EXCHANGE-BASED DIFFUSION IN HB-GRAPHS: HIGHLIGHTING COMPLEX RELATIONSHIPS

A PREPRINT

Xavier Ouvrard
University of Geneva, CERN
CERN, 1 Esplanade des Particules, CH-1211 Geneva 23
xavier.ouvrard@cern.ch

Jean-Marie Le Goff
CERN
CERN, 1 Esplanade des Particules, CH-1211 Geneva 23
jean-marie.le.goff@cern.ch

Stéphane Marchand-Maillet
CUI Batelle A, University of Geneva, Route de Drize, 7, CH-1227 Carouge
stephane.marchand-maillet@unige.ch

May 30, 2019

ABSTRACT

Most networks tend to show complex and multiple relationships between entities. Networks are usually modeled by graphs or hypergraphs; nonetheless a given entity can occur many times in a relationship: this brings the need to deal with multisets instead of sets or simple edges. Diffusion processes are useful to highlight interesting parts of a network: they usually start with a stroke at one vertex and diffuse throughout the network to reach a uniform distribution. Several iterations of the process are required prior to reaching a stable solution. We propose an alternative solution to highlighting the main components of a network using a diffusion process based on exchanges: it is an iterative two-phase step exchange process. This process allows to evaluate the importance not only of the vertices but also of the regrouping level. To model the diffusion process, we extend the concept of hypergraphs that are families of sets to families of multisets, that we call hb-graphs.

Keywords exchange · diffusion · multiset · hyperbag-graph · information retrieval · ranking

This article is an extended version of [1] (pre-printed in arXiv:1809.00190v1): the text of the extended version is in blue, the text in black is the one of [1]. All the figures except Figure 2 have been either modified or added in this extended version to take into account the new developments. The contributions of this extended version are: the proofs of conservation and convergence of the extracted sequences of the diffusion process, as well as the illustration of the speed of convergence and comparison to classical and modified random walks; the algorithms of the exchange-based diffusion and the modified random walk; the application to a use case based on Arxiv publications.

1 Introduction

Many relationships are more than pairwise relations: entities are often grouped into sets, corresponding to n-adic relationships. Each of these sets can be viewed as a collaboration between entities. Hypergraphs naturally represent n-adic relations. It has been shown that facets of an information space can be modeled by hypergraphs [2]: each facet corresponds to a type of metadata. The different facets are then linked by reference data attached to hyperedges within that facet. The step forward is to highlight important information contained in those facets. This is commonly

achieved in hypergraphs using random walks [3, 4]. Reference [4] shows that the weighting of vertices at the level of the hyperedges in a hypergraph provides better information retrieval. These two approaches - [3, 4] - mainly focus on vertices; but as hyperedges are linked to references that can be used as pivots in between the different facets [5, 2], it is also interesting to highlight important hyperedges. For instance, in a document database, different metadata can be used to label authors, author keywords, processed keywords, categories, added tags: the pivots between the different facets of this information space correspond to the documents themselves. In the specific case of tags, it can be important to have weights attached to them if the users are able to attach tags to documents.

Hyperedge-based weighting of vertices is easier to achieve through multisets: multisets store information on multiplicity of elements. We use multisets family over a set of vertices, called hyper-bag-graph - hb-graph for short - as an extension of hypergraphs. Hb-graph multisets play the role of the hyperedges in hypergraph: they are called hb-edges.

We want to address the following research questions: "Can we find a network model and a diffusion process that not only rank vertices but also rank hb-edges in hb-graphs?". We develop an iterative exchange approach in hb-graphs with two-phase steps that allows to extract information not only at the vertex level but also at the hb-edge level.

We validate our approach by using randomly generated hb-graphs. The hb-graph visualisation highlights not only vertices but also hb-edges using the exchange process. We show that the exchange-based diffusion process provides proper coloring of vertices with high connectivity and highlights hb-edges with a normalisation approach - allowing small hb-edges to have a chance to be highlighted. We apply this approach to process the metadata contained in the results retrieved by querying Arxiv through its API in order to visualize the results: we will show how it can be used to allow further query expansion.

This paper contributes to present an exchange-based diffusion process that enables not only the ranking of vertices but also of hb-edges. It formalizes exchanges by using hb-graphs that can naturally cope with elements multiplicity. It contributes also to a novel visualisation of this kind of network depicted in each facet of the information space.

In Section II, the mathematical background and the related work is given. The construction of the formalisation of the exchange process is presented in Section IV. Results and evaluation are given in Section V and future work and conclusion are addressed in Section VI.

2 Mathematical background and Related work

2.1 Hypergraphs

A hypergraph $\mathcal{H} = (V, E)$ over a finite set of vertices $V = \{v_1\,;\, v_2;\, ...\,;\, v_n\}$ is defined in [6] as a family of hyperedges $E = (e_1, e_2, ..., e_p)$ where each hyperedge is a non-empty subset of V and such that $\bigcup_{i=1}^{p} e_i = V$. A hypergraph $\mathcal{H}_w = (V, E, w_e)$ is said edge-weighted if there exists an application $w_e : E \to \mathbb{R}^{+*}$.

In a weighted hypergraph the degree $\deg(v_i)$ of a vertex v_i is defined as:

$$d_i = \deg(v_i) = \sum_{e_j \in E : v_i \in e_j} w_e(e_j).$$

The volume of $S \subseteq V$ is defined as:

$$\text{vol}(S) = \sum_{v_i \in S} \deg(v_i).$$

The incident matrix of a hypergraph is the matrix $H = [h_{ij}]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant p}}$ of $M_{n \times p}(\{0\,;\,1\})$, where $h_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases}$.

Random walks are largely used to evaluate the importance of vertices in hypergraphs. In [3], 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:

$$p(v_{i_1}, v_{i_2}) = \sum_{j=1}^{p} w_e(e_j) \frac{h_{i_1 j}}{d_{i_1}} \times \frac{h_{i_2 j}}{\delta_j},$$

where $\delta_j = \deg(e_j)$, $1 \leqslant j \leqslant p$ is the degree of a hyperedge defined in [3] as its cardinality. This random walk has a stationary state which is shown to be $\pi = (\pi_i)_{1 \leqslant i \leqslant n}$ with $\pi_i = \dfrac{d_i}{\text{vol}V}$ for $1 \leqslant i \leqslant n$ [7]. This process differs from

the one we propose: our diffusion process is done by successive steps from a random initial vertex on vertices and hyperedges.

Reference [4] defines a random walk for weighted hypergraphs using weight functions both for hyperedges and vertices: a vector of weights is built for each vertex 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. Reference [4] shows that a random walk on the (double-) weighted hypergraph enables vertex ranking with higher precision than random walks using unweighted vertices. This process differs again 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. Reference [8] uses random walks in hypergraph for image matching. Reference [9] builds higher order random walks in hypergraph and constructs a generalised Laplacian attached to the graphs generated from their random walks.

Hypergraphs fit to model multi-adicity in structures where the traditional pairwise relationship of graphs is insufficient: they are used in many areas such as social networks in particular in collaboration networks - [10, 11] -, co-author networks - [12], [13] -, chemical reactions - [14] - , genome - [15] -, VLSI design - [16] - and other applications. Hypergraphs are also used in information retrieval for different purposes such as query formulation in text retrieval [17], in music recommandation [18],... Several applications of hypergraphs exist based on the diffusion process firstly developped by [3]. [19] uses [3] for 3D-object retrieval and recognition by building multiple hypergraphs of objects based on their 2D-views that are analysed using the same approach. In [20], multiple hypergraphs are constructed to characterize the complex relations between landmark images and are gathered into a multimodal hypergraph that allows the integration of heterogeneous sources providing content-based visual landmark searches. Hypergraphs are also used in multi-feature indexing to help image retrieval [21]. For each image, a hyperedge gathers the first n most similar images based on different features. Hyperedges are weighted by average similarity. A spectral clustering algorithm is then applied to divide the dataset into k sub-hypergraphs. A random walk on these sub-hypergraphs allows to retrieve significant images: they are used to build a new inverted index, useful to query images. In [22], a joint-hypergraph learning is achieved for image retrieval, combining efficiently a semantic hypergraph based on image tags with a visual hypergraph based on image features.

2.2 Multisets

Multisets - also known as bags or msets - have a long use in many domains. But before developping their use in different domains, we firstly give main definitions on multisets mainly based on [23].

A **multiset** is a pair $A_m = (A, m)$ where A is a set of distinct objects and m is an application from A to $\mathbb{W} \subseteq \mathbb{R}$ or $\mathbb{N}$. A is called the **universe** of the multiset A_m, m is called the **multiplicity function** of the multiset A_m. $A_m^\star = \{x \in A : m(x) \neq 0\}$ is called the **support** of A_m. The elements of the support of an mset are called its **generators**.

A multiset where $\mathbb{W} \subseteq \mathbb{N}$ is called a **natural multiset**.

The **m-cardinality** of A_m written $\#_m A_m$ is defined as:

$$\#_m A_m = \sum_{x \in A} m(x).$$

Several notations of msets exist. Among the common notations mentioned in [24], we note in this article a mset A_m of universe $A = \{x_i : i \in [\![n]\!]\}$ by:

$$A_m = \{x_i^{m_i} : i \in [\![n]\!]\}$$

where $m_i = m(x_i)$.

If A_m is a natural multiset, another notation of A_m similar to an unordered list is:

$$\left\{\left\{\underbrace{x_1, \ldots, x_1}_{m_1 \text{ times}}, \ldots, \underbrace{x_n, \ldots, x_n}_{m_n \text{ times}}\right\}\right\}.$$

Considering $\mathcal{A} = \Omega_{m_\mathcal{A}}$ and $\mathcal{B} = \Omega_{m_\mathcal{B}}$ two msets on the same universe Ω, we define the empty mset, written $\emptyset_\Omega$ the set of empty support on the universe Ω. $\mathcal{A}$ is said to be **included** in $\mathcal{B}$ - written $\mathcal{A} \subseteq \mathcal{B}$ - if for all $x \in \Omega$: $m_\mathcal{A}(x) \leqslant m_\mathcal{B}(x)$.

In this case, $\mathcal{A}$ is called a **submset** of $\mathcal{B}$. The power multiset of A, written $\widetilde{\mathcal{P}}(A)$, is the multiset of all submsets of A. Different operations can be defined on multisets of same universe as union, intersection, sum, complementation and difference: for details one can refer to [23].

Multisets, under the appellation bag, appear in different domains such as text modeling, image description and audio [25]. In text representation, bag of words have been first introduced in [26]: 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 [27]. More recently bag of words have been used successfully for translation by neural nets as a target for the translation as a sentence can be translated in many different ways [28]. In [29], 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 that allows the description of images as bags of these features. Since their introduction in [30], many applications have been realized: in visual categorization [31], in image classification and filtering [32], in image annotation [33], in action recognition [34], in land-use scene classification [35], in identifying mild traumatic brain injuries [36] and in word image retrieval [37].

Bags of concepts are an extension of bags of words to successive concepts in a text [38]. A recent extension of these concepts is given in [39] where bag of graphs are introduced to encode in graphs the local structure of a digital object: bags of graphs are declined into bags of singleton graphs and bags of visual graphs. Using the hb-graphs as we propose in this article will allow to extend this approach, by taking advantage of multi-adicity and also of the multiplicity of vertices specific to each hb-edge.

2.3 Hb-graphs

Hb-graphs are introduced in [24]. A **hb-graph** is a family of multisets with the same universe V and with 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 article a hb-graph $\mathcal{H} = (V, E)$, with $V = \{v_1, ..., v_n\}$ and $E = \{e_1, ..., e_p\}$ the family of its hb-edges.

Each hb-edge $e_i \in E$ has V as universe and a multiplicity function associated to it: $m_{e_i} : V \to \mathbb{W}$ where $\mathbb{W} \subset \mathbb{R}^+$. For a general hb-graph, each hb-edge has to be seen as a weighted system of vertices, where the weights of each vertex are hb-edge dependent.

A hb-graph where the multiplicity range of each hb-edge is a subset of $\mathbb{N}$ is called a **natural hb-graph**. A **hypergraph** is a natural hb-graph where the hb-edges have multiplicity one for every vertex of their support.

The **order** of a hb-graph $\mathcal{H}$ - written $O(\mathcal{H})$ - is:

$$O(\mathcal{H}) = \sum_{v_i \in V} \max_{e_j \in E} \left(m_{e_j}(v_i) \right).$$

In a natural hb-graph, the order corresponds to the number of copies needed to generate the copy hypergraph of the hb-graph.

The **m-size** of a hb-graph $\mathcal{H}$ - written $s_m(\mathcal{H})$ - is:

$$s_m(\mathcal{H}) = \sum_{e_j \in E} \sum_{v_i \in e_j^\star} m_{e_j}(v_i).$$

In a natural hb-graph the m-size corresponds to the sum of the m-cardinalities of the hb-edges of the hb-graph.

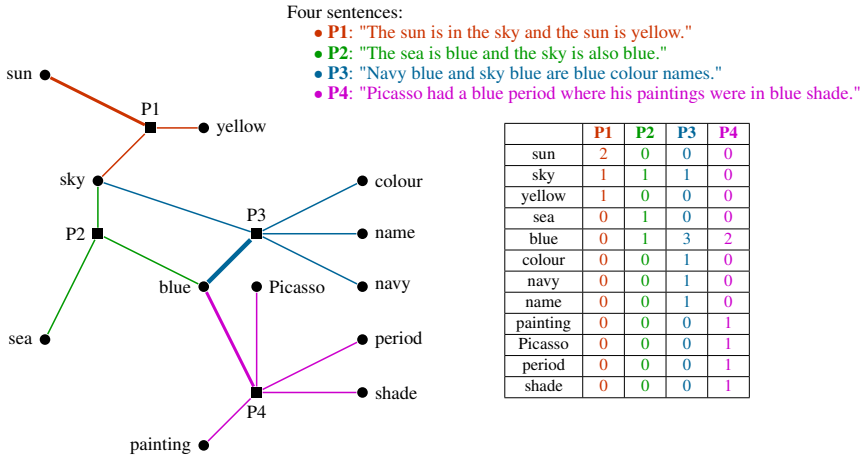
The **support hypergraph** of a hb-graph $\mathcal{H} = (V, E)$ is the hypergraph whose vertices are the ones of the hb-graph and whose hyperedges are the support of the hb-edges in a one-to-one way. We write it $\underline{\mathcal{H}} = (V, \underline{E})$, where $\underline{E} = \{e^\star : e \in E\}$.

The **hb-star** of a vertex $v \in V$ is the multiset - written $H(v)$ and abusively writing e_i, $1 \leqslant i \leqslant p$ for designating the elements of the universe of $H(v)$ corresponding to the hb-edges of $\mathcal{H}$ of same name - defined as:

$$H(v_i) = \left\{ e_j^{m_{e_j}(v_i)} \ : \ \forall 1 \leqslant j \leqslant p \text{: } e_j \in E \wedge v_i \in e_j^* \right\}.$$

The **m-degree of a vertex** $v_i \in V$ of a hb-graph $\mathcal{H}$ - written $\deg_m(v_i) = d_m(v_i)$ - is defined as:

$$\deg_m(v_i) = \#_m H(v_i).$$



	P1	P2	P3	P4
sun	2	0	0	0
sky	1	1	1	0
yellow	1	0	0	0
sea	0	1	0	0
blue	0	1	3	2
colour	0	0	1	0
navy	0	0	1	0
name	0	0	1	0
painting	0	0	0	1
Picasso	0	0	0	1
period	0	0	0	1
shade	0	0	0	1

Figure 1: An example of hb-graphs: four sentences and their associated bag of words with removed stop words and the incidence matrix of the hb-graph.

We have:

$$\sum_{v_i \in V} \deg_m (v_i) = s_m (\mathcal{H}) .$$

The **degree of a vertex** $v \in V$ of a hb-graph $\mathcal{H}$ - written $\deg (v) = d(v)$ - corresponds to the degree of this vertex in the support hypergraph $\underline{\mathcal{H}}$.

The matrix $H = [m_j (v_i)]_{\substack{1 \leqslant i \leqslant n \\ 1 \leqslant j \leqslant p}}$ is called the **incident matrix** of the hb-graph $\mathcal{H}$.

A **weighted hb-graph** $\mathcal{H}_w = (V, E, w_e)$ is a hb-graph $\mathcal{H} = (V, E)$ where the hb-edges are weighted by $w_e : E \to \mathbb{R}^{+*}$. An unweighted hb-graph is then a weighted hb-graph with $w_e (e_j) = 1$ for all $e_j \in E$.

A **strict m-path** $v_0 e_1 v_1 ... e_s v_s$ in a hb-graph from a vertex u to a vertex w is a vertex / hb-edge alternation with hb-edges e_1 to e_s and vertices v_0 to v_s such that $v_0 = u$, $v_s = w$, $u \in e_1$ and $w \in e_s$ and that for all $1 \leqslant i \leqslant s - 1$, $v_i \in e_i \cap e_{i+1}$.

A strict m-path $v_0 e_1 v_1 ... e_s v_s$ in a hb-graph corresponds to a unique path in the hb-graph support hypergraph called the **support path**. In this article we abusively call it a path of the hb-graph. The **length of a path** corresponds to the number of hb-edges it is going through.

Representations of hb-graphs can be achieved either by using sub-mset representations or by using edge representations. In the edge representation, an extra-node is added per hb-edge and the thickness of the link between the extra-node of a hb-edge and the vertices in the support of the hb-edge is made proportional to the multiplicity of vertices. Except in Figure 1 where we use this representation, in this article we use a simplified representation corresponding to the extra-vertex representation of the support hypergraph of the hb-graph: an extra-vertex is added for each hb-edge and the links with the vertices in the support of the hb-edges are all represented with the same thickness. More details on these representations can be found in [24].

We give in Figure 1 an example of the representation of a hb-graph of keywords extracted from sentences in which stop words have been removed. The number of occurences of the words differs from one sentence to an other: it is given as a multiplicity that is specific to the corresponding hb-edge representing the sentence. The universe of the hb-graph is the set of words where the stop words has been removed.

3 Exchange-based diffusion in hb-graphs

Diffusion processes lead to 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 structure

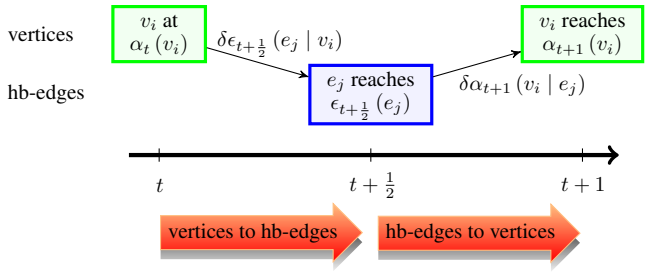


Figure 2: Diffusion by exchange: principle

of the network itself. Several random walks with random choices of the starting vertex are needed to achieve ranking by averaging. Moreover to avoid loops, teleportation of vertices is needed.

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

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

$$\alpha_t : \left\{ \begin{array}{c} V \to \mathbb{R} \\ v_i \mapsto \alpha_t(v_i) \end{array} \right. .$$

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

$$\epsilon_t : \left\{ \begin{array}{c} E \to \mathbb{R} \\ e_j \mapsto \epsilon_t(e_j) \end{array} \right. .$$

We write $P_{V,t} = (\alpha_t(v_i))_{1 \leqslant i \leqslant n}$ the row state vector of the vertices at time t and $P_{E,t} = (\epsilon_t(e_j))_{1 \leqslant j \leqslant p}$ 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, hence, 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 $e_j \in E$, $\epsilon(e_j) = 0$.

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

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

During the first phase between time t and $t + \dfrac{1}{2}$, each vertex v_i of the hb-graph shares the value $\alpha_t(v_i)$ it holds 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 $\dfrac{m_j(v_i)}{\deg_m(v_i)}$, which corresponds to the ratio of multiplicity of the vertex v_i due to the hb-edge e_j over the total m-degree of hb-edges that contains v_i in their support.

In a weighted hb-graph, each hb-edge has a weight $w_e(e_j)$. The value $\alpha_t(v_i)$ of a vertex v_i has to be shared by taking not only the multiplicity of the vertices in the hb-edge but also the weight $w_e(e_j)$ of a hb-edge e_j into account.

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

$$w_E = (w_e(e_j))^{\top}_{1 \leqslant j \leqslant p}.$$

We also consider the weight diagonal matrix

$$W_E = \text{diag}\left((w_e(e_j))_{1 \leqslant j \leqslant p} \right).$$

We introduce the weighted m-degree matrix:

$$D_{w,V} = \text{diag}\left(\left(d_{w,v_i}\right)_{1\leqslant i\leqslant n}\right) = \text{diag}\left(Hw_E\right).$$

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

$$d_{w,v_i} = \deg_{w,m}\left(v_i\right) = \sum_{1\leqslant j\leqslant p} m_j\left(v_i\right) w_e\left(e_j\right).$$

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

$$\delta\epsilon_{t+\frac{1}{2}}\left(e_j \mid v_i\right) = \frac{m_j\left(v_i\right) w_e\left(e_j\right)}{d_{w,v_i}} \alpha_t\left(v_i\right).$$

It corresponds to the ratio of weighted multiplicity of the vertex v_i in e_j over the total weighted m-degree of the hb-edges where v_i is in the support.

We remark that if $v_i \notin e_j^\star$: $\delta\epsilon_{t+\frac{1}{2}}\left(e_j \mid v_i\right) = 0$.

And the value $\epsilon_{t+\frac{1}{2}}\left(e_j\right)$ is calculated by summing over the vertex set:

$$\epsilon_{t+\frac{1}{2}}\left(e_j\right) = \sum_{i=1}^{n} \delta\epsilon_{t+\frac{1}{2}}\left(e_j \mid v_i\right).$$

Hence, we obtain:

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

The value given to the hb-edges is subtracted to the value of the corresponding vertex, hence for all $1 \leqslant i \leqslant n$:

$$\alpha_{t+\frac{1}{2}}\left(v_i\right) = \alpha_t\left(v_i\right) - \sum_{j=1}^{p} \delta\epsilon_{t+\frac{1}{2}}\left(e_j \mid v_i\right)$$

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

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

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

$$\begin{aligned}
\alpha_{t+\frac{1}{2}}\left(v_i\right) &= \alpha_t\left(v_i\right) - \sum_{j=1}^{p} \delta\epsilon_{t+\frac{1}{2}}\left(e_j \mid v_i\right) \\
&= \alpha_t\left(v_i\right) - \sum_{j=1}^{p} \frac{m_j\left(v_i\right) w_e\left(e_j\right)}{d_{w,v_i}} \alpha_t\left(v_i\right) \\
&= \alpha_t\left(v_i\right) - \alpha_t\left(v_i\right) \frac{\sum_{j=1}^{p} m_j\left(v_i\right) w_e\left(e_j\right)}{d_{w,v_i}} \\
&= 0.
\end{aligned}$$

□

□

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

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

Proof. We have:

$$\begin{aligned}
\sum_{v_i \in V} \alpha_{t+\frac{1}{2}}(v_i) + \sum_{e \in E} \epsilon_{t+\frac{1}{2}}(e) &= \sum_{e_j \in E} \epsilon_{t+\frac{1}{2}}(e_j) \\
&= \sum_{e_j \in E} \sum_{i=1}^{n} \delta\epsilon_{t+\frac{1}{2}}(e_j \mid v_i) \\
&= \sum_{e_j \in E} \sum_{i=1}^{n} \frac{m_j(v_i)\, w_e(e_j)}{d_{w,v_i}} \alpha_t(v_i) \\
&= \sum_{i=1}^{n} \alpha_t(v_i) \frac{\sum_{e_j \in E} m_j(v_i)\, w_e(e_j)}{d_{w,v_i}} \\
&= \sum_{i=1}^{n} \alpha_t(v_i) \\
&= 1
\end{aligned}$$

□

□

During the second phase that starts at time $t + \frac{1}{2}$, the hb-edges share their values across the vertices they hold taking into account the multiplicity of the vertices in the hb-edge. Every value is modulated by the weight $w_e(e_j)$ of the hb-edge e_j it comes from.

The contribution to $\alpha_{t+1}(v_i)$ given by a hb-edge e_j is propotional 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:

$$\delta\alpha_{t+1}(v_i \mid e_j) = \frac{m_j(v_i)}{\#_m e_j} \epsilon_{t+\frac{1}{2}}(e_j).$$

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:

$$\alpha_{t+1}(v_i) = \sum_{j=1}^{p} \delta\alpha_{t+1}(v_i \mid e_j).$$

Writing $D_E = \operatorname{diag}(\#_m e_j)_{1 \leqslant j \leqslant p}$ the diagonal matrix of size $p \times p$, it comes:

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

The values given to the vertices are subtracted to the value associated to the corresponding hb-edge. Hence, for all $1 \leqslant j \leqslant p$:

$$\epsilon_{t+1}(e_j) = \epsilon_{t+\frac{1}{2}}(e_j) - \sum_{i=1}^{n} \delta\alpha_{t+1}(v_i \mid e_j)$$

Claim 3 (The hb-edges have no value at $t+1$)**.** *It holds:*

$$\epsilon_{t+1}(e_j) = 0.$$

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

$$\begin{aligned}
\epsilon_{t+1}(e_j) &= \epsilon_{t+\frac{1}{2}}(e_j) - \sum_{i=1}^{n} \delta\alpha_{t+1}(v_i \mid e_j) \\
&= \epsilon_{t+\frac{1}{2}}(e_j) - \sum_{i=1}^{n} \frac{m_j(v_i)}{\#_m e_j} \epsilon_{t+\frac{1}{2}}(e_j) \\
&= \epsilon_{t+\frac{1}{2}}(e_j) \left(1 - \frac{\sum_{i=1}^{n} m_j(v_i)}{\#_m e_j}\right) \\
&= 0.
\end{aligned}$$

□

□

Claim 4 (Conservation of the information of the hb-graph at $t+1$)**.** *It holds:*

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

Proof.

$$\begin{aligned}
\sum_{v_i \in V} \alpha_{t+1}(v_i) + \sum_{e \in E} \epsilon_{t+1}(e) &= \sum_{v_i \in V} \alpha_{t+1}(v_i) \\
&= \sum_{v_i \in V} \sum_{j=1}^{p} \delta\alpha_{t+1}(v_i \mid e_j) \\
&= \sum_{v_i \in V} \sum_{j=1}^{p} \frac{m_j(v_i)}{\#_m e_j} \epsilon_{t+\frac{1}{2}}(e_j) \\
&= \sum_{j=1}^{p} \epsilon_{t+\frac{1}{2}}(e_j) \frac{\sum_{v_i \in V} m_j(v_i)}{\#_m e_j} \\
&= \sum_{j=1}^{p} \epsilon_{t+\frac{1}{2}}(e_j) \\
&= 1.
\end{aligned}$$

□

□

Regrouping (1) and (2):

$$P_{V,t+1} = P_{V,t} D_{w,V}^{-1} H W_E D_E^{-1} H^\top. \tag{3}$$

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

Writing $T = D_{w,V}^{-1} H W_E D_E^{-1} H^\top$, it follows from 3:

$$P_{V,t+1} = P_{V,t} T. \tag{4}$$

Claim 5 (Stochastic transition matrix)**.** *T is a square row stochastic matrix of dimension n.*

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

A and B are nonnegative rectangular matrices. Moreover:

$a_{ij} = \dfrac{m_j(v_i)\, w_e(e_j)}{d_{w,v_i}}$ and it holds:

$$\sum_{j=1}^{p} a_{ij} = \frac{\sum_{j=1}^{p} m_j(v_i)\, w_e(e_j)}{d_{w,v_i}} = 1.$$

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

$$\sum_{k=1}^{n} b_{jk} = \frac{\sum_{k=1}^{n} m_j(v_k)}{\#_m e_j} = 1.$$

We have: $P_{V,t+1} = P_{V,t} AB$ where:

$$AB = \left(\sum_{j=1}^{p} a_{ij} b_{jk} \right)_{\substack{1\leqslant i\leqslant n\\ 1\leqslant k\leqslant n}}.$$

It yields:

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

Hence AB is a nonnegative square matrix with its row sums all equal to 1: it is a row stochastic matrix.

□

□

Claim 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 $1 \leqslant i \leqslant n$.

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

□

□

Claim 7. *The sequence* $(P_{V,t})_{t\in\mathbb{N}}$, *with* $P_{V,t} = (\alpha_t(v_i))_{1\leqslant i\leqslant n}$ *in a connected hb-graph converges to the state vector* π_V *such that:*

$$\pi_V = \left(\frac{d_{w,v_i}}{\sum_{k=1}^{n} d_{w,v_k}}\right)_{1\leqslant i\leqslant n}.$$

Proof. We designate by π an eigenvector of T associated to the eigenvalue 1. We have $\pi T = \pi$.

Let consider $u = (d_{w,v_i})_{1\leqslant i\leqslant n}$.

We have

$$\begin{aligned}
(uT)_k &= \sum_{i=1}^{n} d_{w,v_i} \sum_{j=1}^{p} c_{ik} \\
&= \sum_{i=1}^{n} d_{w,v_i} \sum_{j=1}^{p} \frac{m_j(v_i)\, w_e(e_j)}{d_{w,v_i}} \times \frac{m_j(v_k)}{\#_m(e_j)} \\
&= \sum_{j=1}^{p} \sum_{i=1}^{n} m_j(v_i)\, w_e(e_j) \times \frac{m_j(v_k)}{\#_m(e_j)} \\
&= \sum_{j=1}^{p} w_e(e_j)\, m_j(v_k) \frac{\sum_{i=1}^{n} m_j(v_i)}{\#_m(e_j)} \\
&= \sum_{j=1}^{p} w_e(e_j)\, m_j(v_k) \\
&= d_{w,v_k} = u_k
\end{aligned}$$

Hence, u is a nonnegative eigenvector of T associated to the eigenvalue 1.

When we iterate over T which is a stochastic matrix aperiodic and irreducible for a connected hb-graph we are then ensured to converge to a stationary state which is the probability vector associated to the eigenvalue 1. It is unique and is equal to αu such that $\sum_{k=1}^{n} \alpha u_k = 1$.

We have $\alpha = \dfrac{1}{\sum_{k=1}^{n} d_{w,v_k}}$ and hence the result.

□

Claim 8. *The sequence* $\left(P_{E,t+\frac{1}{2}}\right)_{t\in\mathbb{N}}$, *with* $P_{E,t+\frac{1}{2}} = \left(\epsilon_{t+\frac{1}{2}}(e_j)\right)_{1\leqslant j\leqslant p}$ *in a connected hb-graph converges to the state vector* π_E *such that:* $\left(\dfrac{w_e(e_j)\times \#_m(e_j)}{\sum_{k=1}^{n} d_{w,v_k}}\right)_{1\leqslant j\leqslant p}$.

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

We have:

$$\begin{aligned}
\pi_E &= \left(\sum_{i=1}^{n} \frac{d_{w,v_i}}{\sum_{k=1}^{n} d_{w,v_k}} \times \frac{m_j(v_i) \times w_e(e_j)}{d_{w,v_i}} \right)_{1 \leqslant j \leqslant p} \\
&= \left(\sum_{i=1}^{n} \frac{m_j(v_i) \times w_e(e_j)}{\sum_{k=1}^{n} d_{w,v_k}} \right)_{1 \leqslant j \leqslant p} \\
&= \left(\frac{w_e(e_j) \times \sum_{i=1}^{n} m_j(v_i)}{\sum_{k=1}^{n} d_{w,v_k}} \right)_{1 \leqslant j \leqslant p} \\
&= \left(\frac{w_e(e_j) \times \#_m(e_j)}{\sum_{k=1}^{n} d_{w,v_k}} \right)_{1 \leqslant j \leqslant p}.
\end{aligned}$$

All components are nonnegative and we check that the components of this vector sum to one:

$$\begin{aligned}
\sum_{j=1}^{p} \pi_{E,j} &= \frac{\sum_{j=1}^{p} w_e(e_j) \times \sum_{i=1}^{n} m_j(v_i)}{\sum_{k=1}^{n} d_{w,v_k}} \\
&= \frac{\sum_{i=1}^{n} \sum_{j=1}^{p} w_e(e_j) \times m_j(v_i)}{\sum_{k=1}^{n} d_{w,v_k}} \\
&= \frac{\sum_{i=1}^{n} d_{w,v_i}}{\sum_{k=1}^{n} d_{w,v_k}} \\
&= 1.
\end{aligned}$$

□

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

We have gathered the two-phase steps of the exchange-based diffusion process in Algorithm 1. The time complexity of this algorithm is $O(T(d_{\mathcal{H}} n + r_{\mathcal{H}} p))$ where $d_{\mathcal{H}} = \max_{v_i \in V}(d_i)$ is the maximal degree of vertices in the hb-graph and $r_{\mathcal{H}} = \max_{e_j \in E} \left| e_j^{\star} \right|$ is the maximal cardinality of the support of a hb-graph. Usually, $d_{\mathcal{H}}$ and $r_{\mathcal{H}}$ are small compared to n and p. Algorithm 1 can be refined to determine automatically the number of iterations needed by fixing an accepted error to ensure convergence on the values of the vertices and storing the previous state.

4 Results and evaluation

This section firstly adresses the validation of the approach taken on random hb-graphs. Secondly, this approach is applied to help in the processing of the results of Arxiv querying.

Algorithm 1 Exchange-based diffusion

Given:
A hb-graph $\mathcal{H} = (V, E, w_e)$ with $|V| = n$ and $|E| = p$
Number of iterations: T

Initialisation:
For all $v_i \in V : \alpha_i := \dfrac{1}{n}$
For all $e_j \in E : \epsilon_j := 0$

DiffuseFromVerticesToHbEdges():
For $j := 1$ to p:
$\epsilon_j := 0$
For $v_i \in e_j^\star$:
$$\epsilon_j := \epsilon_j + \frac{m_j(v_i)\, w_e(e_j)}{d_{w,m}(v_i)} \alpha_i$$

DiffuseFromHbEdgesToVertices():
For $i := 1$ to n:
$\alpha_i := 0$
For e_j such that $v_i \in e_j^\star$:
$$\alpha_i := \alpha_i + \frac{m_j(v_i)}{\#_m e_j} \epsilon_j$$

Main():
Calculate for all $i : d_{w,m}(v_i)$ and for all $j : \#_m e_j$
For $t = 1$ to T:
DiffuseFromVerticesToHbEdges()
DiffuseFromHbEdgesToVertices()

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 show in Figure 3, 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 $(V_j)_{1 \leqslant j \leqslant k}$. 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 $N_{j,1} \ll N_{j,2}$, $1 \leqslant j \leqslant k$. The number of hb-edges to be built is adjustable: their number is shared between the different groups. The m-cardinality $\#_m(e)$ 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.

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

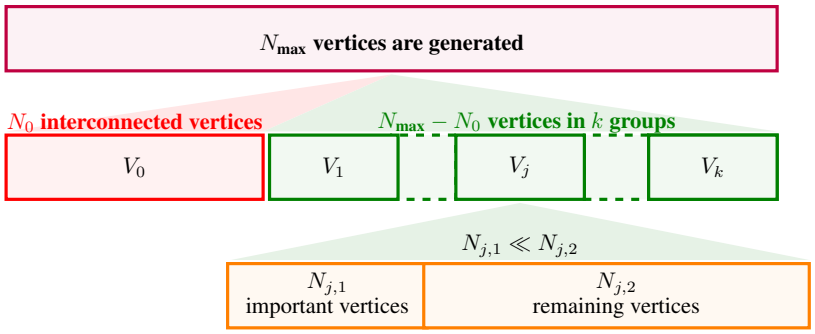


Figure 3: Random hb-graph generation principle

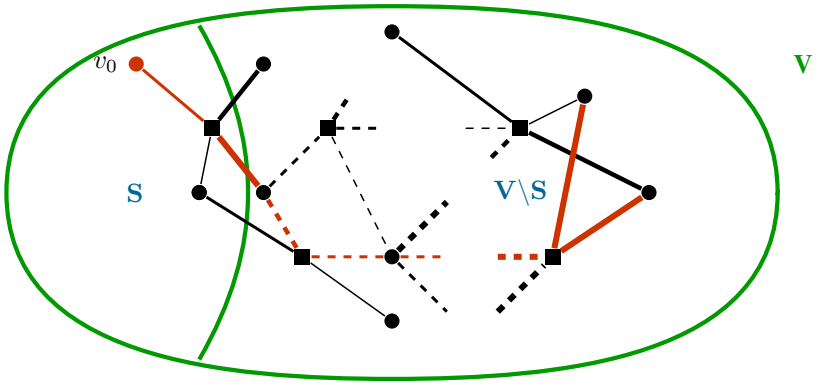


Figure 4: Relative excentricity: finding the length of a maximal shortest path in the hb-graph starting from a given vertex v_0 of S and finishing with any vertex in $V \backslash S$

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 article, 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 excentricity is illustrated in Figure 4.

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 $B_V(s_V) = V \backslash A_V(s_V)$, the set of vertices with α values below this 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 $\alpha_{T,\max} = \max_{v \in V}(\alpha_T(v))$ - 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:

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

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

We show the results obtained in Figure 5 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

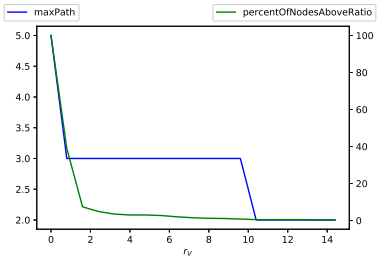


Figure 5: Maximum path length and percentage of vertices in $A_V(s)$ over vertices in V vs ratio r_V.

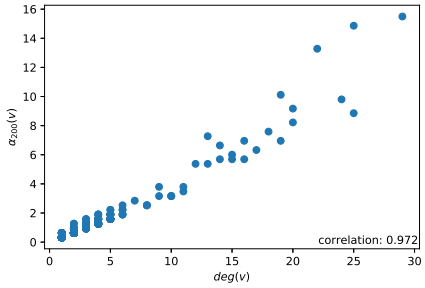


Figure 6: Alpha value of vertices at step 200 and degree of vertices.

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 6 and Figure 7 show that high values of $\alpha_T(v)$ correspond to vertices that are highly connected either by degree or by m-degree. Hence vertices in Figure 8 that are on the positive side of the scale color correspond to highly connected vertices: the closer to red on the right scale they are, the higher the value of $\alpha_T(v)$ is.

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_E: $A_E(s_E)$ of the hb-edges that have ϵ values above the threshold and $B_E(s_E)$ the one gathering hb-edges that have ϵ values below s_E.

s_E varies from 0 to $\epsilon_{T-\frac{1}{2},\max} = \max_{e\in E}\left(\epsilon_{T-\frac{1}{2}}(e)\right)$ by incremental steps while keeping the eccentricity above 0, first of the two conditions achieved. In the hb-graph representation, each hb-edge corresponds to an extra-vertex. Each time we evaluate the length of the maximal shortest path linking one vertex of $A_E(s_E)$ to one vertex of $B_E(s_E)$ for connected vertices in the hb-graph support hypergraph extra-vertex graph representation: the length of the path

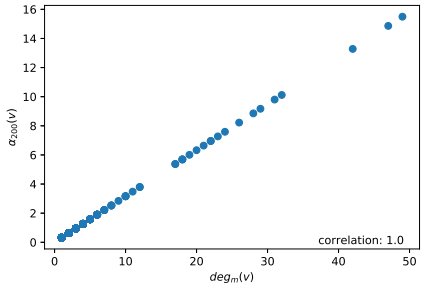


Figure 7: Alpha value of vertices at step 200 and m-degree of vertices.

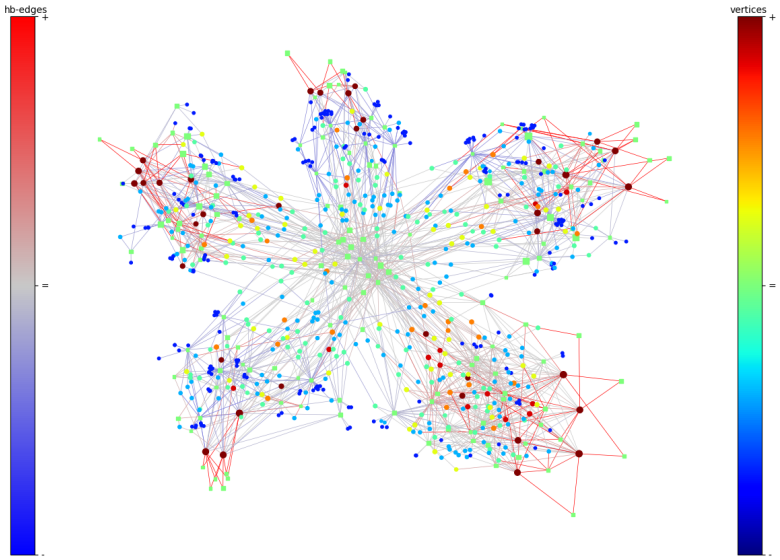


Figure 8: Exchange-based diffusion in hb-graphs after 200 iterations of Algorithm 1: highlighting important hb-edges. Simulation with 807 vertices (chosen randomly out of 10 000) gathered in 5 groups of vertices (with 6, 5, 7, 3 and 5 important vertices and 2 important vertices per hb-edge), 220 hb-edges (with cardinality of support less or equal to 25), 20 vertices in between the 5 groups. Extra-vertices are colored in green and have square shape.

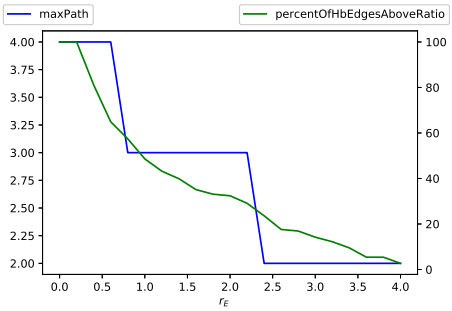


Figure 9: Path maximum length and percentage of vertices in $A_E(s)$ vs ratio.

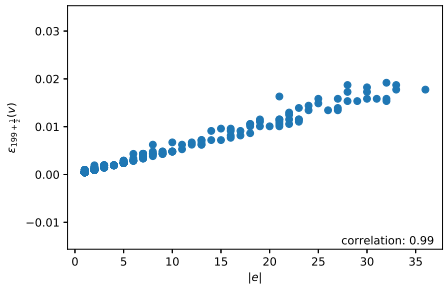


Figure 10: Epsilon value of hb-edge at stage 199+$\frac{1}{2}$ and cardinality of hb-edge.

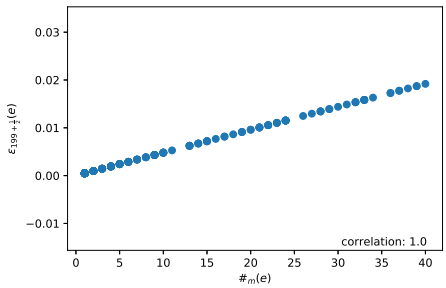


Figure 11: Epsilon value of hb-edge at stage 199+$\frac{1}{2}$ and (m-)cardinality of hb-edge.

corresponds to half of the one obtained from the graph for the same reason than before. We define the ratio

$$r_E = \frac{s_E}{\beta_{\text{ref}}}$$

where $\beta_{\text{ref}} = \frac{1}{|E|}$ that corresponds to the normalised value that would be used in the dual hb-graph to initialise the diffusion process. In Figure 9, we observe for the hb-edges the same trend than the one observed for vertices: the length of the maximal path between two hb-edges decreases as the ratio r_E increases while the percentage of vertices in $A_E(s_E)$ over V decreases.

Figure 10 shows the high correlation between the value of $\epsilon(e)$ and the cardinality of e; Figure 11 shows that the correlation between value of $\epsilon(e)$ and the m-cardinality of e is even stronger.

The number of iterations needed to have a significant convergence depends on the initial conditions; we tried different initialisations, 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 12 and in Figure 13 depict the convergence observed on a uniform initial distribution as it is described in the former section. In Figure 12, we can see how the α-values as we observed in Figure 6 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 13 we can observe an analoguous 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.

The 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 behaviour depends on the node connectivity and the hb-edge composition. It is an open question to investigate on this transitory phase to have more indications on the way the ϵ and the α-values vary.

As we already mentioned the results on hb-edges show that the values obtained are highly correlated to the m-cardinality of the hyperedges. To color the hb-edges as it is done in Figure 8, we calculate the ratio $r_{T-\frac{1}{2}}(e) =$

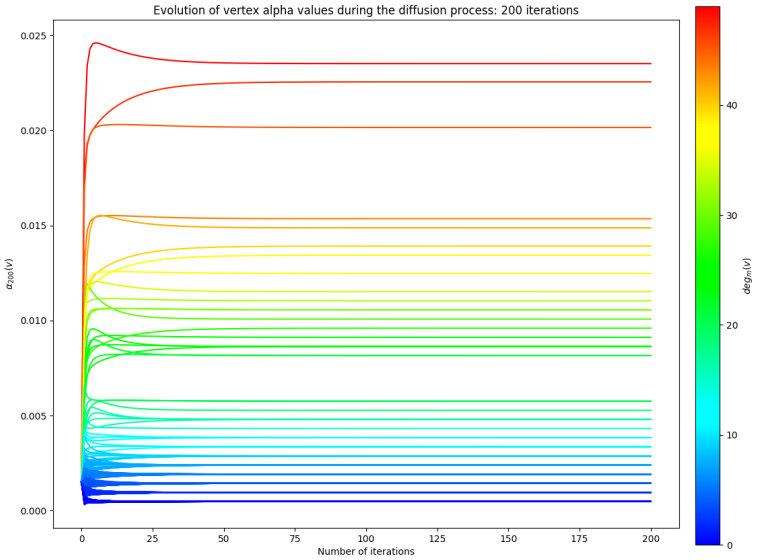


Figure 12: Alpha value convergence of the vertices vs number of iterations. The plots are m-degree-based gradiently colored.

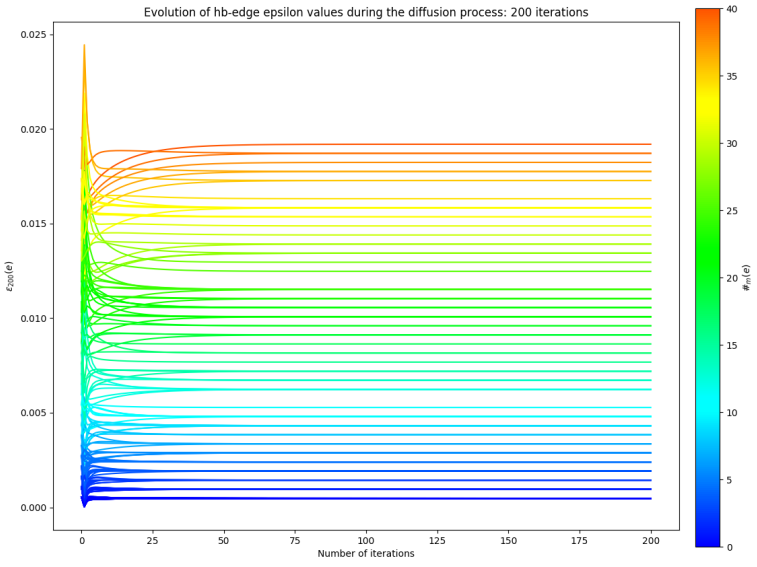


Figure 13: Epsilon value convergence of hb-edges vs number of iterations. The plots are m-cardinality-based gradiently colored.

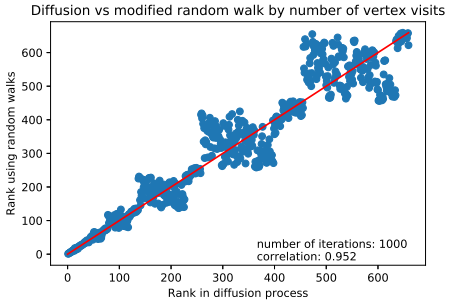


Figure 14: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and rank obtained with 200 iterations of the exchange-based diffusion process.

$\frac{\epsilon_{T-\frac{1}{2}}(e)}{\epsilon_{\text{norm}}(e)}$, where $\epsilon_{\text{norm}}(e) = \sum_{v \in e^\star} \frac{m_e(v)}{\deg_m(v)} v_{\text{ref}}$ corresponds to the value obtained from the vertices of the hb-edge support by giving to each of them the reference value. Hb-edges are colored using $r_{T-\frac{1}{2}}(e)$, the higher the value, the closer to red the color of the left gradient color bar is.

To compare our exchange-based diffusion process to a baseline we considered a classical random walk. In this classical random walk, the walker who is on a vertex v chooses randomly a hb-edge that is incident with a uniform probability law and when the walker is on a hb-edge e he chooses a vertex inside the hb-edge randomly with a uniform probability law. We let the possibility of teleportation to an other vertex from a vertex with a tunable value γ: $1-\gamma$ represents the probability to be teleported. We choose $\gamma = 0.85$. 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 of the classical random walk we propose a modified random walk - described in Algorithm 2 - on the hb-graphs with random choice of hb-edges when the walker is on a vertex v with a distribution of probability $\left(\frac{w_e(e_i)\, m_i(v)}{\deg_{w,m}(v)}\right)_{1 \leqslant i \leqslant p}$ and a random choice of the vertex when the walker is on a hb-edge e with a distribution of probability $\left(\frac{m_e(v_i)}{\#_m(e)}\right)_{1 \leqslant i \leqslant n}$. We let 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 for every hb-edge - with the vertex degree and the hb-edge cardinality instead of the multiplicity - retrieves the classical random walk from the modified random walk.

Figure 14 shows that there is a good correlation between the rank obtained by a thousand modified random walks and after two hundreds iterations of our diffusion process, especially for the first hundred vertices of the network, which is generally the ones that are targetted. 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 15 and Figure 16 is mainly due to the vertices with low m-degrees / degrees.

We can remark in Figure 17 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 18, we can see that the ranks in the classical random walk relies more on the degree than on the m-degree as shown in Figure 19, especially for vertices with small (m-)degrees; but there is still a misclassification for lower (m-)degree vertices.

We have compared the three methods from a computational time perspective; the results are shown in Table 1. The diffusion process is clearly faster; the modified random walk, essentially due to the overhead due to the large number of divisions, requires longer than the classical random walk. A lot of optimisation can be foreseen to make this modified random walk running faster. The random walks can be easily parallelised; it is also the case for the diffusion process. The number of iterations in the diffusion process can also be optimised. These issues will be addressed in future work.

Algorithm 2 Modified random walk in hb-graphs

Given:

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

Initialisation:

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

OneRW():

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

Main():

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

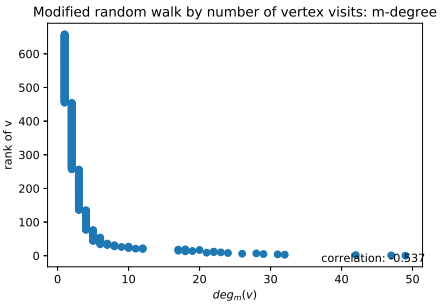


Figure 15: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and m-degree of vertices

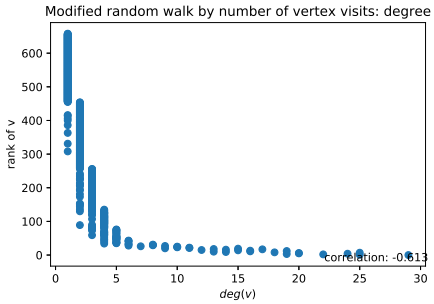


Figure 16: Comparison of the rank obtained by a thousand modified random walks after total discovery of the vertices in the hb-graph and degree of vertices

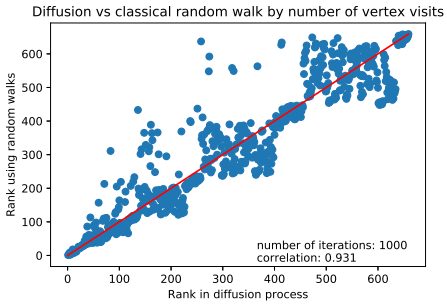


Figure 17: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and rank obtained with 200 iterations of the exchange-based diffusion process.

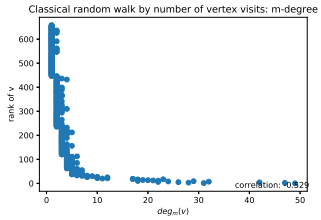


Figure 18: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and m-degree of vertices

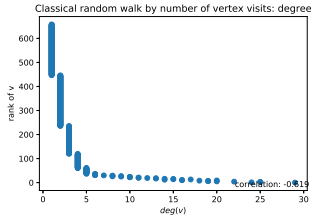


Figure 19: Comparison of the rank obtained by a thousand classical random walks after total discovery of the vertices in the hb-graph and degree of vertices

$\|E\|$	$\|V\|$	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	106	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 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

4.2 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 natural language processing Python 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:

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

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

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

$$\text{TF}(x,d) = \frac{n_x}{n_d}$$

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

$$\text{IDF}\,(x,d) = \log_{10}\left(\frac{N}{n_{x\in d}}\right)$$

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.

The exchange-based diffusion process is applied to the hb-graph $\mathcal{H}_Q$. We show two typical examples on the same query the first 50 results in Figure 20 and the first 100 results in Figure 21. The number of iterations needed to have

[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/

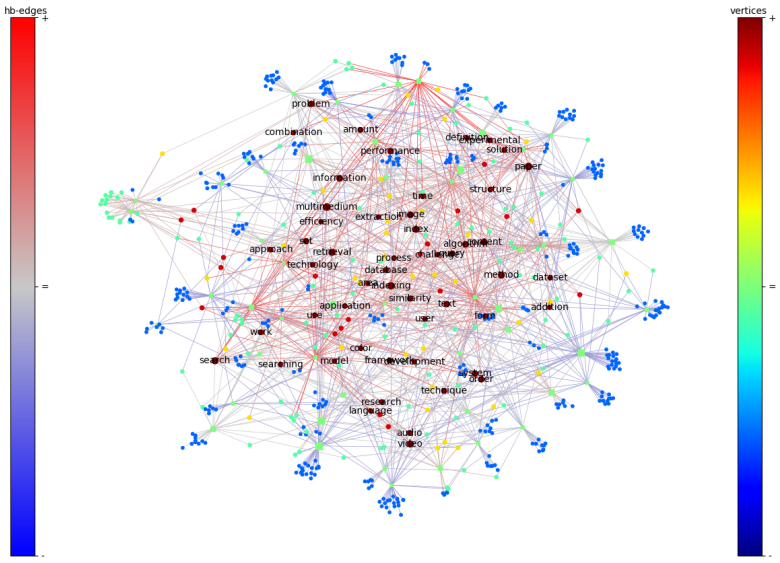


Figure 20: Querying Arxiv. The search performed is “content-based multimedia indexing” for which 50 most relevant articles have been retrieved with 50 iterations.

convergence is less than 10 in these two cases; with 500 results, around 10 iterations are needed for all hb-edges but one where 30 iterations are needed.

As the hb-edges correspond to documents in Arxiv database we compared 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, top 10 documents sometimes change drastically depending on the retrieval of new documents that are more central in the words they contain. If the gap seems not big with a few documents retrieved, this gaps increase as the number of documents increases. The increasing number of results reveal the full theorical 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 importance of documents by considering central subjects in the processing of the results of the query.

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 [21]. This approach can also be viewed as a mean 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.

Acknowledgments

This work is part of the PhD of Xavier OUVRARD, done at UniGe, supervised by Stéphane MARCHAND-MAILLET and founded by a doctoral position at CERN, in Collaboration Spotting team, supervised by Jean-Marie LE GOFF.

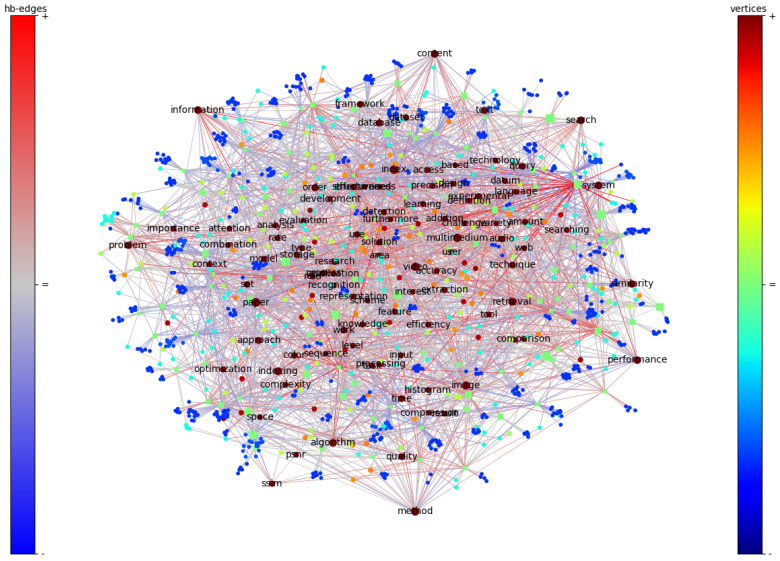


Figure 21: Querying Arxiv. The search performed is “content-based multimedia indexing” for which 100 most relevant articles have been retrieved

References

[1] X. Ouvrard, J.-M. Le Goff, and S. Marchand-Maillet, “Diffusion by exchanges in hb-graphs: Highlighting complex relationships,” *CBMI Proceedings*, 2018.

[2] X. Ouvrard, J.-M. Le Goff, and S. Marchand-Maillet, “A hypergraph based framework for modelisation and visualisation of high dimension multi-facetted data,” *Soon on Arxiv*, 2018.

[3] D. Zhou, J. Huang, and B. Schölkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” in *Advances in neural information processing systems*, pp. 1601–1608, 2007.

[4] A. Bellaachia and M. Al-Dhelaan, “Random walks in hypergraph,” in *Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods, Venice Italy*, pp. 187–194, 2013.

[5] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais, “Pivotpaths: Strolling through faceted information spaces,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2709–2718, 2012.

[6] C. Berge and E. Minieka, *Graphs and hypergraphs*, vol. 7. North-Holland publishing company Amsterdam, 1973.

[7] A. Ducournau and A. Bretto, “Random walks in directed hypergraphs and application to semi-supervised image segmentation,” *Computer Vision and Image Understanding*, vol. 120, pp. 91–102, 2014.

[8] J. Lee, M. Cho, and K. M. Lee, “Hyper-graph matching via reweighted random walks,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1633–1640, IEEE, 2011.

[9] L. Lu and X. Peng, “High-ordered random walks and generalized laplacians on hypergraphs.,” in *WAW*, pp. 14–25, Springer, 2011.

[10] M. E. Newman, “Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality,” *Physical review E*, vol. 64, no. 1, p. 016132, 2001.

[11] M. E. Newman, “Scientific collaboration networks. i. network construction and fundamental results,” *Physical review E*, vol. 64, no. 1, p. 016131, 2001.

[12] J. W. Grossman and P. D. Ion, “On a portion of the well-known collaboration graph,” *Congressus Numerantium*, pp. 129–132, 1995.

[13] C. Taramasco, J.-P. Cointet, and C. Roth, "Academic team formation as evolving hypergraphs," *Scientometrics*, vol. 85, no. 3, pp. 721–740, 2010.

[14] O. N. Temkin, A. V. Zeigarnik, and D. Bonchev, *Chemical reaction networks: a graph-theoretical approach*. CRC Press, 1996.

[15] C. Chauve, M. Patterson, and A. Rajaraman, "Hypergraph covering problems motivated by genome assembly questions," in *International Workshop on Combinatorial Algorithms*, pp. 428–432, Springer, 2013.

[16] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: applications in vlsi domain," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 69–79, 1999.

[17] M. Bendersky and W. B. Croft, "Modeling higher-order term dependencies in information retrieval using query hypergraphs," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 941–950, ACM, 2012.

[18] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He, "Music recommendation by unified hypergraph: combining social media information and music content," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 391–400, ACM, 2010.

[19] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-d object retrieval and recognition with hypergraph analysis," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4290–4303, 2012.

[20] L. Zhu, J. Shen, H. Jin, R. Zheng, and L. Xie, "Content-based visual landmark search via multimodal hypergraph learning," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2756–2769, 2015.

[21] Z. Xu, J. Du, L. Ye, and D. Fan, "Multi-feature indexing for image retrieval based on hypergraph," in *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on*, pp. 494–500, IEEE, 2016.

[22] Y. Wang, L. Zhu, X. Qian, and J. Han, "Joint hypergraph learning for tag-based image retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4437–4451, 2018.

[23] D. Singh, A. Ibrahim, T. Yohanna, and J. Singh, "An overview of the applications of multisets," *Novi Sad Journal of Mathematics*, vol. 37, no. 3, pp. 73–92, 2007.

[24] X. Ouvrard, J.-M. Le Goff, and S. 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.

[25] M. Schmitt, C. Janott, V. Pandit, K. Qian, C. Heiser, W. Hemmert, and B. Schuller, "A bag-of-audio-words approach for snore sounds' excitation localisation," in *Speech Communication; 12. ITG Symposium*, pp. 1–5, VDE, 2016.

[26] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[27] L. Purda and D. Skillicorn, "Accounting variables, deception, and a bag of words: assessing the tools of fraud detection," *Contemporary Accounting Research*, vol. 32, no. 3, pp. 1193–1223, 2015.

[28] S. Ma, X. Sun, Y. Wang, and J. Lin, "Bag-of-words as target for neural machine translation," *arXiv preprint arXiv:1805.04871*, 2018.

[29] N. Cummins, S. Amiriparian, S. Ottl, M. Gerczuk, M. Schmitt, and B. Schuller, "Multimodal bag-of-words for cross domains sentiment analysis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4954–4958, IEEE, 2018.

[30] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*, p. 1470, IEEE, 2003.

[31] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, pp. 1–2, Prague, 2004.

[32] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *2008 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE, 2008.

[33] C.-F. Tsai, "Bag-of-words representation in image annotation: A review," *ISRN Artificial Intelligence*, vol. 2012, 2012.

[34] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.

[35] L.-J. Zhao, P. Tang, and L.-Z. 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*, vol. 7, no. 12, pp. 4620–4631, 2014.

[36] S. Minaee, S. Wang, Y. Wang, S. Chung, X. Wang, E. Fieremans, S. Flanagan, J. Rath, and Y. 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)*, pp. 1–5, IEEE, 2017.

[37] R. Shekhar and C. Jawahar, "Word image retrieval using bag of visual words," in *2012 10th IAPR International Workshop on Document Analysis Systems*, pp. 297–301, IEEE, 2012.

[38] H. K. Kim, H. Kim, and S. Cho, "Bag-of-concepts: Comprehending document representation through clustering words in distributed representation," *Neurocomputing*, vol. 266, pp. 336–352, 2017.

[39] F. B. Silva, R. d. O. Werneck, S. Goldenstein, S. Tabbone, and R. d. S. Torres, "Graph-based bag-of-words for classification," *Pattern Recognition*, vol. 74, pp. 266–285, 2018.