[\![n]\!]}}.$$

It yields:

$$\begin{aligned}\sum_{k\in[\![n]\!]}\sum_{j\in[\![p]\!]} a_{ij}b_{jk} &= \sum_{j\in[\![p]\!]} a_{ij}\sum_{k\in[\![n]\!]} b_{jk}\\ &= \sum_{j\in[\![p]\!]} a_{ij}\\ &= 1.\end{aligned}$$

Hence AB is a non-negative square matrix with its row sums all equal to 1: it is a row stochastic matrix.

□

Claim D.6 (Properties of T)**.** *Assuming that the hb-graph is connected, the biased feature exchange-based diffusion matrix T is aperiodic and irreducible.*

Proof. This stochastic matrix is aperiodic, due to the fact that any vertex of the hb-graph retrieves a part of the value it has given to the hb-edge, hence $t_{ii} > 0$ for all $i \in [\![n]\!]$.

Moreover, as the hb-graph is connected, the matrix is irreducible as any state can be joined from any other state.

□

The fact that T is a stochastic matrix aperiodic and irreducible for a connected hb-graph ensures that $(\alpha_t)_{t\in\mathbb{N}}$ converges to a stationary state which is the probability vector π_V associated to the eigenvalue 1 of T. Nonetheless, due to the presence of the different functions for vertices and hb-edges, the simplifications do not occur anymore as in the case of Chapter 2 and thus we do not have an explicit expression for the stationary state vector of the vertices.

The same occurs for the expression of the hb-edge stationary state vector π_E which is still calculated from π_V using the following formula:

$$\pi_E = \pi_V G_V^{-1} B_V.$$

D.1.3.3. Results and evaluation

We consider different biases on a randomly generated hb-graph using still the same features that in the exchange-based diffusion realized in Chapter 2. We generate hb-graphs with 200 collaborations—built out of 10,000 potential vertices—with a maximum m-cardinality of 20, such that the hb-graph has five groups that are generated with two of the vertices chosen out of a group of 10, that have to occur in each of the collaboration; there are 20 vertices that have to stand as central vertices, i.e. that ensures the connectivity in between the different groups of the hb-graph.

The approach is similar to the one taken in Chapter 2, using the same hb-edge based vertex abstract information function and the same vertex-based hb-edge abstract information function, but putting different biases as it is presented in Table D.2.

Experiment	1	2	3	4	5
Vertex bias function $g_V(x) =$	x	x^2	$x^{0.2}$	e^{2x}	e^{-2x}
Hb-edge bias function $g_E(x) =$	x	x^2	$x^{0.2}$	e^{2x}	e^{-2x}

Experiment	6	7	8	9
Vertex bias function $g_V(x) =$	x^2	e^{2x}	$x^{0.2}$	e^{-2x}
Hb-edge bias function $g_E(x) =$	x	x	x	x

Experiment	10	11	12	13	14	15
Vertex bias function $g_V(x) =$	x	x	x	x	e^{2x}	e^{-2x}
Hb-edge bias function $g_E(x) =$	x^2	e^{2x}	$x^{0.2}$	e^{-2x}	e^{-2x}	e^{2x}

Table D.2.: Biases used during the 15 experiments.

We compare the rankings obtained on vertices and hb-edges after 200 iterations of the exchange-based diffusion using the strict and large Kendall tau correlation coefficients for the different biases proposed in Table D.2. We present the results as a visualisation of correlation matrices in Figure D.1 and in Figure D.3, lines and columns of these matrices being ordered by the experiment index presented in Table D.2.

We write $\sigma_{i,t}$ the ranking obtained with Experiment i biases for $t \in \{V, E\}$ indicating whether the ranking is performed on vertices or hb-edges—the absence of t means that

it works for both rankings. The ranking obtained by Experiment 1 is the one of Chapter 2, and is called the reference ranking.

In Experiments 2 to 5, the same bias is applied to both vertices and hb-edges. In Experiments 2 and 3, the biases are increasing functions on $[0; +\infty[$, while in Experiments 4 and 5, they are decreasing functions.

Experiments 2 and 3 lead to rankings that are well correlated with the reference ranking given the large Kendall tau correlation coefficient value. The higher value of $\tau_L(\sigma_i, \sigma_j)$ compared to the one of $\tau(\sigma_i, \sigma_j)$ marks the fact that the rankings with pair of similar biases agree with the ties in this case. The exponential bias yields to a ranking that is more granular in the tail for vertices, and reshuffles the way the hb-edges are ranked; similar observations can be done for both the vertex and hb-edge rankings in Experiments 2 and 3.

In Experiments 4 and 5, the rankings remain well correlated with the reference ranking but the large Kendall tau correlation coefficient values show that there is much less agreement on the ties, but it is very punctual in the rankings, with again more discrimination with an exponential bias. This slight changes imply a reshuffling of the hb-edge rankings in both cases, significantly emphasized by the exponential form.

None of these simultaneous pairs of biases reshuffle very differently the rankings obtained in the head of the rankings of vertices, but most of them have implications on the head of the rankings of the hb-edges: typical examples are given in Figure D.5. It would need further investigations using the Jaccard index.

Dissimilarities in rankings occur when the bias is applied only to vertices or to hb-edges. The strict Kendall tau correlation coefficients between the rankings obtained when applying the bias of Experiments 6 to 9—bias on vertices—and 10 to 13—bias on hb-edges—and the reference ranking for the vertices show weak consistency for vertices with values around 0.4—Figure D.1—, while the large Kendall tau correlation coefficient values show a small disagreement with values around -0.1—Figure D.2. For hb-edges, the gap is much less between the strict—values around 0.7 as shown in Figure D.3—and large Kendall tau correlation coefficient values—with values around 0.6 as shown in Figure D.4.

Biases with same monotony variations—$g_t(x) = x^2$ and $g_t(x) = e^{2x}$ on the one hand and $g_t(x) = x^{0.2}$ and $g_t(x) = e^{-2x}$ on the other hand—have similar effects independently of their application to vertices xor to hb-edges. It is also worth to remark that increasing biases lead to rankings that have no specific agreement or disagreement with rankings of decreasing biases—as it is shown with $\tau_L(\sigma_i, \sigma_j)$ and $\tau(\sigma_i, \sigma_j)$ for $i \in [\![6; 13]\!]$.

We remark also that increasing biases applied only to vertices correlate with the corresponding decreasing biases applied only to hb-edges, and vice-versa. This is the case for Experiments 6 and 12, Experiments 7 and 13, Experiments 8 and 10, and Experiments 9 and 11 for both vertices—Figures D.1 and D.2—and hb-edges—Figures D.3 and D.4.

Finally, we conduct two more experiments—Experiments 14 and 15—combining the biases $g_t(x) = e^{2x}$ and $g_t(x) = e^{-2x}$ in two different manners. With no surprise, they reinforce the disagreement with the reference ranking both on vertices and hb-edges, with a stronger disagreement when the decreasing bias is put on vertices. We can remark that Experiment 14—$g_V(x) = e^{2x}$ and $g_E(x) = e^{-2x}$—has the strongest

correlations with the rankings of dissimilar biases that are either similar to the one of vertices—Experiments 6 and 7— or to the one of hb-edges—Experiments 12 and 13.

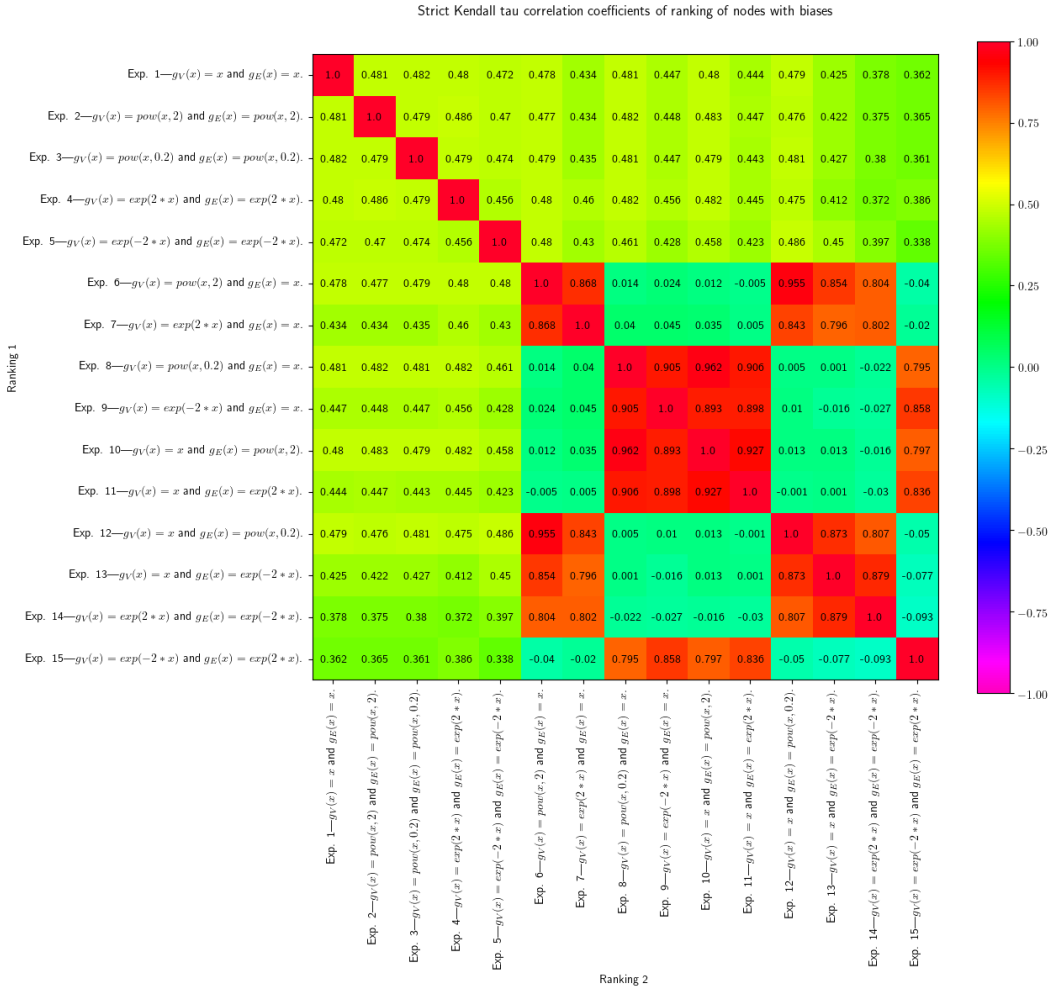


Figure D.1.: Strict Kendall tau correlation coefficient for node ranking with biases. Realized on 100 random hb-graphs with 200 hb-edges of maximal size 20, with 5 groups.

A last remark is on the variability of the results: if the values of the correlation coefficients change, from one hb-graph to another, the phenomenon observed remains the same, whatever the first hb-graph observed; however, the number of experiments performed ensures already a minimized fluctuation in these results.

D.1.3.4. Further comments

The biased-exchange-based diffusion proposed in this Chapter enhances a tunable diffusion that can be integrated into the hb-graph framework to tune adequately the ranking of the facets. The results obtained on randomly generated hb-graphs have still to be applied to real hb-graphs, with the known difficulty of the connectedness: it will be ad-

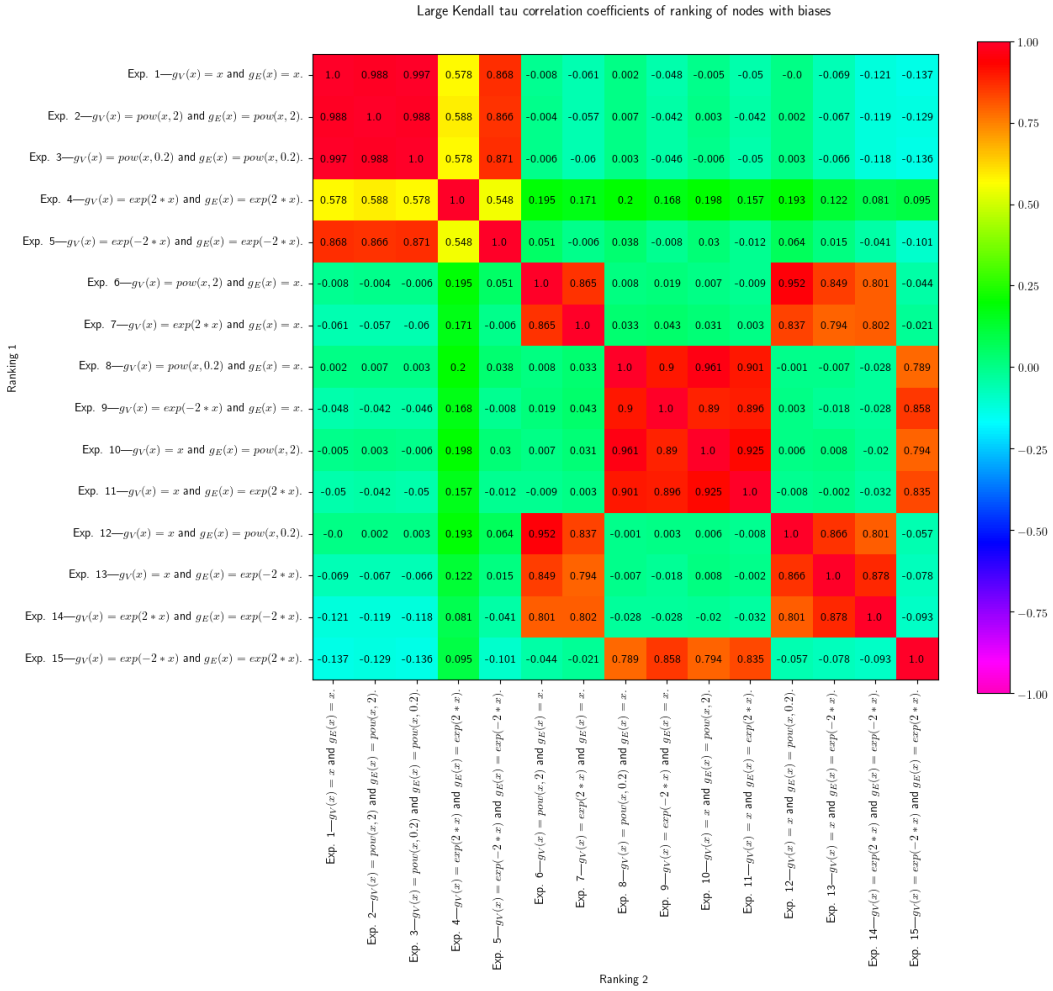


Figure D.2.: Large Kendall tau correlation coefficient for node ranking with biases. Realized on 100 random hb-graphs with 200 hb-edges of maximal size 20, with 5 groups.

dressed in future work. There remains a lot to explore on the subject in order to refine the query results obtained with real searches. The difficulty remains that in ground truth classification by experts, only a few criteria can be retained, that ends up in most cases in pairwise comparison of elements, and, hence, does not account for higher order relationships.

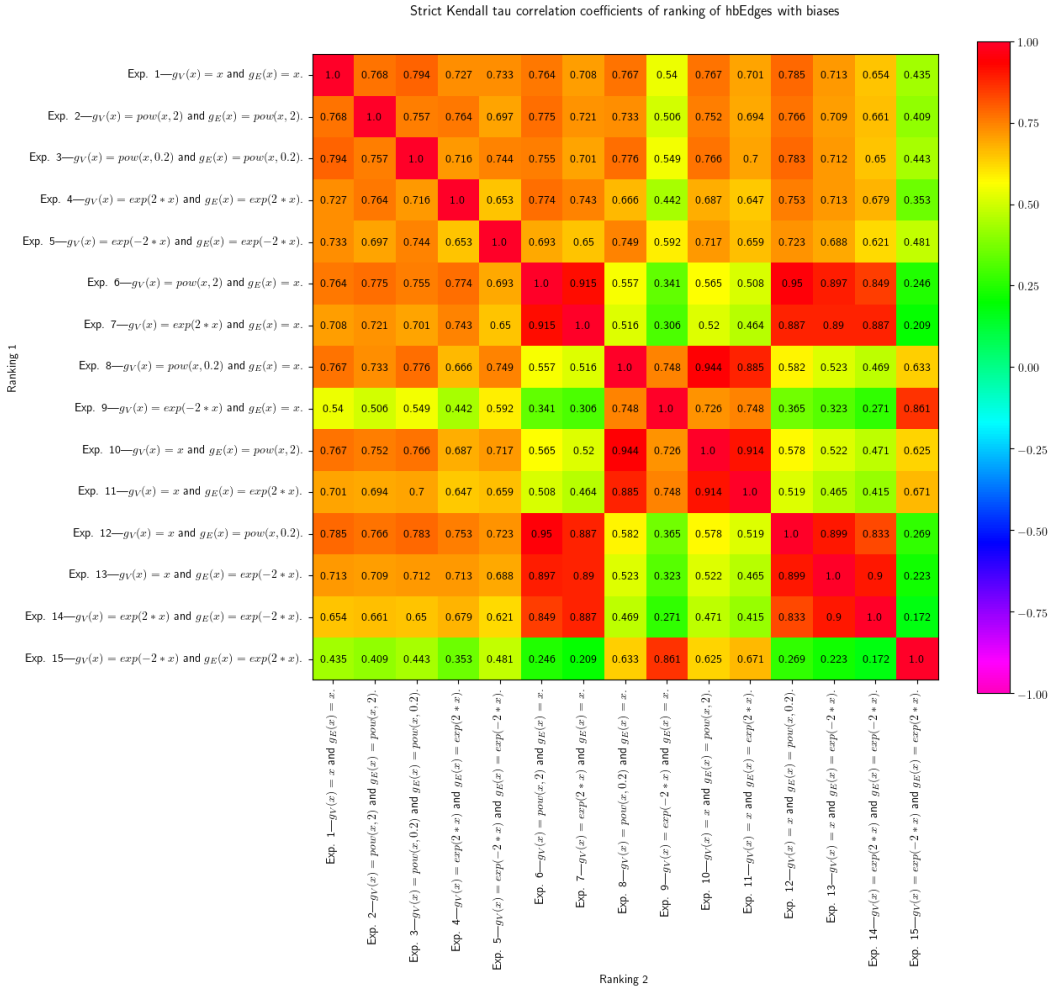


Figure D.3.: Strict Kendall tau correlation coefficient for hb-edge ranking with biases. Realized on 100 random hb-graphs with 200 hb-edges of maximal size 20, with 5 groups.

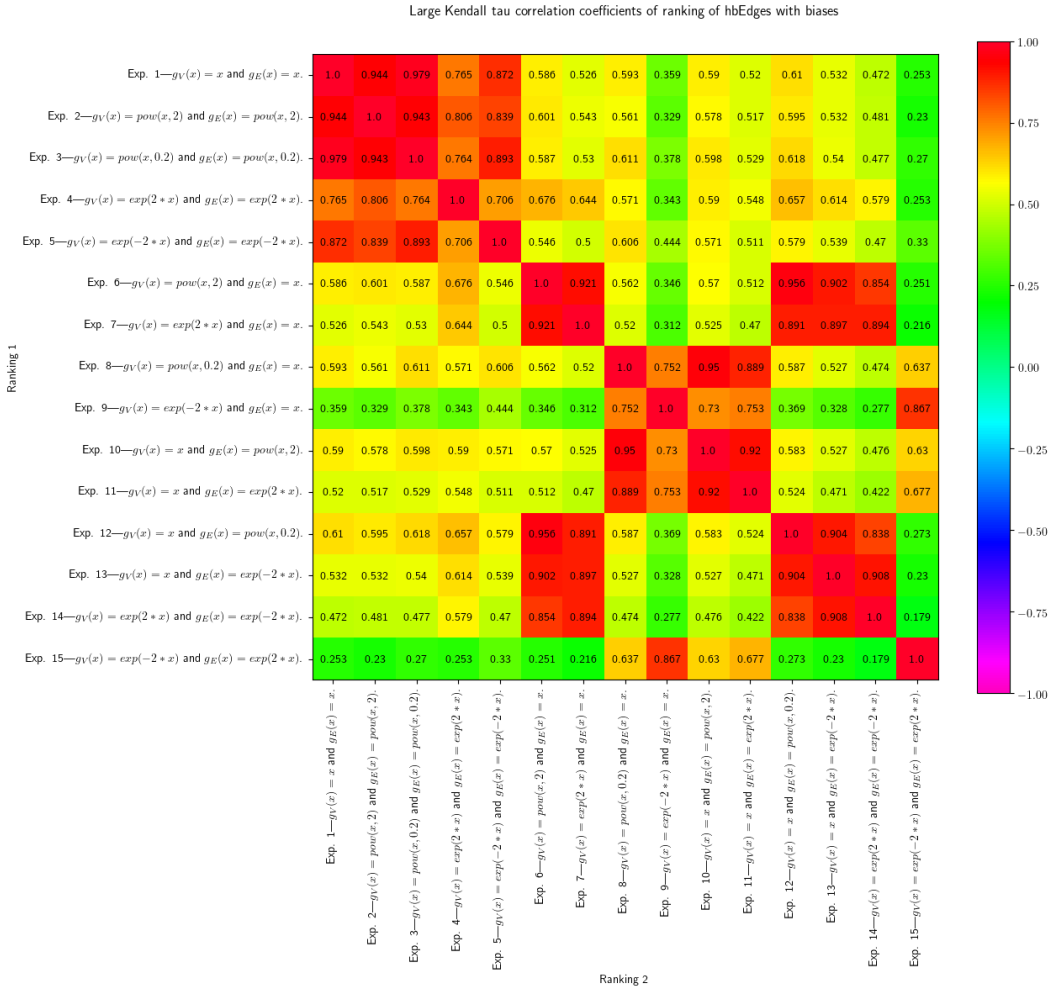
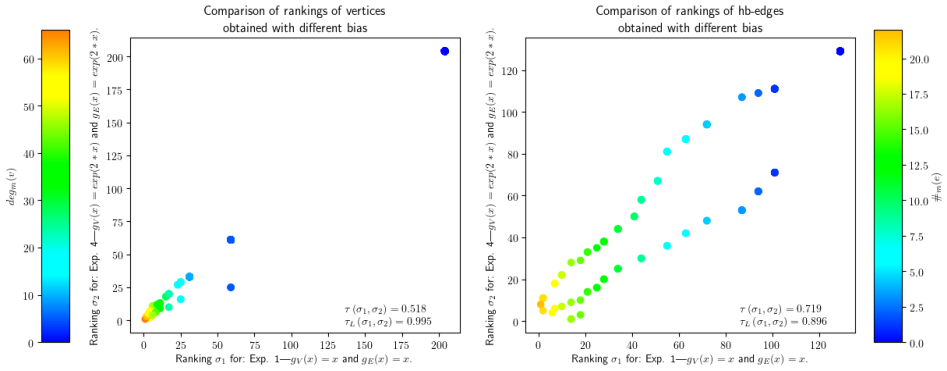
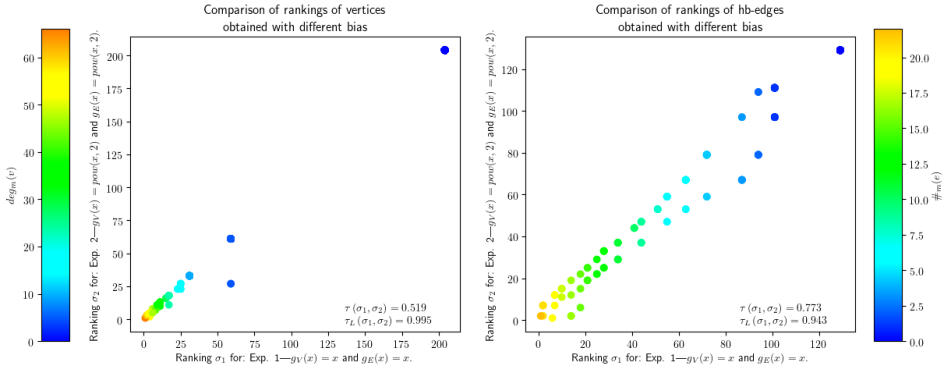


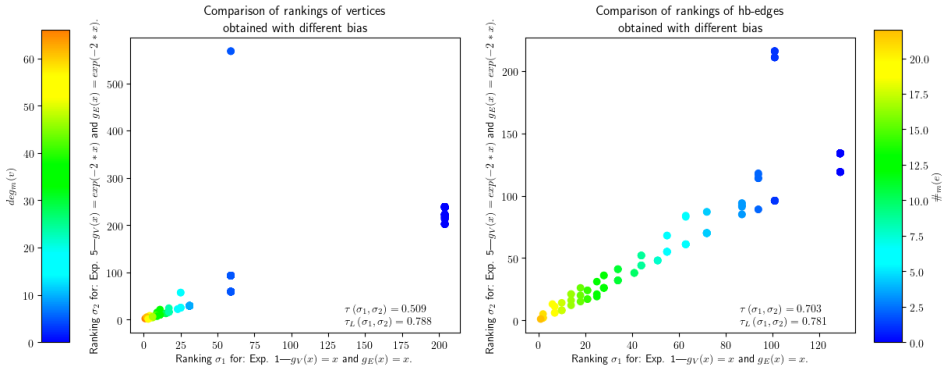
Figure D.4.: Large Kendall tau correlation coefficient for hb-edge ranking with biases. Realized on 100 random hb-graphs with 200 hb-edges of maximal size 20, with 5 groups.



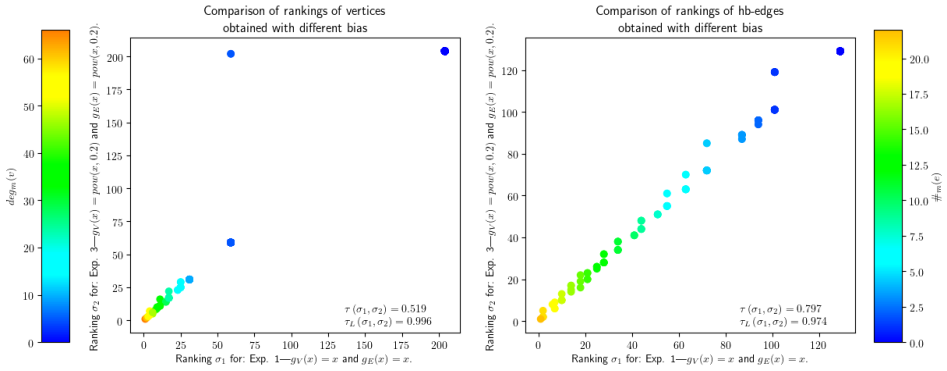
(a) First ranking: $g_V(x) = x$ and $g_E(x) = x$; Second ranking: $g_V(x) = e^{2x}$ and $g_E(x) = e^{2x}$.



(b) First ranking: $g_V(x) = x$ and $g_E(x) = x$; Second ranking: $g_V(x) = x^2$ and $g_E(x) = x^2$.



(c) First ranking: $g_V(x) = x$ and $g_E(x) = x$; Second ranking: $g_V(x) = e^{-2x}$ and $g_E(x) = e^{-2x}$.



(d) First ranking: $g_V(x) = x$ and $g_E(x) = x$; Second ranking: $g_V(x) = x^{0.2}$ and $g_E(x) = x^{0.2}$.

Figure D.5.: Effect of vertex biases on ranking.

E. Complements Chapter 3: e-adjacency tensor of natural hb-graphs

E.1. Building a tensor of e-adjacency for general hypergraphs

This chapter is based on [OLGMM17a]. A condensed version has been published in [OLGMM18a].

Prerequisites: Section B.1 and B.3.

E.1.1. Motivation

An edge in a graph connects only two vertices while a hyperedge in a hypergraph enables multiple vertices to be connected. Recent improvements in tensor spectral theory have made the research on the spectra of hypergraphs more relevant. For studying such spectra, a proper definition of general hypergraph Laplacian tensor is needed and, therefore, the concepts of adjacency and consequently of (—as it will be defined later—e-)adjacency tensor must be properly defined.

In [Pu13], a clear distinction is made between the pairwise relationship which is a binary relation and the co-occurrence relationship which is presented as the extension of the pairwise relationship to a multi-adic relationship. The notion of co-occurrence is often used in linguistic data as the simultaneous appearance of linguistic units in a reference. The co-occurrence concept can be easily extended to vertices contained in a hyperedge: we designate it in hypergraphs by the term e-adjacency.

Nonetheless, it is more than an extension. Graph edges connect vertices by pair: graph adjacency concept is clearly a pairwise relationship. For a given edge, only two vertices can be e-adjacent. Thus, adjacency and e-adjacency are equivalent in graphs.

Extending the adjacency notion to hypergraphs, two vertices are said adjacent if there exists a hyperedge that connects them. Hence, the adjacency notion still captures a binary relationship and can be modeled by an adjacency matrix. But, in this case, e-adjacency is no more a pairwise relationship, as, for a given hyperedge, more than two vertices can occur, since a hyperedge contains potentially multiple vertices. Also, a multi-adic relationship has to be captured and to be modeled using tensor. Hence, the adjacency matrix and the e-adjacency tensor of a hypergraph are two separated notions. Nonetheless, the e-adjacency tensor is often abusively named the adjacency tensor in the literature.

We aim at answering in this chapter the following research questions:

Research question E.1. *What is the proper concept of adjacency for general hypergraphs?*

This research question aim is to redefine adjacency in hypergraphs, by considering different kinds of adjacency.

Research question E.2. *How to achieve the modeling of the newly defined adjacency concept with a tensor which not only preserves all the structural information of the hypergraph, but, also, captures separately the information on its hyperedges, and that is interpretable in term of hypergraph-based process?*

This second research question will bring us to consider an uniformisation process that transforms a general hypergraph into a uniform hypergraph, and that will lead to defining a new (e-)adjacency tensor which will not solely preserve all the structural information of the hypergraph, but, also, will capture separately the information on its hyperedges.

After sketching the background and the related work on the adjacency and e-adjacency concepts for hypergraphs in Section E.1.3, we propose a new unnormalized e-adjacency tensor in Section E.1.4. Section E.1.5 tackles the particular case of graphs seen as 2-uniform hypergraphs and the link with the Disjunctive Normal Form (DNF for short). Some additional comments are addressed in Section E.1.7. A full example is given in Appendix E.2.2.

Notation Exponents are indicated into parenthesis—for instance $y^{(n)}$—when they refer to the order of the corresponding tensor. Indices are written into parenthesis when they refer to a sequence of objects—for instance $a_{(k)ij}$ is the elements at row i and column j of the matrix $A_{(k)}$. The context should make it clear.

For the convenience of readability, we write $\boldsymbol{z}_0$ for $z^1, ..., z^n$. Hence given a polynomial P, $P(\boldsymbol{z}_0)$ has to be understood as $P(z^1, ..., z^n)$.

Given additional variables $y^1, ..., y^k$, we write $\boldsymbol{z}_k$ for $z^1, ..., z^n, y^1, ..., y^k$.

$\mathcal{S}_k$ is the set of permutations on the set $[\![k]\!]$.

Hypergraph basic definitions are reminded in Chapter B. In this chapter, we make the following postulate:

Postulate E.1. *The hypergraphs considered are undirected and with no repeated hyperedges.*

E.1.2. Adjacency in hypergraphs

We consider a hypergraph $\mathcal{H} = (V, E)$, with $V = \{v_i : i \in [\![n]\!]\}$ and $E = (e_j)_{j \in [\![p]\!]}$, and two vertices $v_{i_1} \in V$ and $v_{i_2} \in V$ with $i_1 \in [\![n]\!]$ and $i_2 \in [\![n]\!]$. Traditionally, v_{i_1} and v_{i_2} are said **adjacent** if there exists $e \in E$ such that $v_{i_1} \in e$ and $v_{i_2} \in e$. This adjacency is a pairwise concept.

To switch to a higher order concept of adjacency, we consider an integer $k \geqslant 1$ and k vertices v_{i_j}, with $i_j \in [\![n]\!]$ for all $j \in [\![k]\!]$ and say that $v_{i_1}, ..., v_{i_k}$ are ***k*-adjacent** if there exists $e \in E$ such that for all $j \in [\![k]\!]$, $v_{i_j} \in e$. With $k = 2$, the usual notion of adjacency is retrieved.

Trivially:

Proposition E.1. *If k vertices are k-adjacent, then any subset of these k vertices of size $l \leqslant k$ is l-adjacent.*

The maximal k-adjacency a hypergraph can support is called the $\overline{k}$**-adjacency**. As the maximal number of vertices a hyperedge can have in a general hypergraph corresponds to the rank of this hypergraph, the $\overline{k}$-adjacency corresponds to $r_{\mathcal{H}}$-adjacency.

We now consider a hyperedge $e \in E$. The vertices constituting e are said **e-adjacent** vertices.

If the hypergraph is k-uniform, then the $\overline{k}$-adjacency is exactly the k-adjacency and is equivalent to the e-adjacency of vertices in a hyperedge.

For a general hypergraph, if it is true that $\overline{k}$-adjacency implies e-adjacency, the reciprocal is no longer true, as any hyperedge $e \in E$ such that $|e| < r_{\mathcal{H}}$[1] does not hold any $\overline{k}$-adjacent vertices. In this case, the notions of k-adjacency and of e-adjacency are definitively distinct.

E.1.3. Background and related work

E.1.3.1. The matrix approach

In Section B.1.7.2, we already mentioned that the adjacency matrix does not keep the multi-adicity. The adjacency matrix captures only links between pair of vertices: this is equivalent to extend the hypergraph to its 2-section and to capture the information of the corresponding graph. Therefore, the multi-adicity cannot be captured by a matrix.

Following a lemma cited in [DPP16], the following lemma can be enunciated:

Lemma E.1. *Let $\mathcal{H} = (V, E)$ be a hypergraph and let $v_{i_1}, v_{i_2} \in V$. If v_{i_1} and v_{i_2} are 2-adjacent in $\mathcal{H}$, then they are adjacent in the 2-section $[\mathcal{H}]_2$.*

Reciprocally, two vertices that are adjacent in the 2-section of a hypergraph are also 2-adjacent in the hypergraph. Nonetheless, it doesn't imply any isomorphism between $\mathcal{H}$ and $[\mathcal{H}]_2$: $[\mathcal{H}]_2$ appears as an equivalence class of the equivalence relation $\mathcal{R}$ defined on the set of hypergraphs with no loop, where two hypergraphs are in relation by $\mathcal{R}$, if they have the same graph as 2-section. Therefore, the 2-section of a hypergraph corresponds to an equivalence class of 2-adjacency.

Moving to the concept of e-adjacency will preserve the higher order relationships held in the hypergraph.

E.1.3.2. Existing $\overline{k}$ and e-adjacency tensors

In [MN12], an unnormalized version of the $\overline{k}-$adjacency tensor of a k-uniform hypergraph is given. This definition is also adopted in [GD17].

[1] $r_{\mathcal{H}}$ is the rank of the hypergraph $\mathcal{H}$ as defined in Section B.1.3.

Definition E.1. *The **unnormalized [$\overline{k}$-]adjacency tensor** of a k-uniform hypergraph $\mathcal{H} = (V, E)$ on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a family of hyperedges $E = (e_j)_{j\in[\![p]\!]}$ of equal cardinality k is the tensor $\mathcal{A}_{raw}$ of CHR $\boldsymbol{A}_{raw} = (a_{raw\,i_1\dots i_k})_{i_1,\dots,i_k\in[\![n]\!]}$ such that:*

$$a_{raw\,i_1\dots i_k} = \begin{cases} 1 & \text{if } \{v_{i_1}, ..., v_{i_k}\} \in E; \\ 0 & \text{otherwise.} \end{cases}$$

In [CD12], a slightly different version exists for the definition of the adjacency tensor, called the degree normalized $\overline{k}$-adjacency tensor.

Definition E.2. *The [degree normalized $\overline{k}$-]**adjacency tensor** of a k-uniform hypergraph $\mathcal{H} = (V, E)$ on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a family of hyperedges $E = (e_j)_{j\in[\![p]\!]}$ of equal cardinality k is the tensor $\mathcal{A}$ of CHR $\boldsymbol{A} = (a_{i_1\dots i_k})_{i_1,\dots,i_k\in[\![n]\!]}$ such that:*

$$a_{i_1\dots i_k} = \frac{1}{(k-1)!} \begin{cases} 1 & \text{if } \exists e \in E : e = \{v_{i_1}, ..., v_{i_k}\}; \\ 0 & \text{otherwise.} \end{cases}$$

Introducing in this definition, the coefficient $\dfrac{1}{(k-1)!}$ allows to retrieve the degree of a vertex v_i summing the elements of index i on the first mode of the tensor. Hence, we call it the degree normalized adjacency tensor.

Proposition E.2. *Let $\mathcal{H} = (V, E)$ be a k-uniform hypergraph. Let $v_i \in V$ be a vertex. It holds by considering the degree normalized $k-$adjacency tensor $\mathcal{A}$ of CHR $\boldsymbol{A} = (a_{i_1\dots i_k})_{i_1,\dots,i_k\in[\![n]\!]}$:*

$$\deg(v_i) = \sum_{i_2,\dots,i_k\in[\![n]\!]} a_{ii_2\dots i_k}.$$

Proof. On the first mode of the degree normalized adjacency tensor, for a given vertex v_i that occurs in a hyperedge $e = \{v_i, v_{i_2}, ..., v_{i_k}\}$, the elements $a_{i\sigma(i_2)\dots\sigma(i_k)} = \dfrac{1}{(k-1)!}$, where $\sigma \in \mathcal{S}_{k-1}$, exist in quantity $(k-1)!$ in the first mode. Hence $\sum\limits_{\sigma\in\mathcal{S}_{k-1}} a_{i\sigma(i_2)\dots\sigma(i_k)} = 1$.

Repeating it for all hyperedges where v_i is an element retrieves the degree of v_i.

□

This definition could be interpreted as the definition of the e-adjacency tensor for a uniform hypergraph since the notion of k-adjacency and e-adjacency are equivalent in a k-uniform hypergraph.

In [Hu13], a full study of the spectra of an uniform hypergraph using the Laplacian tensor is given. The definition of the Laplacian tensor is linked to the existence and definition of the normalized [k-]adjacency tensor.

Definition E.3. *The* [**eigenvalues**] ***normalized*** [$\overline{k}$-]***adjacency tensor*** *of a k-uniform hypergraph $\mathcal{H} = (V, E)$ on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a family of hyperedges $E = (e_j)_{j\in[\![p]\!]}$ of equal cardinality k is the tensor $\overline{\mathcal{A}}$ of CHR $\overline{\boldsymbol{A}} = (a_{i_1\dots i_k})_{i_1,\dots,i_k\in[\![n]\!]}$ such that:*

$$\overline{a_{i_1\dots i_k}} = \begin{cases} \dfrac{1}{(k-1)!}\prod\limits_{1\leqslant j\leqslant k}\dfrac{1}{\sqrt[k]{d_{i_j}}} & \text{if } \{v_{i_1},\dots,v_{i_k}\}\in E; \\ 0 & \text{otherwise.} \end{cases}$$

The aim of the normalization is motivated by the need of bounding the different tensor eigenvalues.

The normalized Laplacian tensor $\overline{\mathcal{L}}$ is given in the following definition.

Definition E.4. *The* ***normalized Laplacian tensor*** *of a k-uniform hypergraph $\mathcal{H} = (V, E)$ on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a family of hyperedges $E = (e_j)_{j\in[\![p]\!]}$ of equal cardinality k is the tensor $\overline{\mathcal{L}} = \mathcal{I} - \overline{\mathcal{A}}$ where $\mathcal{I}$ is the k-th order n-dimensional diagonal tensor of CHR that has its j-th diagonal element $i_{j\dots j} = 1$ if $d_j > 0$ and 0 otherwise.*

In [BCM17], the definition of the *[e-]*adjacency tensor is extended to general hypergraphs. The information on hyperedges of lower cardinality is spread inside the tensor. This approach focuses on the spectra of the constructed hypermatrix.

Definition E.5. *Let $\mathcal{H} = (V, E)$ be a general hypergraph on a finite set of vertices $V = \{v_i : i \in [\![n]\!]\}$ and a family of hyperedges $E = (e_j)_{j\in[\![p]\!]}$ of rank $r_{\mathcal{H}}$.*

The [**e-**]***adjacency hypermatrix*** *of $\mathcal{H}$ written $\mathcal{B}_{\mathcal{H}}$ of CHR:*

$$\boldsymbol{B_{\mathcal{H}}} = \left(b_{i_1\dots i_{r_{\mathcal{H}}}}\right)_{1\leqslant i_1,\dots,i_{r_{\mathcal{H}}}\leqslant n}$$

is such that for a hyperedge: $e = \{v_{l_1},\dots,v_{l_s}\}$ of cardinality $s \leqslant r_{\mathcal{H}}$, we have:

$$b_{p_1\dots p_{r_{\mathcal{H}}}} = \frac{s}{\alpha},\ \text{where } \alpha = \sum_{\substack{k_1,\dots,k_s\geqslant 1 \\ \sum_{i\in[\![s]\!]} k_i = r_{\mathcal{H}}}} \frac{r_{\mathcal{H}}!}{k_1!\dots k_s!}$$

with p_1, ..., $p_{r_{\mathcal{H}}}$ chosen in all possible ways from $\{l_1,\dots,l_s\}$ with at least once from each element of $\{l_1,\dots,l_s\}$.

The other values of the hypermatrix are zero.

The e-adjacency cubical hypermatrix of order the rank of the hypergraph $r_{\mathcal{H}}$ is kept at a dimension corresponding to the number of vertices n at the cost of splitting the information of the hyperedges in a varying number of elements of the tensor depending on the hyperedge cardinalities. A hyperedge e, of cardinality $\#e = s$, is represented by a set of elements in the tensor whose indices correspond to permutations of the different multisets $\left\{\left\{l_i^{k_i}, i \in [\![s]\!]\right\}\right\}$ such that: $\forall i, k_i \geqslant 1$ and $\sum\limits_{i\in[\![s]\!]} k_i = r_{\mathcal{H}}$, i.e. all the multisets

that can be built of universe $\{l_1, ..., l_s\}$ and of support $\{l_1, ..., l_s\}$ and m-cardinality $r_{\mathcal{H}}$. A detailed example is given in Appendix E.2.1.

We will come back to the consequences of filling the tensor elements this way in Chapter 4.

A s-composition of an integer m is a tuple $(k_i)_{i\in[\![s]\!]}$ such that $\sum_{i\in[\![s]\!]} k_i = m$.

Let $(k_i)_{i\in[\![s]\!]}$ be a s-composition of $r_{\mathcal{H}}$ with $k_i \geqslant 1$ for all $i \in [\![s]\!]$. Then the number of combinations of $r_{\mathcal{H}}$ elements of $\{l_1, ..., l_s\}$, at least repeated once and following the distribution given by $(k_i)_{i\in[\![s]\!]}$, is given by the multinomial coefficient:

$$\begin{pmatrix} r_{\mathcal{H}} \\ k_1 \dots k_s \end{pmatrix} = \frac{r_{\mathcal{H}}!}{\prod\limits_{i\in[\![s]\!]} k_i!}.$$

We now focus on the number of compositions $p_s(m)$ of an integer m in s ordered parts. A composition of an integer is a tuple. This is equivalent to putting $s-1$ sticks in the space between two stones of a line of m stones, i.e. in $m-1$ spaces. We have then:

$$p_s(m) = \begin{pmatrix} m-1 \\ s-1 \end{pmatrix}$$

possibilities to do it.

Hence for representing a single hyperedge of cardinality s, $p_s(r_{\mathcal{H}})$ partitions $(k_i)_{i\in[\![s]\!]}$ of $r_{\mathcal{H}}$ are involved, each of them implies to fill:

$$\begin{pmatrix} r_{\mathcal{H}} \\ k_1 \dots k_s \end{pmatrix} = \frac{r_{\mathcal{H}}!}{\prod\limits_{i\in[\![s]\!]} k_i!}$$

elements of the tensor.

It is worth mentioning another definition of e-adjacency tensor that has been recently published in [SZB19].

Definition E.6. *The* [**prefixed e-**]***adjacency tensor****—written*** $\mathcal{S}_{\mathcal{H}}$ *—of a general hypergraph* $\mathcal{H} = (V, E)$, *with* V *as vertex set*—[identified to $[\![n]\!]$]—*and* E *as hyperedge set is the cubical tensor of CHR* $\boldsymbol{S_{\mathcal{H}}} = \left(s_{i_1\dots i_{r_{\mathcal{H}}}}\right)$ *of order* $r_{\mathcal{H}}$ *and dimension* $|V|$ *with entries:*

$$s_{i_1\dots i_{r_{\mathcal{H}}}} = \begin{cases} \dfrac{1}{(s-1)!} & i_1 = \dots = i_{k-s+1}, \{i_{k-s+1}, \dots, i_k\} \in E; \\ 0 & \textit{otherwise.} \end{cases}$$

This definition has the advantage of reducing the number of elements that contain information, at the cost of having a non-symmetric hypermatrix.

In this Thesis, the proposed method to elaborate an e-adjacency tensor focuses on the fact that the construction is interpretable: a uniformisation process is proposed in which a general hypergraph is transformed into a uniform hypergraph by adding elements. The strong link made with homogeneous polynomials reinforces the choice made and allows

to retrieve a proper matrix of a uniform hypergraph at the end of the process. The additional vertices help capturing not solely the e-adjacency but also give the ability to hold the information on k-adjacency at whatever level it occurs.

The approach is based on the homogenization of sums of polynomials of different degrees and by considering a family of uniform hypergraphs. It is also motivated by the fact that the information on the cardinality of the hyperedges has to be kept in some ways and, that the elements should not be mixed with the different layers of the hypergraph.

E.1.4. Towards an e-adjacency tensor of a general hypergraph

To build an e-adjacency tensor for a general hypergraph, we need a way to store elements which represent the hyperedges. As these hyperedges have different cardinalities, the representation of the e-adjacency for vertices in a unique tensor can be achieved only by filling the hyperedges with additional elements. The problem of finding an e-adjacency tensor of a general hypergraph is then transformed into a uniformisation problem.

This uniformisation process should be at least interpretable in term of uniform hypergraphs. It should capture the structural information of the hypergraph, which includes information on the number of hyperedges, degrees of vertices and additional information on the main features of the hypergraph.

We propose a framework based on homogeneous polynomials that are iteratively summed by weighting with technical coefficients and homogenized. This uniformisation process allows the construction of a weighted uniform hypergraph. The technical coefficients are adjusted such that the handshake lemma holds in the constructed uniform hypergraph.

E.1.4.1. Family of tensors attached to a hypergraph

Let $\mathcal{H} = (V, E)$ be a hypergraph. A hypergraph can be decomposed in a family of uniform hypergraphs.

To achieve it, let $\mathcal{R}$ be the equivalency relation: $e\mathcal{R}e' \Leftrightarrow |e| = |e'|$.

$E/\mathcal{R}$ is the set of classes of hyperedges of same cardinality. The elements of $E/\mathcal{R}$ are the sets: $E_k \triangleq \{e \in E : |e| = k\}$.

Considering $K = \{k : E_k \in E/\mathcal{R}\}$, we set $E_k \triangleq \emptyset$ for all $k \in [\![r_{\mathcal{H}}]\!] \setminus K$.

Let us consider the hypergraphs: $\mathcal{H}_k \triangleq (V, E_k)$ for all $k \in [\![r_{\mathcal{H}}]\!]$ which are all k-uniform.

It holds that: $E = \bigcup_{k \in [\![r_{\mathcal{H}}]\!]} E_k$ and $E_j \cap E_k = \emptyset$ for all $j \neq k$, hence the family $(E_k)_{k \in [\![r_{\mathcal{H}}]\!]}$ forms a partition of E which is unique by the way it has been defined.

Hence:

$$\mathcal{H} = \bigoplus_{k \in [\![r_{\mathcal{H}}]\!]} \mathcal{H}_k.$$

The hypergraph $\mathcal{H}$ is said to be **decomposed** into a family of hypergraphs $(\mathcal{H}_k)_{k \in [\![r_{\mathcal{H}}]\!]}$ where $\mathcal{H}_k$ is k-uniform.

An illustration of the decomposition of a hypergraph into a family of uniform hypergraphs is shown in Figure E.1. This family of uniform hypergraphs decomposes the original hypergraph in layers. A layer holds a k-uniform hypergraph—with $k \in [\![r_{\mathcal{H}}]\!]$—: therefore, the layer is said to be of level k.

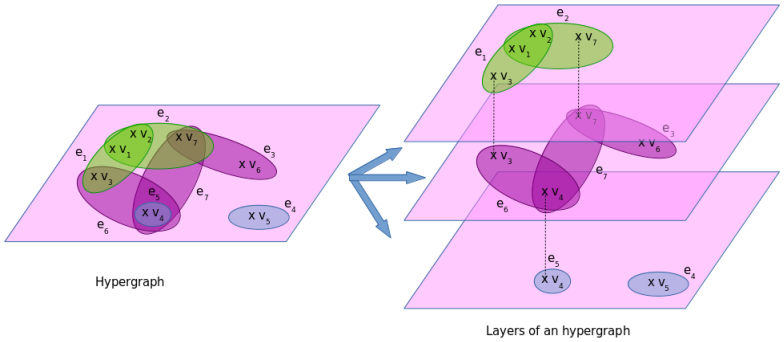


Figure E.1.: Illustration of a hypergraph decomposed in three layers of uniform hypergraphs.

At each k-uniform hypergraph $\mathcal{H}_k$ can be mapped a (k-adjacency) e-adjacency tensor $\mathcal{A}_k$ with CHR $\boldsymbol{A_k}$ of order k which is hyper-cubic and symmetric of dimension $|V| = n$. This hypermatrix can be unnormalized or normalized, but this choice has to be made the same for the whole family of $\mathcal{H}_k$.

Once chosen a type of tensor—normalized or unnormalized—, the hypergraph $\mathcal{H}$ is then fully described by the family of e-adjacency tensors $\mathcal{A}_{\mathcal{H}} = (\mathcal{A}_k)$. In the case where all the $\mathcal{A}_k$ are chosen normalized, this family is said pre-normalized. The final choice will be made further in Section E.1.4.7 and explained to fulfill the expectations listed in the next section.

E.1.4.2. Expectations for an e-adjacency tensor for a general hypergraph

The definition of the family of tensors attached to a general hypergraph has the advantage to open the way to the spectral theory for uniform hypergraphs which is quite advanced.

Nonetheless, many problems remain in keeping a family of tensors of different orders: for instance, studying the spectra of the whole hypergraph could be hard to achieve by this means. Also, it is necessary to get an e-adjacency tensor for the whole hypergraph which retains the information on the whole structure.

The underlying idea is to "fill" the hyperedges with sufficient elements such that the general hypergraph is transformed into an uniform hypergraph through a uniformisation process. A similar approach has been taken in [BCM17] where the filling elements are the vertices belonging to the hyperedge itself, involving a hb-graph m-uniformisation, as we will tackle it in Chapter 3. In the next sections, the justification of the approach taken will be made via homogeneous polynomials. Before getting to the construction, expected properties of such a tensor have to be listed.

Expectation E.1. *The tensor should be symmetric and its generation should be simple.*

This expectation emphasizes the fact that in between two constructed e-adjacency tensors, the one that can be easily generated has to be chosen: this includes the fact that the tensor has to be described in a simple way.

Expectation E.2. *The unnormalized e-adjacency tensor keeps the overall structure of the hypergraph.*

This expectation emphasized the fact that the e-adjacency tensor should help to retrieve the different layers of the hypergraph, in such a way that it respects the structure of each layer.

Expectation E.3. *The e-adjacency tensor should allow the retrieval of the vertex degrees.*

In the adjacency matrix of a graph, the information on the vertex degrees is encoded directly. It is still the case with the k-adjacency degree normalised tensor defined by [Sha13] and [PZ14].

Expectation E.4. *The e-adjacency tensor should allow the retrieval of the number of hyperedges of same cardinality.*

This expectation is linked with the expectation on the keeping of the overall structure of the hypergraph.

Expectation E.5. *The tensor should be invariant to vertex permutations either globally or at least locally.*

This expectation is motivated by the fact that in a hyperedge the vertices have no order. This expectation can be local, since the special vertices will not have the same status than the ones of the original hypergraph. Also, the invariance by permutation is expected on the vertices of the original hypergraph.

Expectation E.6. *The e-adjacency tensor should allow the unique retrieval of the hypergraph it is originated from.*

This expectation seems important when rebuilding the original hypergraph from its e-adjacency tensor. Hence, all the necessary information to uniquely retrieve the original hyperedges has to be encoded in the tensor.

Expectation E.7. *Giving the choice of two representations the e-adjacency tensor chosen should be the sparsest one.*

Sparsity enables information compression, and lowers both the space and the complexity calculus. Also, sparsity is a desirable property for some statistical reasons as shown in [Nik00] or expected in [BDE09] for signal processing and image encoding.

E.1.4.3. Tensors family and family of homogeneous polynomials

To construct a homogeneous polynomial representing a general hypergraph, the family of e-adjacency tensors obtained in the previous section has to be mapped to a family of homogeneous polynomials. This mapping is used in [CQU15] where the author links symmetric tensors and homogeneous polynomials of degree s to show that the problem of the CP decomposition of different symmetric tensors of different orders and the decoupled representation of multivariate polynomial maps are related.

Homogeneous polynomials family of a hypergraph We consider a hypergraph $\mathcal{H}$ with its family of unnormalized tensor $\mathcal{A}_{\mathcal{H}} = (\mathcal{A}_k)$ of CHR which is cubical and symmetric of order k and dimension n. We can also attach to $\mathcal{H}$ a family $P_{\mathcal{H}} = (P_k)$ of homogeneous polynomials with $\deg(P_k) = k$. To achieve this, we consider a vector $z \in \mathbb{K}^n$ and the multilinear matrix multiplication defined in B.3.2: it holds that $(z)_{[k]}.\boldsymbol{A_k}$—with $(z)_{[k]} \triangleq (z, ..., z) \in (\mathbb{K}^n)^k$—contains only one element written $P_k(z^1, ..., z^n) = P_k(\boldsymbol{z}_0)$ such that:

$$P_k(\boldsymbol{z}_0) = \sum_{i_1,...,i_k \in [\![n]\!]} a_{(k)\, i_1 ... i_k} z^{i_1} ... z^{i_k}. \tag{E.1}$$

The formulation of P_k can be reduced taking into account that $\mathcal{A}_k$ is symmetric for a uniform hypergraph:

$$P_k(\boldsymbol{z}_0) = \sum_{1 \leqslant i_1 \leqslant ... \leqslant i_k \leqslant n} k! a_{(k)\, i_1 ... i_k} z^{i_1} ... z^{i_k}. \tag{E.2}$$

Writing:

$$\widetilde{P_k}(\boldsymbol{z}_0) = \sum_{1 \leqslant i_1 \leqslant ... \leqslant i_k \leqslant n} a_{(k)\, i_1 ... i_k} z^{i_1} ... z^{i_k}. \tag{E.3}$$

the reduced form of P_k, it holds:

$$P_k(\boldsymbol{z}_0) = k! \widetilde{P_k}(\boldsymbol{z}_0).$$

Writing for $1 \leqslant i_1 \leqslant ... \leqslant i_k \leqslant n$:

$$\alpha_{(k)\, i_1 ... i_k} \triangleq k! a_{(k)\, i_1 ... i_k}$$

and: $\alpha_{(k)\, \sigma(i_1) ... \sigma(i_k)} \triangleq 0$ for $\sigma \in \mathcal{S}_k$, $\sigma \neq \mathrm{Id}$.

It holds:

$$\begin{aligned} P_k(\boldsymbol{z}_0) &= \sum_{1 \leqslant i_1 \leqslant ... \leqslant i_k \leqslant n} \alpha_{(k)\, i_1 ... i_k} z^{i_1} ... z^{i_k} \\ &= \sum_{1 \leqslant i_1, ..., i_k \leqslant n} \alpha_{(k)\, i_1 ... i_k} z^{i_1} ... z^{i_k}. \end{aligned} \tag{E.4}$$

and:

$$\widetilde{P_k}(\boldsymbol{z}_0) = \sum_{1 \leqslant i_1 \leqslant ... \leqslant i_k \leqslant n} \frac{\alpha_{(k)\, i_1 ... i_k}}{k!} z^{i_1} ... z^{i_k}. \tag{E.5}$$

Reversibility of the process Reciprocally, given a homogeneous polynomial of degree k, a unique hyper-cubic tensor of order k can be built: its dimension is the number of different variables in the homogeneous polynomial. If the homogeneous polynomial of degree k is supposed reduced and ordered, then only one hyper-cubic and symmetric hypermatrix can be built. It reflects uniquely a k-uniform hypergraph adding the constraint that each monomial is composed of the product of k different variables.

Proposition E.3. *Let* $P(z_0) = \sum_{i_1,...,i_k \in [\![n]\!]} a_{i_1...i_k} z^{i_1}...z^{i_k}$ *be a homogeneous polynomial of degree* k *where:*

- *for* $j \neq k : z^j \neq z^k$;
- *for all* $j \in [\![n]\!] : \deg(z^j) = 1$;
- *and such that for all* $\sigma \in \mathcal{S}_k : a_{\sigma(i_1)...\sigma(i_k)} = a_{i_1...i_k}$.

Then P *is the homogeneous polynomial attached to a unique* k*-uniform hypergraph* $\mathcal{H} = (V, E, w)$*—up to the indexing of vertices.*

Proof. Considering the vertices $(v_i)_{i \in [\![n]\!]}$ labeled by the elements of $[\![n]\!]$.

If $a_{i_1...i_k} \neq 0$, then for all $\sigma \in \mathcal{S}_k : a_{\sigma(i_1)...\sigma(i_k)}$ a unique hyperedge e_j is attached corresponding to the vertices $v_{i_1}, ..., v_{i_k}$ and which has weight $w(e_j) = k a_{i_1...i_k}$.

□

E.1.4.4. Uniformisation and homogenization processes

A single tensor is always easier to use than a family of tensors; the same applies for homogeneous polynomials. Building a single tensor from tensors of different order requires to fill in the "gaps"; summing homogeneous polynomials of varying degrees always gives a new polynomial: but, most frequently, this polynomial is not anymore homogeneous. Homogenization techniques for polynomials are wellknown and require additional variables.

Different homogenization processes can be envisaged to get a homogeneous polynomial that represents a single cubic and symmetric tensor by making different choices on the variables added in the homogenization phase. As a link has been made between the variables and the vertices of the hypergraph, we want that this link continues to occur during the homogenization of the polynomial as each term of the reduced polynomial corresponds to a unique hyperedge in the original hypergraph; the homogenization process is interpretable in terms of hypergraph uniformisation process of the original hypergraph: hypergraph uniformisation process and polynomial homogenization process are the two sides of the same coin.

So far, we have separated the original hypergraph $\mathcal{H}$ in layers of increasing k-uniform hypergraphs $\mathcal{H}_k$ such that:

$$\mathcal{H} = \bigoplus_{k \in [\![r_{\mathcal{H}}]\!]} \mathcal{H}_k.$$

Each k-uniform hypergraph can be represented by a symmetric and cubic tensor. This symmetric and cubic tensor is mapped to a homogeneous polynomial. The reduced

homogeneous polynomial is interpretable, if we omit the coefficients of each term, as a disjunctive normal form. Each term of the homogeneous polynomial is a conjunctive form which corresponds to simultaneous presence of vertices in a hyperedge: adding all the layers allows to retrieve the original hypergraph; adding the different homogeneous polynomials allows to retrieve the disjunctive normal form associated with the original hypergraph.

In the hypergraph uniformisation process, iterative steps are done starting with the lower layers to the upper layers of the hypergraph. In parallel, the polynomial homogenization process is the algebraic justification of the hypergraph uniformisation process. It allows to retrieve a polynomial attached to the uniform hypergraph built at each step and hence a tensor.

Hypergraph uniformisation process We can describe algorithmically the hypergraph uniformisation process: it transforms the original hypergraph in a uniform hypergraph.

Initialisation The initialisation requires that each layer hypergraph is associated to a weighted hypergraph.

To each uniform hypergraph $\mathcal{H}_k$, we associate a weighted hypergraph $\mathcal{H}_{w_k,k} = (V, E_k, w_k)$, with: $\forall e \in E_k : w_k(e) = c_k$, $c_k \in \mathbb{R}^{+*}$.

The coefficients c_k are technical coefficients that will be chosen when considering the homogenization process and the fulfillment of the expectations of the e-adjacency tensor. The coefficients c_k can be seen as dilatation coefficients only dependent of the layers of the original hypergraph.

We initialize: $k := 1$ and $\mathcal{K}_w := \mathcal{H}_{w_1,1}$ and generate $r_{\mathcal{H}} - 1$ distinct vertices y_j, $j \in [\![r_{\mathcal{H}} - 1]\!]$ that are not in V.

Iterative steps Each step in the hypergraph uniformisation process includes three phases: an inflation phase, a merging phase and a concluding phase.

Inflation phase: The inflation phase consists in increasing the cardinality of each hyperedge obtained from the hypergraph built at the former step to reach the cardinality of the hyperedges of the second hypergraph used in the merge phase.

We define a first operation that consists in inflating each hyperedge of the hypergraph with an additional vertex that does not belong to the original vertex to obtain a vertex-augmented hypergraph.

Definition E.7. *The* ***y-vertex-augmented hypergraph*** *of a weighted hypergraph* $\mathcal{H}_w = (V, E, w)$ *is the hypergraph* $\overline{\mathcal{H}_{\overline{w}}} \triangleq \left(\overline{V}, \overline{E}, \overline{w}\right)$ *obtained by the following rules:*

- $y \notin V$;
- $\overline{V} \triangleq V \cup \{y\}$;
- *Writing* $\phi : \mathcal{P}(V) \to \mathcal{P}\left(\overline{V}\right)$ *the map such that for* $A \in \mathcal{P}(V) : \phi(A) = A \cup \{y\}$, *it holds:*

– $\overline{E} \triangleq \{\phi(e) : e \in E\}$;

– $\forall e \in E,\ \overline{w}(\phi(e)) \triangleq w(e)$.

Proposition E.4. *The vertex-augmented hypergraph of a k-uniform hypergraph is a $k+1$-uniform hypergraph.*

The inflation phase at step k generates from $\mathcal{K}_w$ the y_k-vertex augmented hypergraph $\overline{\mathcal{K}_{\overline{w}}}$.

As $\mathcal{K}_w$ is k-uniform at step k, $\overline{\mathcal{K}_{\overline{w}}}$ is $k+1$-uniform.

Merging phase: The merging phase generates the sum of two weighted hypergraphs called the merged hypergraph.

We start by defining the merging operation of two hypergraphs.

Definition E.8. *The **merged hypergraph** $\widehat{\mathcal{H}_{\widehat{w}}} \triangleq \left(\widehat{V}, \widehat{E}, \widehat{w}\right)$ of two weighted hypergraphs $\mathcal{H}_a = (V_a, E_a, w_a)$ and $\mathcal{H}_b = (V_b, E_b, w_b)$ is the weighted hypergraph defined as follow:*

- $\widehat{V} \triangleq V_a \cup V_b$;
- $\widehat{E_{k+1}} \triangleq E_a \cup E_b$;
- $\forall e \in E_a :\ \widehat{w}(e) \triangleq \overline{w_a}(e)$ *and* $\forall e \in E_b :\ \widehat{w}(e) \triangleq w_b$.

The merging phase at step k generates from $\overline{\mathcal{K}_w}$ and $\mathcal{H}_{w_{k+1},k+1}$ the merged hypergraph $\widehat{\mathcal{K}_{\widehat{w}}}$. As it is generated from two $k+1$-uniform hypergraphs, it is also a $k+1$-uniform hypergraph.

Ending phase: If k equals $r_{\mathcal{H}} - 1$, the iterative part ends and returns $\widehat{\mathcal{K}_{\widehat{w}}}$.

Otherwise, a next step is needed with $\mathcal{K}_w := \widehat{\mathcal{K}_{\widehat{w}}}$ and $k := k+1$.

Termination: We obtain by this algorithm a weighted $r_{\mathcal{H}}$-uniform hypergraph associated to $\mathcal{H}$ which is the returned hypergraph from the iterative part: we write it $\widehat{\mathcal{H}_{\widehat{w}}} = \left(\widehat{V}, \widehat{E}, \widehat{w}\right)$.

Definition E.9. *Writing $V_s = \{y_j : j \in [\![r_{\mathcal{H}} - 1]\!]\}$, $\widehat{\mathcal{H}_{\widehat{w}}} \triangleq \left(\widehat{V}, \widehat{E}, \widehat{w}\right)$ is called the **V_s-layered uniform** of $\mathcal{H}$.*

Proposition E.5. *Let $\mathcal{H} = (V, E)$ be a hypergraph of order $r_{\mathcal{H}}$.*

Let consider $V_s = \{y_j : j \in [\![1, r_{\mathcal{H}} - 1]\!]\}$ such that $V \cap V_s = \emptyset$ and let $\widehat{\mathcal{H}_{\widehat{w}}} = \left(\widehat{V}, \widehat{E}, \widehat{w}\right)$ be the V_s-layered uniform of $\mathcal{H}$. Then:

- *(V, V_s) is a partition of $\widehat{V}$;*

- $\forall e \in E, \exists! \widehat{e} \in \widehat{E} : e \subseteq \widehat{e} \wedge \widehat{e} \backslash e = \{y_j : j \in [\![|e| , r_{\mathcal{H}} - 1]\!]\}$.

Proof. The way the V_s-layered uniform of $\mathcal{H}$ is generated justifies the results.

□

Proposition E.6. *Let $\mathcal{H} = (V, E)$ be a hypergraph of order $r_{\mathcal{H}}$.*
Let consider $V_{s,i} = \{y_j : j \in [\![r_{\mathcal{H}} - 1]\!]\}$ such that $V_s = \bigcup_{i \in [\![r_{\mathcal{H}} - 1]\!]} V_{s,i}$, $V \cap V_s = \emptyset$ and let $\widehat{\mathcal{H}_{\widehat{w}}} = \left(\widehat{V}, \widehat{E}, \widehat{w}\right)$ be the V_s-layered uniform of $\mathcal{H}$.
Then, the vertices of $\mathcal{H}$ that are e-adjacent in $\mathcal{H}$ in a hyperedge e are e-adjacent with the vertices of $V_{s,|e|}$ in $\widehat{\mathcal{H}_{\widehat{w}}}$.
Reciprocally, if vertices are e-adjacent in $\widehat{\mathcal{H}_{\widehat{w}}}$, the ones that are not in V_s are e-adjacent in $\mathcal{H}$.

As a consequence, $\widehat{\mathcal{H}_{\widehat{w}}}$ captures the e-adjacency of $\mathcal{H}$.

Polynomial homogenization process In the polynomial homogenization process, we build a new family $R_{\mathcal{H}} = (R_k)$ of homogeneous polynomials of degree k iteratively from the family of homogeneous polynomials $P_{\mathcal{H}} = (P_k)$ by following the subsequent steps that respect the construction phases in Figure E.2. Each of these steps can be linked to the steps of the homogenization process.

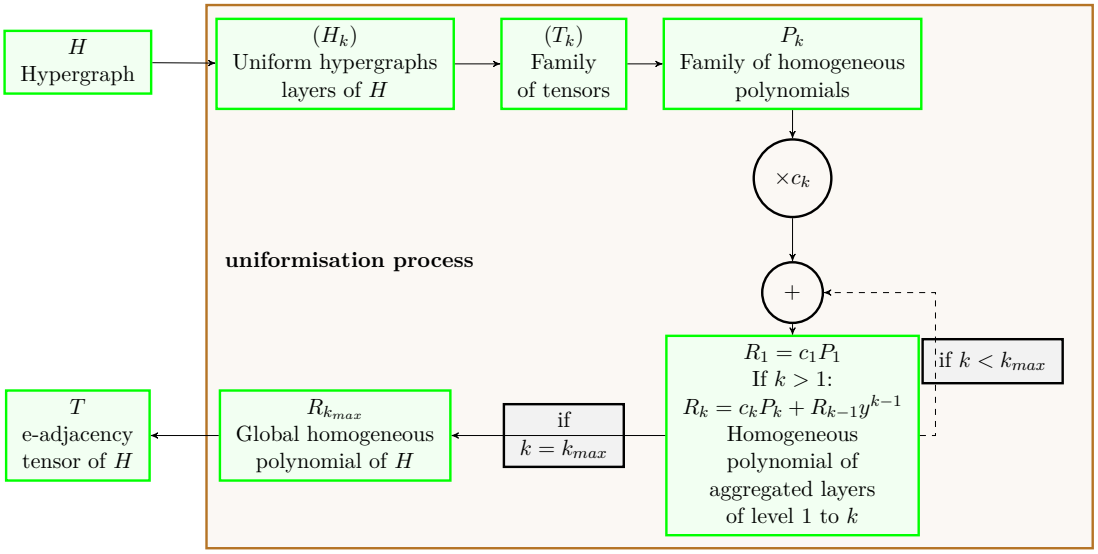


Figure E.2.: Construction phases of the e-adjacency tensor.

Initialisation Each polynomial P_k, $k \in [\![r_{\mathcal{H}}]\!]$ attached to the corresponding layer k-uniform hypergraph $\mathcal{H}_k$ is multiplied by a coefficient c_k equal to the corresponding dilatation coefficient of the hypergraph uniformisation process as discussed further in Section E.1.4.6. $c_k P_k$ represents the reduced homogeneous polynomial attached to $\mathcal{H}_{w_k,k}$.

Using the notations $\boldsymbol{z}_{k-1}$ and $\boldsymbol{z}_0$ as defined in E.1.1, we initialize: $k := 1$ and:

$$R_k\left(\boldsymbol{z}_{k-1}\right) = R_1\left(\boldsymbol{z}_0\right) = c_1 P_1\left(\boldsymbol{z}_0\right) = c_1 \sum_{i=1}^{n} a_{(1)\,i} z^i.$$

We generate $r_{\mathcal{H}} - 1$ distinct 2 by 2 variables y^j, $j \in [\![r_{\mathcal{H}} - 1]\!]$ that are also pairwise distinct from the z^i, $i \in [\![n]\!]$.

Iterative steps At each step, we sum the current $R_k\left(z_{k-1}\right)$ with the next layer coefficiented polynomial $c_{k+1}P_{k+1}$ in a way to obtain a homogeneous polynomial $R_{k+1}\left(z_k\right)$. To help the understanding we describe the first step, then generalize to any step.

Case $k = 1$: To build R_2, an homogenization of the sum of R_1 and c_2P_2 is needed. It holds:

$$R_1\left(\boldsymbol{z}_0\right) + c_2 P_2\left(\boldsymbol{z}_0\right) = c_1 \sum_{i \in [\![n]\!]} a_{(1)\,i} z^i + c_2 \sum_{i_1, i_2 \in [\![n]\!]} a_{(2)\,i_1 i_2} z^{i_1} z^{i_2}.$$

To achieve the homogenization of $R_1\left(\boldsymbol{z}_0\right) + c_2 P_2\left(\boldsymbol{z}_0\right)$, a new variable y^1 is introduced.

It follows for $y^1 \neq 0$:

$$\begin{aligned} R_2\left(\boldsymbol{z}_1\right) &= R_2\left(\boldsymbol{z}_0, y^1\right) \\ &= y^{1(2)}\left(R_1\left(\frac{\boldsymbol{z}_0}{y^1}\right) + c_2 P_2\left(\frac{\boldsymbol{z}_0}{y^{1\,(2)}}\right)\right) \\ &= c_1 \sum_{i \in [\![n]\!]} a_{(1)\,i} z^i y^1 + c_2 \sum_{i_1, i_2 \in [\![n]\!]} a_{(2)\,i_1 i_2} z^{i_1} z^{i_2}. \end{aligned}$$

By continuous prolongation of R_2, it is set:

$$R_2\left(\boldsymbol{z}_0, 0\right) = c_2 \sum_{i_1, i_2 \in [\![n]\!]} a_{(2)\,i_1 i_2} z^{i_1} z^{i_2}.$$

In this step, the weighted degree 1 polynomial $R_1\left(\boldsymbol{z}_0\right) = c_1 P_1\left(\boldsymbol{z}_0\right)$ attached to $\mathcal{H}_{w_1,1}$ is transformed in a degree 2 homogeneous polynomial $y^1 R_1\left(z_0\right) = c_1 y^1 P_1\left(\boldsymbol{z}_0\right) : y^1 R_1\left(\boldsymbol{z}_0\right)$ corresponds to the homogeneous polynomial of the weighted y_1-vertex-augmented 1-uniform hypergraph $\overline{\mathcal{H}_{\overline{w_1},1}}$ built during the inflation phase in the hypergraph uniformisation process.

$y_1 R_1\left(\boldsymbol{z}_0\right)$ is then summed with the homogeneous polynomial c_2P_2 attached to $\mathcal{H}_{w_2,2}$ to get a homogeneous polynomial of degree 2: $R_2\left(\boldsymbol{z}_1\right).\ R_2\left(\boldsymbol{z}_1\right)$ is the homogeneous polynomial of the merged 2-uniform hypergraph $\widehat{\mathcal{H}_{\widehat{w_1},1}}$ of $\overline{\mathcal{H}_{w_1,1}}$ and $\mathcal{H}_{w_2,2}$.

General case: Assuming that $R_k\left(\boldsymbol{z}_{k-1}\right)$ is a homogeneous polynomial of degree k that can be written as:

$$R_k\left(\boldsymbol{z}_{k-1}\right) = \sum_{j \in [\![k]\!]} c_j \sum_{i_1, \ldots, i_j \in [\![n]\!]} a_{(j)\,i_1 \ldots i_j} z^{i_1} \ldots z^{i_j} \prod_{l=j}^{k-1} y^l,$$

with the convention that: $\prod_{l=j}^{k-1} y^l = 1$ if $j > k-1$ and $\boldsymbol{z}_{k-1} = z^1, \ldots, z^n, y^1, \ldots, y^{k-1}$.

R_{k+1} is built as a homogeneous polynomial from the sum of R_k and $c_{k+1}P_{k+1}$ by adding a variable y^k and factorizing by its $k+1$-th power.

Therefore, for $y^{k-1} \neq 0$:

$$\begin{aligned} R_{k+1}\left(\boldsymbol{z}_k\right) &= y^{k\,(k+1)}\left(R_k\left(\frac{\boldsymbol{z}_{k-1}}{y^{k\,(k)}}\right) + c_{k+1}P_{k+1}\left(\frac{\boldsymbol{z}_0}{y^{k\,(k+1)}}\right)\right) \\ &= \left(\sum_{j\in[\![k]\!]} c_j \sum_{i_1,\ldots,i_j\in[\![n]\!]} a_{(j)\,i_1\ldots i_j} z^{i_1}\ldots z^{i_j} \prod_{l=j}^{k-1} y^l\right) y^k \\ &\quad + c_{k+1} \sum_{i_1,\ldots,i_{k+1}\in[\![n]\!]} a_{(k+1)\,i_1\,\ldots\,i_{k+1}} z^{i_1}\ldots z^{i_{k+1}}. \end{aligned}$$

And for $y_k = 0$, it is set by continuous prolongation:

$$R_{k+1}\left(\boldsymbol{z}_{k-1}, 0\right) = c_{k+1} \sum_{i_1,\ldots,i_{k+1}\in[\![n]\!]} a_{(k+1)\,i_1\,\ldots\,i_{k+1}} z^{i_1}\ldots z^{i_{k+1}}.$$

The fact that $P_{k+1}\left(\boldsymbol{z}_0\right)$ can be null does not prevent to do the step: the degree of R_k will then be elevated of 1.

The interpretation of this step is similar to the one done for the case $k = 1$.

End step phase: If k equals $r_{\mathcal{H}} - 1$, the iterative part ends up, else $k := k + 1$ and, the next iteration is started.

Conclusion: The algorithm build a family of homogeneous polynomial which is interpretable in term of uniformisation of a hypergraph.

E.1.4.5. Building an unnormalized symmetric tensor from this family of homogeneous polynomials

Based on $\boldsymbol{R_{\mathcal{H}}}$:

It is now insightful to interpret the built polynomials.

The notation $\boldsymbol{w}_{(k)} = w^1_{(k)}, ..., w^{n+k-1}_{(k)}$ is used, with for $i \in [\![n]\!]$: $w^i_{(k)} = z^i$ and for $j \in [\![k-1]\!]$: $w^{n+j}_{(k)} = y^j$.

- The interpretation of R_1 is trivial as it holds the single element hyperedges of the hypergraph;
- R_2 is a homogeneous polynomial with $n+1$ variables of order 2:

$$\begin{aligned} R_2\left(\boldsymbol{z}_1\right) &= c_1 \sum_{i\in[\![n]\!]} a_{(1)\,i} z^i y^1 + c_2 \sum_{i_1,i_2\in[\![n]\!]} a_{(2)\,i_1 i_2} z^{i_1} z^{i_2} \\ &= c_1 \sum_{i\in[\![n]\!]} \alpha_{(1)\,i} z^i y^1 + c_2 \sum_{1\leqslant i_1\leqslant i_2\leqslant n} \alpha_{(2)\,i_1 i_2} z^{i_1} z^{i_2}. \end{aligned}$$

  It can be rewritten:

$$R_2\left(\boldsymbol{w}_{(2)}\right) = \sum_{i_1,i_2\in[\![n+1]\!]} r_{(2)\,i_1 i_2} w^{i_1}_{(2)} w^{i_2}_{(2)}$$

  where:

– for $1 \leqslant i_1 \leqslant i_2 \leqslant n$ and $\sigma \in \mathcal{S}_2$:

$$r_{(2)\,\sigma(i_1)\sigma(i_2)} = \frac{c_2 \alpha_{(2)\,i_1 i_2}}{2!} = c_2 a_{(2)\,i_1 i_2};$$

– for $i \in [\![n]\!]$ and $\sigma \in \mathcal{S}_2$:

$$r_{(2)\,\sigma(i)\,\sigma(n+1)} = \frac{c_1 \alpha_{(1)\,i}}{2!} = \frac{c_1 a_{(1)\,i}}{2!};$$

– the other coefficients: $r_{(2)\,i_1 i_2}$ are null.

Also R_2 can be linked to a symmetric hyper-cubic tensor of order 2 and dimension $n+1$.

- R_k is a homogeneous polynomial with $n+k-1$ variables of order k.

$$\begin{aligned} R_k\left(\boldsymbol{z}_{k-1}\right) &= \sum_{j \in [\![k]\!]} c_j \sum_{i_1,\ldots,i_j \in [\![n]\!]} a_{(j)\,i_1 \ldots i_j} z^{i_1} \ldots z^{i_j} \prod_{l=j}^{k-1} y^l \\ &= \sum_{j \in [\![k]\!]} c_j \sum_{1 \leqslant i_1 \leqslant \ldots \leqslant i_j \leqslant n} \alpha_{(j)\,i_1 \ldots i_j} z^{i_1} \ldots z^{i_j} \prod_{l=j}^{k-1} y^l \end{aligned}$$

with the convention that: $\prod_{l=j}^{k-1} y^l = 1$ if $j > k-1$.

It can be rewritten:

$$R_k\left(\boldsymbol{w}_{(k)}\right) = \sum_{i_1,\ldots,i_k \in [\![n+k-1]\!]} r_{(k)\,i_1 \ldots i_k} w_{(k)}^{i_1} \ldots w_{(k)}^{i_k},$$

where:

– for $1 \leqslant i_1 \leqslant \ldots \leqslant i_k \leqslant n$, for all $j \in [\![k-1]\!]$, for all $\sigma \in \mathcal{S}_k$:

* $r_{(k)\,\sigma(i_1)\ldots\sigma(i_k)} = \dfrac{c_k \alpha_{(k)\,i_1 \ldots i_k}}{k!} = c_k a_{(k)\,i_1 \ldots i_k};$

* $r_{(k)\,\sigma(i_1)\ldots\sigma(i_j)\sigma(n+j)\ldots\sigma(n+k-1)} = \dfrac{c_j \alpha_{(j)\,i_1 \ldots i_j}}{k!} = \dfrac{j!}{k!} c_j a_{(j)\,i_1 \ldots i_j};$

– the other elements $r_{(k)\,i_1 \ldots i_k}$ are null.

Also R_k can be linked to a symmetric hyper-cubic tensor of order k and dimension $n+k-1$ written $\boldsymbol{R_k}$ whose elements are $r_{(k)\,i_1 \ldots i_k}$.

The hypermatrix $\boldsymbol{R_{r_{\mathcal{H}}}}$ is called the unnormalized e-adjacency hypermatrix.

E.1.4.6. Interpretation and choice of the coefficients for the unnormalized tensor

There are different ways of setting the coefficients $c_1, \ldots, c_{k_{\max}}$ that are used. These coefficients can be seen as a way of normalizing the tensors of e-adjacency generated from the k-uniform hypergraphs.

A first way of choosing them is to set them all equal to 1. In this case, no normalization occurs. The impact on the e-adjacency tensor of the original hypergraph is that e-adjacency in hyperedges of size k have a weight of k times bigger than the e-adjacency in hyperedges of size 1.

A second way of choosing these coefficients is to consider that in a k-uniform hypergraph, every hyperedge holds k vertices and then contributes k to the total degree. Representing this k-uniform hypergraph by the $k-$adjacency degree normalized tensor $\mathcal{A}_k$ of CHR $\boldsymbol{A_k} = \left(a_{(k)\,i_1\ldots i_k}\right)_{i_1,\ldots,i_k\in[\![n]\!]}$, it holds a revisited hand-shake lemma for k-uniform hypergraphs:

$$\begin{aligned}\sum_{1\leqslant i_1,\ldots,i_k\leqslant n} a_{(k)\,i_1\ldots i_k} &= \sum_{i\in[\![n]\!]}\sum_{i_2,\ldots,i_k\in[\![n]\!]} a_{(k)\,ii_2\ldots i_k}\\ &= \sum_{i\in[\![n]\!]} d_{(k)\,i}\\ &= k\,|E_k|\end{aligned}$$

where $d_{(k)\,i}$ is the degree of the vertex v_i in $\mathcal{H}_k$.

This formula can be extended to general hypergraphs:

$$\begin{aligned}|E| &= \sum_{k\in[\![r_{\mathcal{H}}]\!]} |E_k|\\ &= \sum_{k\in[\![r_{\mathcal{H}}]\!]} \frac{1}{k}\sum_{i\in[\![n]\!]} d_{(k)\,i}\\ &= \sum_{k\in[\![r_{\mathcal{H}}]\!]} \frac{1}{k}\sum_{i_1,\ldots,i_k\in[\![n]\!]} a_{(k)\,i_1\ldots i_k}.\end{aligned}$$

For general hypergraphs, the tensor is of order $r_{\mathcal{H}}$.

$$\begin{aligned}\sum_{i_1,\ldots,i_{r_{\mathcal{H}}}\in[\![n+r_{\mathcal{H}}-1]\!]} r_{i_1\ldots i_{r_{\mathcal{H}}}} &= \sum_{i\in[\![n+r_{\mathcal{H}}-1]\!]}\sum_{i_2,\ldots,i_k\in[\![n+r_{\mathcal{H}}-1]\!]} r_{ii_2\ldots i_{r_{\mathcal{H}}}}\\ &= \sum_{i\in[\![n]\!]}\deg(v_i) + \sum_{i=n+1}^{n+r_{\mathcal{H}}-1}\deg(y_i).\end{aligned}$$

The constructed tensor corresponds to the tensor of a $r_{\mathcal{H}}$-uniform hypergraph with $n + r_{\mathcal{H}} - 1$ vertices. It holds:

$$\sum_{i_1,\ldots,i_{r_{\mathcal{H}}}\in[\![n+r_{\mathcal{H}}-1]\!]} r_{i_1\ldots i_{r_{\mathcal{H}}}} = r_{\mathcal{H}}|E|.$$

And, therefore:

$$\sum_{i_1,\ldots,i_{r_{\mathcal{H}}}\in[\![n+r_{\mathcal{H}}-1]\!]} r_{i_1\ldots i_{r_{\mathcal{H}}}} = \sum_{k\in[\![r_{\mathcal{H}}]\!]} \frac{r_{\mathcal{H}}}{k}\sum_{i_1,\ldots,i_k\in[\![n]\!]} a_{(k)\,i_1\ldots i_k}.$$

Also, $c_k = \dfrac{r_{\mathcal{H}}}{k}$ seems to be a good choice in this case.

The final choice will be taken in the next paragraph to answer to the required specifications on degrees. It will also fix the matrix chosen for the uniform hypergraphs.

E.1.4.7. Fulfillment of the unnormalized e-adjacency tensor expectations

Guarantee E.1. *The tensor should be symmetric and its generation should be simple.*

Proof. By construction, the e-adjacency tensor is symmetric. To generate it, only one element has to be described for a given hyperedge the other elements obtained by permutation of the indices being the same. Also, the built e-adjacency tensor is fully described by giving $|E|$ elements.

□

Guarantee E.2. *The unnormalized e-adjacency tensor keeps the overall structure of the hypergraph.*

Proof. By construction, the layer of level equal or under j can be seen in the mode 1 at the $n+j$-th component of the mode. To have only elements of level j one can project this mode so that it keeps only the first n dimensions. □

In the expectations of the built co-tensors listed in the paragraph E.1.4.2, the e-adjacency tensor should allow the retrieval of the degree of the vertices. It implies to fix the choice of the k-adjacency tensors used to model each layer of the hypergraph as well as the normalizing coefficient.

Let consider for $k \in [\![r_{\mathcal{H}}]\!]$, $l \in [\![2, r_{\mathcal{H}}]\!]$ and $i \in [\![n + r_{\mathcal{H}} - 1]\!]$:

$$I_{k,l,i} = \{(i_1, ..., i_l) : i_1 = i \wedge \forall j \in [\![2, l]\!] : i_j \in [\![n + k - 1]\!]\}$$

and its subset of ordered tuples:

$$OI_{k,l,i} = \big\{(i_1,...,i_l) : (i_1,...,i_l) \in I_{k,l,i} \wedge \big(l \geqslant 2 \implies \forall (j_1,j_2) \in [\![2,l]\!]^2 : j_1 < j_2 \implies i_{j_1} < i_{j_2}\big)\big\}.$$

Then:

$$\begin{aligned}\sum_{(i_1,...,i_{r_{\mathcal{H}}}) \in I_{r_{\mathcal{H}},r_{\mathcal{H}},i}} r_{i_1...i_{r_{\mathcal{H}}}} &= \sum_{(i_1,...,i_{r_{\mathcal{H}}}) \in OI_{r_{\mathcal{H}},r_{\mathcal{H}},i}} (r_{\mathcal{H}} - 1)! r_{i_1...i_{r_{\mathcal{H}}}} \\ &= \sum_{j=1}^{r_{\mathcal{H}}} \sum_{(i_1,...,i_j) \in OI_{r_{\mathcal{H}},j,i}} \frac{j! c_j a_{(j)\, i_1...i_j}}{r_{\mathcal{H}}}.\end{aligned}$$

Hence, the expectation on the retrieval of degree imposes to set $c_j a_{(j)\, i_1...i_j} = \dfrac{r_{\mathcal{H}}}{j!}$ for the elements of $\boldsymbol{A}_{(j)}$ that are not null, which is coherent with the usage of the coefficient $c_j = \dfrac{r_{\mathcal{H}}}{j}$ and of the degree-normalized tensor for j-uniform hypergraph where not null elements are equals to: $\dfrac{1}{(j-1)!}$. This choice is then made for the rest of the Chapter.

Remark E.1.1. *By choosing $c_j = \dfrac{r_{\mathcal{H}}}{j}$ and the degree-normalized tensor for j-uniform hypergraph where non-zero elements are equals to: $\dfrac{1}{(j-1)!}$, it follows that:*

$$r_{i_1...i_{r_{\mathcal{H}}}} = \frac{1}{(r_{\mathcal{H}} - 1)!}$$

for all elements which is consistent with the fact that we have built a $r_{\mathcal{H}}$-uniform hypergraph by filling each hyperedge with additional vertices. This method is similar to make a plaster molding from a footprint in the sand: the filling elements help reveal the underlying structure.

With this choice, writing $\mathbb{1}_{e\in E} : \begin{cases} 1 & \text{if } e \in E \\ 0 & \text{otherwise} \end{cases}$, it holds:

$$\sum_{(i_1,\dots,i_{r_{\mathcal{H}}})\in I_{r_{\mathcal{H}},r_{\mathcal{H}},i}} r_{i_1\dots i_{r_{\mathcal{H}}}} = \sum_{j\in[\![r_{\mathcal{H}}]\!]} \sum_{(i_1,\dots,i_j)\in OI_{r_{\mathcal{H}},j,i}} \mathbb{1}_{\{v_{i_1},\dots,v_{i_j}\}\in E}.$$

It follows immediately:

Guarantee E.3. *The unnormalized e-adjacency tensor allows the retrieval of the degree of the vertices of the hypergraph.*

Proof. Defining for $i \in [\![n]\!]$: $d_i = \deg(v_i)$.

From the previous choice, it follows that:

$$\begin{aligned}
\sum_{(i_1,\dots,i_{r_{\mathcal{H}}})\in I_{r_{\mathcal{H}},r_{\mathcal{H}},i}} r_{i_1\dots i_{r_{\mathcal{H}}}} &= \sum_{(i_1,\dots,i_{r_{\mathcal{H}}})\in OI_{r_{\mathcal{H}},r_{\mathcal{H}},i}} (r_{\mathcal{H}}-1)!\, r_{i_1\dots i_{r_{\mathcal{H}}}} \\
&= \sum_{j\in[\![r_{\mathcal{H}}]\!]} \sum_{(i_1,\dots,i_j)\in OI_{r_{\mathcal{H}},j,i}} \frac{j!c_j a_{(j)\,i_1\dots i_j}}{r_{\mathcal{H}}} \\
&= \sum_{j\in[\![r_{\mathcal{H}}]\!]} \sum_{(i_1,\dots,i_j)\in OI_{r_{\mathcal{H}},j,i}} \mathbb{1}_{\{v_{i_1},\dots,v_{i_j}\}\in E} \\
&= \deg(v_i)
\end{aligned}$$

as $\frac{j!c_j a_{(j)\,i_1\dots i_j}}{r_{\mathcal{H}}} = 1$ only for hyperedges where v_i is in it (and they are counted only once for each hyperedge).

□

Guarantee E.4. *The unnormalized e-adjacency tensor allows the retrieval of the number of hyperedges of same cardinality.*

Proof. Defining $d_{n+i} = |\{e : |e| \leqslant i\}|$ for $i \in [\![r_{\mathcal{H}}]\!]$.

$$\begin{aligned}
\sum_{(i_1,\dots,i_{r_{\mathcal{H}}})\in I_{r_{\mathcal{H}},r_{\mathcal{H}},n+i}} r_{i_1\dots i_{r_{\mathcal{H}}}} &= \sum_{j\in[\![i]\!]} \sum_{1\leqslant l_1<\dots<l_j\leqslant n} (r_{\mathcal{H}}-1)! \frac{j!c_j a_{(j)\,i_1\dots i_j}}{r_{\mathcal{H}}!} \\
&= \sum_{j\in[\![i]\!]} \sum_{1\leqslant l_1<\dots<l_j\leqslant n} \mathbb{1}_{\{v_{l_1},\dots,v_{l_j}\}\in E}
\end{aligned}$$

due to the fact that $r_{n+i\,i_2\dots i_{r_{\mathcal{H}}}} \neq 0$ if and only if there exists at most i indices i_2 to $i_{r_{\mathcal{H}}}$ that are between 1 and n which correspond to vertices in the general hypergraph and the other indices have value strictly above n which represent additional vertices.

It follows:

$$\sum_{(i_1,\dots,i_{r_{\mathcal{H}}})\in I_{r_{\mathcal{H}},r_{\mathcal{H}},n+i}} r_{i_1\dots i_{r_{\mathcal{H}}}} \quad = \quad d_{n+i}.$$

We set: $d_{n+r_{\mathcal{H}}} = |E|$.

Also d_{n+j} allows to retrieve the number of hyperedges of cardinality equal or less than j.

Therefore:

- for $j \in [\![2, r_{\mathcal{H}}]\!]$: $|\{e : |e| = j\}| = d_{n+j} - d_{n+j-1}$;
- for $j = 1$: $|\{e : |e| = 1\}| = d_{n+1}$.

An other way of keeping directly the cardinality of the layer $r_{\mathcal{H}}$ in the e-adjacency tensor would be to store it in an additional variable $y_{r_{\mathcal{H}}}$.

□

We gather here the fullfilment of Expectation E.5 and E.6.

Guarantee E.5. *The e-adjacency tensor is unique up to the labeling of the vertices for a given hypergraph.*

Reciprocally, given the e-adjacency tensor and the number of vertices, the associated hypergraph is unique.

Proof. Given a hypergraph, the process of decomposition in layers is bijective as well as the formalization by degree normalized k-adjacency tensor. Given the coefficients, the process of building the e-adjacency homogeneous polynomial is also unique and the reversion to a symmetric cubic tensor is unique.

Given the e-adjacency tensor and the number of vertices, as the e-adjacency tensor is symmetric, up to the labeling of the vertices, considering that the first n variables encoded in the e-adjacency tensor in each direction represents variables associated to vertices of the hypergraph and the last variables in each direction encode the information of cardinality, it is possible to retrieve each layer of the hypergraph uniquely and, consequently, the whole hypergraph.

□

Expectation E.7 will be used in Chapter 3 for the final choice of the tensor.

E.1.4.8. Interpretation of the e-adjacency tensor

The general hypergraph layer decomposition allows to retrieve uniform hypergraphs that can be separately modeled by e-adjacency (or equivalently k-adjacency) tensor of k-uniform hypergraphs. We have shown that filling these different layers with additional vertices help to uniformize the original hypergraph by keeping the e-adjacency. The coefficients used in the iterative process have to be seen as weights on the hyperedges of the final $r_{\mathcal{H}}$-uniform hypergraph: these coefficients allow to retrieve the right number of

hyperedges from the uniformed hypergraph tensor so that it corresponds to the number of hyperedges of the original hypergraph.

The additional dimensions in the e-adjacency tensor allow to retrieve the cardinality of the hyperedges. By decomposing a hypergraph in a set of uniform hypergraphs, the hyperedges are normalized using their cardinality.

The iterative approach principle is illustrated in Figure E.3: vertices that are added at each level give indication on the original cardinality of the hyperedge it is added to.

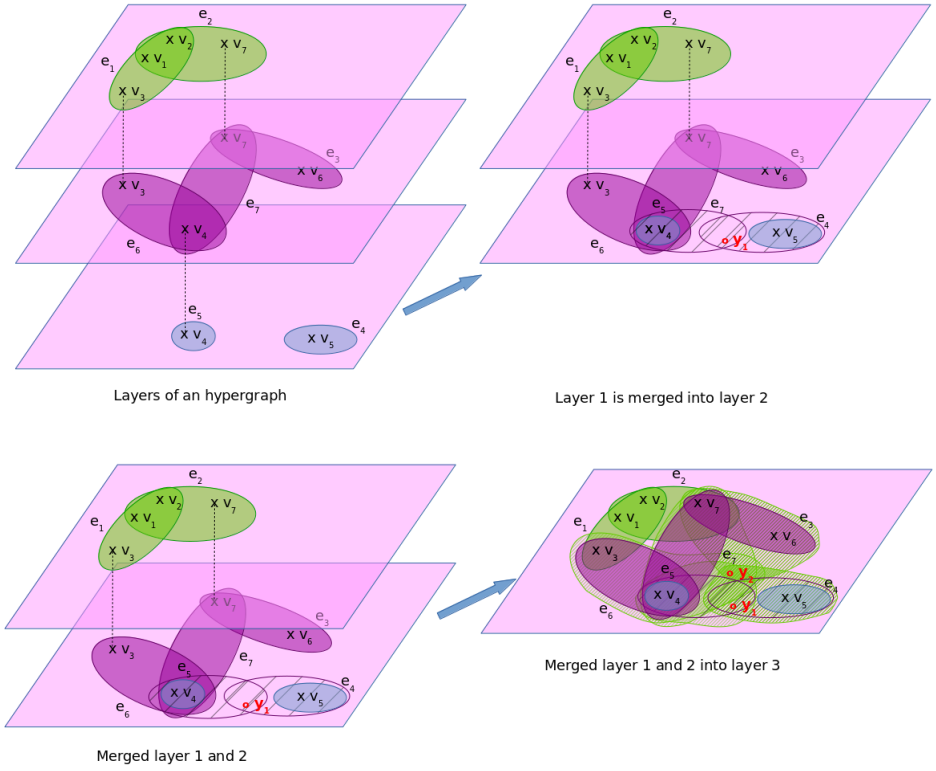


Figure E.3.: Illustration of the iterative approach concept on an example.
In the iterative approach, the layers of level n and $n+1$ are merged together into the layer $n+1$ by adding a filling vertex to the hyperedges of the layer n. On this example, during the first step the layer 1 and 2 are merged to form a 2-uniform hypergraph. In the second step, the 2-uniform hypergraph obtained in the first step is merged to the layer 3 to obtain a 3-uniform hypergraph.

Viewed in an other way, e-adjacency hypermatrix of uniform hypergraph do not need an extra dimension as the hyperedges are uniform, therefore, there is no ambiguity. Adding an extra variable allows to capture the dimensionality of each hyperedge while preventing any ambiguity on the meaning of each element of the tensor.

E.1.5. Some comments on the e-adjacency tensor

E.1.5.1. The particular case of graphs

As a graph $G = (V, E)$ with $|V| = n$ can always be seen as a 2-uniform hypergraph $\mathcal{H}_G$, the approach given in this paragraph should allow to retrieve in a coherent way the spectral theory for normal graphs.

The hypergraph that contains the 2-uniform hypergraph is then composed of an empty level 1 layer and a level 2 layer that contains only $\mathcal{H}_G$.

Let A be the adjacency matrix of G. The e-adjacency tensor of the corresponding 2-uniform hypergraph is of order 2 and obtained from A by multiplying it by c_2 and adding one row and one column of zero. Therefore, the e-adjacency tensor of the two levels of the corresponding hypergraph is: $\mathcal{A}= \left(\begin{array}{c|c} c_2 A & 0 \\ \hline 0 & 0 \end{array}\right)$.

Also, as an eigenvalue λ of $\mathcal{A}$ seen as a matrix is a solution of the characteristic polynomial $\det(\mathcal{A} - \lambda I) = 0 \Leftrightarrow -\lambda \det(c_2 A - \lambda I) = 0 \Leftrightarrow -\lambda c_2^n \det\left(A - \dfrac{\lambda}{c_2} I\right) = 0$, the eigenvalues of $\mathcal{A}$ are c_2 times the ones of A and one additional 0 eigenvalue. This last eigenvalue is attached to the eigenvector $(0...0\,1)^T$. The other eigenvalues have same eigenvectors than A with one additional $n+1$ component which is 0.

Proof. Let consider $Y = \begin{pmatrix} X \\ y \end{pmatrix}$ with X vector of dimension n. Let λ be an eigenvalue of $\mathcal{A}$ and Y an eigenvector of $\mathcal{A}$.

$\mathcal{A}Y = \lambda Y \Leftrightarrow \mathcal{A}\begin{pmatrix} X \\ y \end{pmatrix} = \lambda \begin{pmatrix} X \\ y \end{pmatrix} \Leftrightarrow (c_2 A - \lambda I_n) X = 0 \wedge -\lambda y = 0 \Leftrightarrow X$ eigenvalue of A attached to $\dfrac{\lambda}{c_2}$. y can be always taken equals to 0 to fit the second condition.

□

Therefore, globally there is no change in the spectrum: the eigenvectors remain, the eigenvalues of the initial graph are multiplied by the normalizing coefficient.

E.1.5.2. e-adjacency tensor and disjunctive normal form

Let $\mathcal{H} = (V, E)$ be a hypergraph, $\mathcal{A}$ its e-adjacency tensor and $\tilde{R}_{r_\mathcal{H}}$ the reduced attached homogeneous polynomial:

$$\tilde{R}_{r_\mathcal{H}}\left(\boldsymbol{w}_{(r_\mathcal{H})}\right) = \sum_{1 \leqslant i_1 < \ldots < i_k \leqslant n + r_\mathcal{H} - 1} \tilde{r}_{(r_\mathcal{H})\, i_1 \ldots i_{r_\mathcal{H}}} w^{i_1}_{(r_\mathcal{H})} \ldots w^{i_{r_\mathcal{H}}}_{(r_\mathcal{H})}$$

with $\tilde{r}_{(r_\mathcal{H})\, i_1 \ldots i_{r_\mathcal{H}}} = r_\mathcal{H}! r_{(r_\mathcal{H})\, i_1 \ldots i_{r_\mathcal{H}}}$.

The variables $\left(w^i_{(r_\mathcal{H})}\right)_{i \in [\![n + r_\mathcal{H} - 1]\!]}$ of $R_{k_{\max}}$ can be considered as Boolean variables and, therefore, $R_{k_{\max}}$ can be considered as a Boolean function. The variables $w^i_{(r_\mathcal{H})}$ for

$i \in [\![n]\!]$ captures whether a vertex belongs to the considered hyperedge and for $i \in [\![n+1, n+r_{\mathcal{H}}-1]\!]$ to the layer of level $i-n$.

This Boolean homogeneous polynomial $P_B\left(\boldsymbol{w}_{(r_{\mathcal{H}})}\right)$ is in full disjunctive normal form as it is a sum of products of Boolean variables holding only once in each product and where the conjunctive terms are made of $r_{\mathcal{H}}$ variables.

$P_{B\,r_{\mathcal{H}}}(z_0) = P_B\left(z_0, \underbrace{0, ..., 0}_{r_{\mathcal{H}}-1}\right)$ allows to retrieve the part of the full DNF which stores hyperedges of size $r_{\mathcal{H}}$.

$P_{B\,r_{\mathcal{H}}-1}(z_0) = P_B\left(z_0, \underbrace{0, ..., 0}_{r_{\mathcal{H}}-2}, 1\right) - P_B\left(z_0, \underbrace{0, ..., 0}_{r_{\mathcal{H}}-1}\right)$ allows to retrieve the full DNF which stores hyperedges of size $r_{\mathcal{H}}-1$.

$P_{B\,r_{\mathcal{H}}-j}(z_0) = P_B\left(z_0, \underbrace{0, ..., 0}_{r_{\mathcal{H}}-j-1}, \underbrace{1, ..., 1}_{j}\right) - P_B\left(z_0, \underbrace{0, ..., 0}_{r_{\mathcal{H}}-j-2}, \underbrace{1, ..., 1}_{j-1}\right)$ allows to retrieve the full DNF which stores hyperedges of size $r_{\mathcal{H}}-j$.

Stopping at $P_{B\,1}(z_0) = P_B\left(z_0, \underbrace{1, ..., 1}_{r_{\mathcal{H}}-1}\right) - P_B\left(z_0, 0, \underbrace{1, ..., 1}_{r_{\mathcal{H}}-2}\right)$ allows to retrieve the full DNF which stores hyperedges of size 1.

Considering the adjacency matrix of [ZHS07] of this unweighted hypergraph, it holds that $\boldsymbol{w}_0^\top A \boldsymbol{w}_0$ can be considered as a Boolean homogeneous polynomial in full disjunctive form where the conjunctive terms are composed of only two variables. This shows, if necessary, that this approach is a pairwise approximation of the e-adjacency tensor.

The homogeneous polynomial attached to [BCM17] tensor can be mapped to a Boolean polynomial function by considering the same term elements with coefficient being 1 when the original homogeneous polynomial has a non-zero coefficient and, 0 otherwise. Nonetheless, this Boolean function is no more in DNF. Reducing it to DNF yields to the expression of $P_B\left(\boldsymbol{z}_0, \underbrace{1, ..., 1}_{r_{\mathcal{H}}-1}\right)$.

E.1.5.3. First results on spectral analysis of e-adjacency tensor

Let $\mathcal{H} = (V, E)$ be a general hypergraph of e-adjacency tensor $\mathcal{A}_{\mathcal{H}}$ of CHR $\boldsymbol{A_{\mathcal{H}}} = \left(a_{i_1 \ldots i_{r_{\mathcal{H}}}}\right)$.

In the e-adjacency tensor $\boldsymbol{A_{\mathcal{H}}}$ built, the diagonal entries are equal to zero. As all elements of $\boldsymbol{A_{\mathcal{H}}}$ are all non-negative real numbers and as we have shown that:

$$\forall i \in [\![n + r_{\mathcal{H}} - 1]\!] : \sum_{\substack{i_2, ..., i_{r_{\mathcal{H}}} \in [\![n+r_{\mathcal{H}}-1]\!] \\ \delta_{ii_2 \ldots i_{r_{\mathcal{H}}}} = 0}} a_{ii_2 \ldots i_{r_{\mathcal{H}}}} = d_i.$$

It follows:

Theorem E.1. *The e-adjacency tensor $\mathcal{A}_{\mathcal{H}}$ of CHR $\boldsymbol{A}_{\boldsymbol{\mathcal{H}}} = \left(a_{i_1 \dots i_{r_{\mathcal{H}}}}\right)$ of a general hypergraph $\mathcal{H} = (V, E)$ has its eigenvalues λ such that:*

$$|\lambda| \leqslant \max\left(\Delta, \Delta^\star\right) \tag{E.6}$$

where: $\Delta \triangleq \max_{i \in [\![n]\!]} (d_i)$ *and* $\Delta^\star \triangleq \max_{i \in [\![r_{\mathcal{H}}-1]\!]} (d_{n+i})$.

Proof. From B.2, we can write, as $a_{i \dots i} = 0$ and $a_{i i_2 \dots i_{r_{\mathcal{H}}}}$ are non-negative numbers, that for all λ it holds:

$$|\lambda| \leqslant \sum_{\substack{i_2, \dots, i_{r_{\mathcal{H}}} \in [\![n + r_{\mathcal{H}} - 1]\!] \\ \delta_{i i_2 \dots i_{r_{\mathcal{H}}}} = 0}} a_{i i_2 \dots i_{r_{\mathcal{H}}}}$$

and thus writing $\Delta = \max_{i \in [\![n]\!]} (d_i)$ and $\Delta^\star = \max_{i \in [\![r_{\mathcal{H}}-1]\!]} (d_{n+i})$, it holds: $|\lambda| \leqslant \max\left(\Delta, \Delta^\star\right)$.

□

Proposition E.7. *Let $\mathcal{H}$ be a r-regular r-uniform hypergraph. Then this maximum is reached.*

Proof. In this case:

$$\forall i \in [\![n]\!], \, d_i = \Delta = r$$

and:

$$\Delta^\star = 0,$$

also:

$$\max\left(\Delta, \Delta^\star\right) = r.$$

Considering $\lambda = r$ and the vector $\mathbf{1}$ which components are only 1, $(r, \mathbf{1})$ is an eigenpair of $\boldsymbol{A}_{\boldsymbol{\mathcal{H}}}$ as for all $i \in [\![n]\!]$:

$$\begin{aligned} & \sum_{i_2, \dots, i_{r_{\mathcal{H}}} \in [\![n + r_{\mathcal{H}} - 1]\!]} a_{i i_2 \dots i_{r_{\mathcal{H}}}} x_{i_2} \dots x_{i_{r_{\mathcal{H}}}} &&= \lambda x_i^{r_{\mathcal{H}} - 1} \\ \Leftrightarrow \quad & \sum_{i_2, \dots, i_{r_{\mathcal{H}}} \in [\![n + r_{\mathcal{H}} - 1]\!]} a_{i i_2 \dots i_{r_{\mathcal{H}}}} &&= r \\ \Leftrightarrow \quad & d_i &&= r. \end{aligned}$$

□

Remark E.1.2. *We see that this bound includes $\Delta^\star$ which can be close to the number of hyperedges, for instance where the hyperedges would be constituted of only one vertex per hyperedge except one hyperedge with $r_{\mathcal{H}} \neq 1$ vertices in it.*

E.1.6. Evaluation

We have gathered in Table E.1 some key features of both the e-adjacency tensor proposed by [BCM17]—written $\mathcal{B}_{\mathcal{H}}$—, the new e-adjacency tensor proposed by [SZB19]—written $\mathcal{S}_{\mathcal{H}}$—and the one constructed in this Thesis—written $\mathcal{A}_{\mathcal{H}}$. The constructed tensor has same order than the two others. The dimension of $\mathcal{A}_{\mathcal{H}}$ is $r_{\mathcal{H}}-1$ bigger than $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ ($n-1$ in the worst case). The way $\mathcal{A}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ are built uses potentially $\frac{(n+r_{\mathcal{H}}-1)!}{(n-1)!n^{r_{\mathcal{H}}}}$ times less elements than for $\mathcal{B}_{\mathcal{H}}$—$O\left(\frac{n!}{n^n}\right)$ in the worst case. The number of non-zero elements filled in $\mathcal{A}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ for a given hypergraph is $\left(1+\frac{r_{\mathcal{H}}-1}{n}\right)^{r_{\mathcal{H}}}$ times the number of elements of $\mathcal{B}_{\mathcal{H}}$ ($O\left(4^n\right)$ times in the worst case). But, the number of elements to be filled to have full description of a hyperedge of size s by permutation of indices due to the symmetry of the tensor is only 1 in the case of $\mathcal{A}_H$, which is $\frac{1}{p_s\left(r_{\mathcal{H}}\right)}$ times less than for a hyperedge stored in $\mathcal{B}_{\mathcal{H}}$ and $\frac{1}{s}$ for a hyperedge stored in $\mathcal{S}_{\mathcal{H}}$. The minimum number of elements needed to be described the other being obtained by permutation is $\frac{1}{|E|}\sum\limits_{s\in[\![r_{\mathcal{H}}]\!]} p_s\left(r_{\mathcal{H}}\right)|E_s|$ bigger for $\mathcal{B}_{\mathcal{H}}$ than for $\mathcal{A}_{\mathcal{H}}$ and $\frac{1}{|E|}\sum\limits_{s\in[\![r_{\mathcal{H}}]\!]} s\,|E_s|$ bigger for $S_{\mathcal{H}}$ than for $\mathcal{A}_{\mathcal{H}}$. Moreover, the value of the elements in $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ varies with the cardinality of the hyperedge; in $\mathcal{A}_{\mathcal{H}}$, any element has same value. Both tensors allow the reconstruction of the original hypergraph; for $\mathcal{B}_{\mathcal{H}}$, it requires at least $p_s\left(r_{\mathcal{H}}\right)$ checks per hyperedge and s in $\mathcal{S}_{\mathcal{H}}$ as for $\mathcal{A}_{\mathcal{H}}$ it requires only one element per hyperedge.

In both cases, node degrees can be deduced from the e-adjacency tensor. $\mathcal{A}_{\mathcal{H}}$ allows the retrieval of the structure of the hypergraph in term of hyperedges cardinality which is not straightforward in the case of $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$.

The fact that $\mathcal{A}_{\mathcal{H}}$ can be interpreted in terms of hypergraph uniformisation is possible as it is the e-adjacency tensor of the V_s-layered uniform hypergraph $\widehat{\mathcal{H}_{\widehat{w}}}$ obtained from $\mathcal{H}$. $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{S}_{\mathcal{H}}$ are not interpretable in term of hypergraph uniformisation, as hyperedges do not allow repetition of vertices; we will have to wait the introduction of hb-graphs to get a meaningful interpretation—Chapter 4.

E.1.7. Further comments

The importance of defining properly the concept of adjacency in a hypergraph has helped us to build a proper e-adjacency tensor in a way that allows to contain important information on the structure of the hypergraph. This work contributes to give a methodology to build a uniform hypergraph and hence a cubical symmetric tensor from the different layers of uniform hypergraphs contained in a hypergraph. The built tensor allows to reconstruct with no ambiguity the original hypergraph. Nonetheless, first results on spectral analysis show difficulties to use the tensor built as the additional vertices inflate the spectral radius bound. The uniformisation process is a strong basis for building alternative proposals.

	$\mathcal{B}_\mathcal{H}$	$\mathcal{S}_\mathcal{H}$	$\mathcal{A}_\mathcal{H}$
CHR	$\boldsymbol{B_\mathcal{H}}$	$\boldsymbol{S_\mathcal{H}}$	$\boldsymbol{A_\mathcal{H}}$
Order	$r_\mathcal{H}$	$r_\mathcal{H}$	$r_\mathcal{H}$
Dimension	n	n	$n + r_\mathcal{H} - 1$
Total number of elements	$n^{r_\mathcal{H}}$	$n^{r_\mathcal{H}}$	$(n + r_\mathcal{H} - 1)^{r_\mathcal{H}}$
Total number of elements potentially used by the way the tensor is build	$n^{r_\mathcal{H}}$	$n^{r_\mathcal{H}}$	$\frac{(n + r_\mathcal{H} - 1)!}{(n-1)!}$
Number of non-zero elements for a given hypergraph	$\sum_{s \in [\![r_\mathcal{H}]\!]} \alpha_s \lvert E_s \rvert$ with $\alpha_s = p_s(r_\mathcal{H}) \frac{r_\mathcal{H}!}{k_1!...k_s!}$	$\sum_{s \in [\![r_\mathcal{H}]\!]} s! \lvert E_s \rvert$	$r_\mathcal{H}! \lvert E \rvert$
Number of repeated elements per hyperedge of size s	$\frac{r_\mathcal{H}!}{k_1!...k_s!}$	$s!$	$r_\mathcal{H}!$
Number of elements to be filled per hyperedge of size s before permutation	Varying: $p_s(r_\mathcal{H})$	Varying s if prefix is considered as non-permuting part	Constant: 1
Number of elements to be described to derived the tensor by permutation of indices	$\sum_{s \in [\![r_\mathcal{H}]\!]} p_s(r_\mathcal{H}) \lvert E_s \rvert$	$\sum_{s \in [\![r_\mathcal{H}]\!]} s \lvert E_s \rvert$	$\lvert E \rvert$
Value of elements of a hyperedge	Varying $\frac{s}{\alpha_s}$	Varying $\frac{1}{(s-1)!}$	Constant $\frac{1}{(r_\mathcal{H}-1)!}$
Symmetric	Yes	No	Yes
Reconstructivity	Need computation of duplicated vertices	Need computation of duplicated vertices	Straightforward: delete special vertices
Nodes degree	Yes	Yes	Yes
Spectral analysis	Yes	Yes	Special vertices increase the amplitude of the bounds
Interpretability of the tensor in term of hypergraph uniformisation	No	No	Yes

Table E.1.: Evaluation of the e-adjacency tensor.

$\mathcal{B}_\mathcal{H}$ is the adjacency tensor defined in [BCM17].
$S_\mathcal{H}$ is the adjacency tensor defined in [SZB19].
$\mathcal{A}_\mathcal{H}$ is the layered e-adjacency tensor as defined in this Thesis.

E.2. Examples of hypermatrices for general hypergraphs

E.2.1. An example of the e-adjacency hypermatrix proposed by [BCM17]

For instance, with the hypergraph $\mathcal{H} = (V, E)$ with $V = \{v_1, v_2, v_3, v_4, v_5\}$ with $E = \{e_1, e_2, e_3\}$ where: $e_1 = \{v_1, v_2, v_3, v_4\}$, $e_2 = \{v_2, v_5\}$ and $e_3 = \{v_3\}$ has as e-adjacency hypermatrix following the model proposed by [BCM17] $\boldsymbol{B}_{\mathcal{H}} = (b_{i_1 i_2 i_3 i_4})_{\forall j \in [\![4]\!] : i_j \in [\![5]\!]}$. Considering all permutations of the indices $\sigma \in \mathcal{S}_4$, we have:

- e_1 stored in all the elements: $b_{\sigma(1\,2\,3\,4)}$ and of value $\dfrac{1}{6}$;
- e_2 stored in all the elements: $b_{\sigma(2\,5\,5\,5)}$, $b_{\sigma(2\,2\,5\,5)}$ and $b_{\sigma(2\,2\,2\,5)}$; the value of these elements is $\dfrac{2}{\alpha_2} = \dfrac{1}{7}$ as $\alpha_2 = 2 \times \dfrac{4!}{1!3!} + \dfrac{4!}{2!2!} = 14$;
- e_3 stored in $b_{3\,3\,3\,3} = 1$.
- The other elements of $\boldsymbol{B}_{\mathcal{H}}$ are equal to 0.

E.2.2. An example of layered e-adjacency hypermatrix for general hypergraph

The example given illustrates the layered e-adjacency tensor for general hypergraphs as it is exposed in Appendix E.1.

Example E.2.1. *Given the following hypergraph:* $\mathcal{H} = (V, E)$ *where:* $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ *and* $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ *with:* $e_1 = \{v_1, v_2, v_3\}$, $e_2 = \{v_1, v_2, v_7\}$, $e_3 = \{v_6, v_7\}$, $e_4 = \{v_5\}$, $e_5 = \{v_4\}$, $e_6 = \{v_3, v_4\}$ *and* $e_7 = \{v_4, v_7\}$.

This hypergraph $\mathcal{H}$ is drawn in Figure E.1.

The layers of $\mathcal{H}$ are:

- $\mathcal{H}_1 = (V, \{e_4, e_5\})$ with the associated unnormalized tensor:

  $$\boldsymbol{A}_{\mathbf{1}\,\mathrm{raw}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

  and associated homogeneous polynomial:

  $$P_1(z_0) = z_4 + z_5.$$

  More generally, the version with a normalized tensor is:

  $$P_1(z_0) = a_{(1)\,4} z_4 + a_{(1)\,5} z_5.$$

- $\mathcal{H}_2 = (V, \{e_3, e_6, e_7\})$ with the associated unnormalized tensor:

  $$\boldsymbol{A}_{\mathbf{2}\,\mathrm{raw}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

and associated homogeneous polynomial:

$$P_2(z_0) = 2z_3z_4 + 2z_6z_7 + 2z_4z_7.$$

More generally, the version with a normalized tensor is:

$$P_2(z_0) = 2!a_{(2)\,3\,4}z_3z_4 + 2!a_{(2)\,6\,7}z_6z_7 + 2!a_{(2)\,4\,7}z_4z_7.$$

- $\mathcal{H}_3 = (V, \{e_1, e_2\})$ with the associated unnormalized tensor:

$$\mathbf{A_3}_{\text{raw}} = \left[\begin{array}{ccccccc|ccccccc|ccccccc|c|ccccccc}
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right]$$

and associated homogeneous polynomial:

$$P_3(z_0) = 3!z_1z_2z_3 + 3!z_1z_2z_7.$$

More generally, the version with a normalized tensor is:

$$P_3(z_0) = 3!a_{(3)\,1\,2\,3}z_1z_2z_3 + 3!a_{(3)\,1\,2\,7}z_1z_2z_7.$$

The iterative process of homogenization is then the following using the degree-normalized adjacency tensor $\mathcal{A}_k = \dfrac{1}{(k-1)!}\mathcal{A}_{k\,\text{raw}}$ and the normalizing coefficients $c_k = \dfrac{k_{\max}}{k}$, with $k_{\max} = 3$:

- $R_1(\boldsymbol{z_0}) = \dfrac{3}{1}P_1(z_0)$;
- $R_2(\boldsymbol{z_1}) = R_1(z_0)\,y_1 + \dfrac{3}{2}P_2(z_0)$;
- $R_3(\boldsymbol{z_2}) = R_2(z_1)\,y_2 + \dfrac{3}{3}P_3(z_0)$.

Hence:

$$\begin{aligned}
R_3(\boldsymbol{z_2}) \;=\; & 3\left(a_{(1)\,4}z_4 + a_{(1)\,5}z_5\right)y_1y_2 \\
& + \frac{3}{2}\times 2!\left(a_{(2)\,3\,4}z_3z_4 + a_{(2)\,6\,7}z_6z_7 + a_{(2)\,4\,7}z_4z_7\right)y_2 \\
& + 3!\left(a_{(3)\,1\,2\,3}z_1z_2z_3 + a_{(3)\,1\,2\,7}z_1z_2z_7\right).
\end{aligned}$$

Therefore, the e-adjacency tensor of $\mathcal{H}$ is obtained by writing the corresponding symmetric cubical tensor of order 3 and dimension 9, described by: $r_{489} = r_{589} = r_{349} = r_{679} = r_{479} = r_{123} = r_{127} = \dfrac{1}{2}$. The other remaining not null elements are obtained by permutation on the indices.

Finding the degree of one vertex from the tensor is easily achievable; for instance $\deg(v_4) = 2!\,(r_{489} + r_{349} + r_{479}) = 3$.

F. Used Fraktur fonts

Fraktur font	𝔄	𝔅	ℭ	𝔇	𝔈	𝔢	𝔉	ℌ	𝔥	𝔍	𝔓	ℜ	𝔖	𝔗	𝔙	𝔳
Classical font	A	B	C	D	E	e	F	H	h	I	P	R	S	T	V	v