UNIVERSITÉ DE GENÈVE

Archive ouverte UNIGE
https://archive-ouverte.unige.ch

Thèse 2015

Open Access

This version of the publication is provided by the author(s) and made available in accordance with the copyright holder(s).

The Topology of Directed Complex Networks: Computational Analysis and Applications

Dubuisson, Jimmy

How to cite

DUBUISSON, Jimmy. The Topology of Directed Complex Networks: Computational Analysis and Applications. Doctoral Thesis, 2015. doi: 10.13097/archive-ouverte/unige:73280

This publication URL: https://archive-ouverte.unige.ch/unige:73280
Publication DOI: 10.13097/archive-ouverte/unige:73280

© This document is protected by copyright. Please refer to copyright holder(s) for terms of use.

Université de Genève
Département d'Informatique
Département de Physique Théorique

Faculté des sciences
Professeur Bastien Chopard
Professeur Jean-Pierre Eckmann

The Topology of Directed Complex Networks: Computational Analysis and Applications

Thèse

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

par

Jimmy Tommy Julien Dubuisson

de

France

Thèse n° 4783

Genève
Atelier d'impression ReproMail
2015

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

Doctorat ès sciences
Mention informatique

Thèse de ***Monsieur Jimmy DUBUISSON***

intitulée :

"The Topology of Directed Complex Networks: Computational Analysis and Applications"

La Faculté des sciences, sur le préavis de Monsieur B. CHOPARD, professeur ordinaire et directeur de thèse (Département d'informatique), Monsieur J.-P. ECKMANN, professeur honoraire et codirecteur de thèse (Département de physique théorique) et Monsieur P. DE LOS RIOS, professeur associé (Laboratoire de biophysique statistique, Ecole Polytechnique Fédérale de Lausanne, Lausanne), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 22 mai 2015

Thèse - 4783 -

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".

A ma soeur Soline, à mon frère Robin et à moi-même, notre chère Maman nous a transmis l'envie de se dépasser et l'esprit d'excellence.

Je dédie cette thèse à sa mémoire.

Contents

Chapter 1

Remerciements

Je souhaite en premier lieu remercier le professeur Jean-Pierre Eckmann de m'avoir donné l'opportunité de commencer une thèse, à une époque où une telle marque de confiance m'a été précieuse. Sous sa supervision bienveillante et expérimentée, j'ai appris la rigueur scientifique et développé les qualités intellectuelles requises pour mener à bien un travail de recherche. A travers son approche multidisciplinaire, il m'a aussi enseigné à considérer un problème sous différentes perspectives, et notamment celle d'un physicien.

Je remercie aussi le Professeur Christian Pellegrini qui a accepté de codiriger ma thèse à son commencement en tant que directeur du Département d'Informatique, et le professeur Bastien Chopard qui a accepté de reprendre le flambeau lorsque le professeur Pellegrini a pris sa retraite. J'ai eu des discussions fructueuses avec le professeur Bastien Chopard au Département d'Informatique, et j'espère avoir l'occasion de collaborer avec lui sur des projets de recherche dans le futur.

Je souhaite encore remercier le professeur Peter Wittwer du Département de Physique Théorique, et sa femme Teresa Dip Mercado pour les nombreuses discussions amicales que nous avons eues ensemble au cours de ma thèse, ainsi que le professeur Paolo De Los Rios de l'Institut de Physique Théorique de l'Ecole Polytechnique Fédérale de Lausanne qui a accepté de faire partie de mon jury de thèse.

Je garderai un excellent souvenir de l'ambiance familliale qui règne au Département de Physique Théorique, et des innombrables discussions que nous avons eu avec mes collègues doctorants et postdoctorants sur des sujets aussi divers que passionants. Je pense en particulier à Philippe Jacquet, Cyrille Zbinden, Christoph Boeckle, Maher Younan, Julien Guillod, Noé Cuneo, Tanya Yarmola, Alexandre Boritchev, Andrea Agazzi, Niel Dobbs et Rodrigo Cofre.

Je tiens encore à remercier Andreas Malaspinas qui gère de manière flex-

ible et efficace l'infrastructure informatique du Département de Physique Théorique depuis de nombreuses années, ainsi que les secrétaires Francine Gennai-Nicole et Cécile Jaggi-Chevalley qui, en plus d'être toujours souriantes, traitent de manière admirable les questions administratives du département.

Ayant utilisé de nombreux logiciels libres au cours de ma thèse, je souhaite saluer le travail fourni par la communauté du logiciel libre dans son ensemble, et en particulier les développeurs des librairies JUNG, igraph et scikit-learn ainsi que ceux travaillant à l'amélioration des langages julia, java, R et python.

Je remercie vivement Liliya Kolotilova, Shanti Secretan, mon père Yves Dubuisson et sa femme Francine de m'avoir fourni leur aide lorsque mon travail devenait trop prenant. Je n'oublie pas non plus mes amis d'Ukraine dont Yarina Sheremet qui ont su garder courage et optimisme dans cette période troublée, et qui m'ont à chaque fois acceuilli chaleureusement lorsque j'ai eu l'occasion de leur rendre visite.

J'ai enfin une pensée attendrie pour mon fils Nikita, qui est né pendant ma thèse. Malgré son jeune âge, il m'a prouvé que je pouvais toujours compter sur lui dans les moments importants. Sa joie de vivre et mon bonheur de le voir grandir m'ont donné la force de surmonter les obstacles qui ont jalonné cette aventure.

Chapter 2

Résumé

Cette thèse est consacrée à l'étude de la topologie des réseaux complexes orientés basée sur l'analyse computationnelle et statistique, ainsi qu'au développement de nouveaux outils computationnels dédiés à l'analyse de données modélisées sous forme de graphes orientés.

Dans la première partie, nous faisons suite à un travail dédié à l'étude de graphes de définitions de mots, et nous nous intéressons à l'analyse de graphes orientés modélisant un ensemble d'associations libres basées sur des mots. Nous utilisons pour ce faire une base de données d'associations libres collectée à l'aide de nombreux questionnaires par des chercheurs en psychologie de l'Université de Floride du Sud (USF), depuis les années 70. Après avoir isolé le "core" du graphe d'associations libres (i.e., sa composante fortement connexe principale), nous analysons ses propriétés topologiques fondamentales, et réalisons une première série d'observations mettant en lumière les liens existant entre la topologie du graphe et la sémantique des mots. Nous montrons ensuite que le "core" contient une proportion beaucoup plus élevée de cycles courts (i.e., cycles orientés de longueur 2 ou "colinks", et cycles orientés de longueur 3 ou "triangles") qu'un graphe aléatoire de taille équivalente et que, d'un point de vue topologique, les cycles de longueur supérieure à 3 apparaissent en réalité lorsque des triangles ont des sommets en commun, ou lorsqu'ils sont reliés par des arêtes n'appartenant pas à un triangle que nous appelons "shortcuts". Après avoir vérifié que la grande majorité des noeuds du "core" appartiennent à un triangle, nous en déduisons que les triangles jouent un rôle essentiel dans la structure du graphe. Nous isolons le "seed", c'est à dire la composante fortement connexe principale du sous-graphe induit par l'ensemble des noeuds appartenant à un triangle dans le "core", et observons que le "seed" se décompose de manière homogène en un ensemble de grappes de triangles (i.e., sous-graphes complets de triangles) de tailles comparables. Nous démontrons pour finir que ces grappes de

triangles regroupent des mots fortement liés du point de vue sémantique, et nous en concluons que les triangles constituent des "briques" sémantiques fondamentales du réseau sémantique formé par le "core".

Dans la deuxième partie, nous présentons une nouvelle méthode d'analyse de données basée sur l'utilisation de processus de diffusion sur un graphe orienté. Pour une base de données modélisée sous forme d'un graphe orienté (nous parlons de "domain graph"), l'idée est de générer des vecteurs numériques de grande dimension à partir des valeurs de la distribution d'une marche aléatoire biaisée associées à chacun des noeuds du graphe. En faisant démarrer ces processus de diffusion à partir de sous-ensembles quelconques de la base de données considérée (i.e., à partir de sous-graphes quelconques du "domain graph"), nous sommes ainsi en mesure de générer des "signatures" (nous utilisons le terme de "diffusion fingerprints") permettant de caractériser ces sous-ensembles. Nous montrons d'abord l'efficacité de notre méthode en l'appliquant au problème consistant à extraire l'ensemble des voies métaboliques d'un graphe métabolique. Grâce à un algorithme simple et n'utilisant pas de paramètre, nous parvenons à un niveau de précision comparable aux meilleurs résultats présentés dans la littérature actuelle. Nous fournissons ensuite un guide expliquant comment utiliser notre méthode pour classifier des données: nous illustrons notre approche de manière simple en l'appliquant à deux problèmes classiques de classification textuelle. Nous discutons enfin de manière plus intuitive de différentes problématiques liées à l'implémentation de notre méthode, et présentons en particulier une heuristique permettant de réduire la dimension des vecteurs de diffusion générés, pour un coût computationnel négligeable.

Dans la troisième partie, nous nous intéressons au problème consistant à échantilloner un réseau complexe orienté de très grande taille, dans le cas où il n'est pas possible de l'explorer entièrement, et que sa structure n'est connu que très partiellement. Nous commençons par décrire un ensemble de mesures fondamentales dont nous souhaitons que les valeurs soit préservées au mieux dans l'échantillon extrait, et présentons un ensemble d'algorithmes déterministes et probabilistes permettant de calculer ces mesures sur des graphes de très grande taille. Nous discutons ensuite des méthodes existantes visant à obtenir des échantillons uniformes de noeuds pour des graphes non-orientés et orientés. Enfin, nous décrivons une heuristique qui, sous les contraintes fortes que nous nous sommes fixées (i.e., cas d'un graphe orienté et dont la structure reste inconnue), explore un graphe orienté de manière itérative à partir d'un noeud sélectionné au hasard, en cherchant à en extraire un sous-graphe connecté reproduisant le mieux possible ses caractéristiques fondamentales. Nous testons enfin notre heuristique sur une copie complète du réseau social Twitter téléchargée en 2009, et discutons de différents détails

techniques liés à son implémentation.

Chapter 3

Introduction

3.1 The Fathers of Graph Theory

The "*seven bridges of Königsberg*" is an influential problem which was stated and solved in 1735 by the famous mathematician Leonhard Euler. Its resolution required the development of a new formalism, and is regarded as the first theorem of a new branch of mathematics nowadays called *graph theory* [1]. It also prefigured the fundamental idea of *topology* which has since kept deep bonds with graph theory.

Another seminal work that got Euler's attention in 1759 consists in finding a "Knight's Tour" on a chessboard [2], and some mathematicians including Vandermonde [3] and Warnsdorf [4] later attempted to find a general solution to the problem. In 1857, William Rowan Hamilton invented the "Icosian game" [5] which is based on a similar principle: its goal is indeed to find a path that visits the vertices of a dodecahedron exactly once. In the modern terminology of graph theory, these problems are in fact closely related, and correspond to searching for a so-called "*hamiltonian path*".

Around the same period, the physicist Gustav Robert Kirchhoff made important contributions to the emergent theory by studying properties of electrical networks. In particular, he proved in 1847 the *matrix-tree theorem* which relates the number of spanning trees to the *laplacian matrix* of a graph [6].

The profilic mathematician James Joseph Sylvester is also considered one of the founders of graph theory. While working on his theory of invariants, he was the first to coin the term "*graph*" in 1878 [7], for referring to special diagrams that started to be used by chemists to represent chemical compounds. These diagrams depicted individual atoms by labeled dots (called *vertices* in the modern terminology), and chemical bonds by lines (*i.e.*, *edges*) joining

them. His good friend Arthur Cayley was the first to introduce the term "*tree*" [8], and proved in 1889 a formula named after him which gives the number of trees on n labeled vertices [9].

The diagrams of chemical compounds used by Sylvester are examples of graphs that exhibit an important property called *planarity*: they can indeed be drawn on the Euclidean plane in such a way that no edges cross each other. The study of such planar graphs has lead to a series of essential results, starting with Euler's famous formula published in 1758 [10, 11] that relates the number of edges e, vertices v and faces f of a connected planar graph. Between 1930 and 1937, Kazimierz Kuratowski [12], Hassler Whitney [13] and Klaus Wagner [14] provided different characterizations of planar graphs: Kuratowski's and Wagner's theorems characterize *planarity* in terms of the two forbidden graphs K_5 (the complete graph on five vertices) and $K_{3,3}$ (the complete bipartite graph on six vertices), whereas Withney's *planarity criterion* makes use of *graphic matroids* and states that a graph is planar if and only if it has an algebraic *dual*. In a long series of papers published between 1983 and 2004, Paul Seymour and Neil Robertson generalized these forbidden graph chatacterizations by proving the *graph minor theorem* [15, 16], after Joseph Kruskal proved a weaker result for trees in 1960 [17].

While coloring a map of English counties in 1852, Francis Guthrie conjectured that four colors were sufficient to color a map in such a way that no two contiguous regions share the same color. The now famous "*four color theorem*" can be naturally restated in terms of graph theory, as it amounts to coloring the vertices of the dual graph of the map under consideration. Alfred Kempe gave a first proof of the theorem in 1879 [18], but Percy John Heawood showed that it was flawed in 1890 [19], and proved at the same time the "*five color theorem*". It is only in 1989 that Kenneth Appel and Wolfgang Haken were able to provide the first ever computer-assisted proof of this major theorem [20].

William Thomas Tutte's outstanding research has been highly influential on the development of modern graph theory. In his Ph.D. thesis "An Algebraic Theory of Graphs" completed in 1948, Tutte greatly expanded the initial work of Hassler Whitney on *matroids* and presented his famous "*dichromatic polynomial*" [21], which found applications in fields as diverse as percolation theory, knot theory and statistical mechanics. In a paper published in 1954 [22], he extended his algebraic approach of graph theory and conjectured the still unsolved "*5-flow conjecture*". He also developed what he called a "geometrical version" of the classic "four color problem" [23].

Another prominent figure of graph theory is the famous mathematician Paul Erdős. He made numerous contributions to the field, in particular regarding the study of *random graphs*. In 1959, he attended the first interna-

tional conference in graph theory and proposed with Alfréd Rényi a model of random graph [24]. This classic model is now called the *Erdős–Rényi (ER) random graph model* and actually refers to two closely related models:

— $G(n, M)$ which represents a graph chosen uniformly at random from the set of graphs with n vertices and M edges,

— $G(n, p)$ which represents a graph with n vertices connected randomly with probability p.

In 1960, Erdős and Rényi published a second paper [25] to study the behaviour of $G(n, p)$ for different values of p. In the same vein, Erdős and Béla Bollobas used a probabilistic approach to predict the number of *cliques* in a random graph [26].

Since its rather discrete inception, graph theory has grown into a deep mathematical theory with powerful applications spanning through many areas of science. Scientists who desire to study the field today have excellent reference books at disposal [27, 28, 29, 30].

3.2 What is a Complex Network?

With the quick development of computer technologies following World War II, both the storage capacity of digital data and the processing power of computing systems increased dramatically. This has enabled scientists to start using computers for simulating computational models and for analyzing scientific data [31, 32, 33]. In the nascent field of Machine Learning (ML) for example, artifical neural networks (ANNs) were simulated for the first time on computational machines [34, 35, 36], after Warren McCulloch and Walter Pitts published their seminal paper presenting a computational model for neural networks [37].

The computational analysis of graphs modeling real systems gained momentum in the beginning of the 1990's, in parallel with the development of the World Wide Web (WWW). As more data was becoming available, scientists of various disciplines began to analyze real-world networks (social networks, biological networks, computer networks, the Web, ...), and quickly realized that the topology of most of those networks deviates significantly from the one of lattices and random graphs.

It appeared in particular that real networks have a short average path length between any two nodes, which is known as the *small-world property*. In 1967, the psychologist Stanley Milgram showed for example with his famous *small-world experiment* that the average path length between any two people was around six in the United States [38]. The small-world property of social networks was later confirmed by two other experiments: one using e-mail

messages instead of letters [39], and the other using a huge dataset of instant messages [40].

Although the classical Erdős-Rényi (ER) model of random graph also exhibits a short average path length, it falls short of reproducing certain peculiarities of real-world networks. In particular, ER graphs are known to have a low *clustering coefficient* (*i.e.*, low density of triangles), which contradicts what is generally observed empirically in the wild [41].

In order to account for these discrepancies, Duncan Watts and Steven Strogatz developed in 1998 a model that generates *small-world* random graphs with both a short average path length, and a high clustering coefficient [42]. However, the *Watts-Strogatz* (WS) model fails to recreate the typical heavy-tailed degree distribution of most real-world networks.

In 1999, Albert-László Barabási and Réka Albert coined the term "*scale-free*" to refer to the class of networks whose node degrees follow a power law distribution [43, 44, 45], and proposed a generative model using a *preferential attachment* mechanism to explain the appearance of such degree distributions. But this dynamical process also introduces non-trivial correlations that affect the topological properties of the generated graphs: the clustering coefficient decays for example with the network size, whereas it keeps constant in the small-world models [46].

Thus, most real-world networks present complex topological features that random graph models fail to reproduce in a way or another and, for this reason, get labeled as *complex networks*. Among the main topological characteristics they usually exhibit concurrently, we find:

— the scale-free property (*i.e.*, power law degree distribution),
— the small-world property (*i.e.*, small average shortest path associated with a high clustering coefficient),
— assortativity or disassortativity among vertices [47],
— a high reciprocity in the case of directed networks [48],
— a community structure [49],
— a hierarchical structure [46, 50, 51].

As a recent field of research, the study of complex networks is currently very active, and interested readers can find numerous references to go deeper into the topic [46, 52, 53].

3.3 Our Contributions

In recent years, directed complex networks have been successfully used in various disciplines of science for modeling real complex systems, as they enable the use of powerful graph theoretical tools to get new insights into

the structure and dynamics of the data.

Although their rich topology constitutes an interesting subject of study, directed graphs are known to be harder to handle from a mathematical point of view than their undirected counterparts. Thus, considering their complicated structure, directed complex networks can be most efficiently analyzed by means of computational and statistical approaches.

In the present thesis, we focus on the study of real-world directed complex networks. First, we show how a thorough computational analysis of a network topology enables to get deep insights into the structural properties of the underlying data. Second, we develop a method for leveraging the topological properties of data exhibiting associative properties, after it is modeled as a directed graph. Finally, we discuss the problem of sampling massive directed networks, and present a heuristic algorithm aiming at extracting a representative subgraph from a yet unexplored graph.

3.3.1 The Topology of Semantic Knowledge

The use of graphs for knowledge representation can be traced back to the greek philosopher Porphyry. In the third century AD, he illustrated Aristotle's method of defining categories by drawing the oldest known *semantic network* [54, 55]. But it is much later in 1956 that Richard Richens proposed to use "semantic nets" as an "interlingua" for *Machine Translation* (MT) of natural languages [56]. The first implementations of such networks were made shortly after by some of the pioneers of *computational linguistics*, in particular Margaret Masterman and her colleagues at the Cambridge Language Research Unit (CLRU) [57], Silvio Ceccato [58], as well as Robert Simmons and Ross Quillian at System Development Corporation [59, 60].

Started in 1985 in the Cognitive Science Laboratory of Princeton University, *Wordnet* is a large lexical database of the English language which is organized as a semantic network of "*synsets*" (*i.e.*, sets of synonym words), and has found numerous applications in computational linguistics and natural language processing [61, 62]. Networked-based approaches have thus proved to be useful tools for the automatic analysis of textual data, and we believe many potential applications still remain to be discovered.

One of the key requirements to leverage graph theoretical analysis in computational linguistics is to better understand the specific topology of the graphs used to model natural languages. Our contribution follows a seminal work by [63] in which the authors study the structure and dynamics of cycles in graphs of dictionary definitions.

For a given English dictionary, a directed graph $G(V, E)$ such that $V = \{nouns \in dictionary\}$ and $E = \{(n_s, n_t) \in V \times V : n_t \in D(n_s)\}$ is generated,

where $D(n)$ denotes the set of words defining n. It is observed that a random walker visiting the dictionary graph gets trapped after a few steps in a subset of highly connected nodes. This subgraph is extracted by a method called "*shaving*", which consists in recursively removing the source nodes (*i.e.*, the set of words that appear in no definition). Intuitively, this corresponds to selecting the subset of words used to define all the others, or more formally, to extracting the main strongly connected component of the initial graph. This subset of nodes, which is called "*core*" in the article, shows to have a large intersection with well-known lists of basic words like *Ogden's Basic English* (52%) [64], or *Gutenberg* 1 000 *Top Nouns* (39%) [65].

After this initial step, the edges of the core get labeled by the length of the shortest directed cycle they belong to. It is then observed that the dictionary graph under study exhibits many more 2-cycles (*i.e.*, colinks) and 3-cycles (*i.e.*, triangles) than a random graph of equivalent size, and that the distribution of shortest cycle lengths appears to be universal across dictionaries (*e.g.*, WordNet, Wiktionary, ...). Furthermore, these short cycles group words that are clearly related at the semantic level, whereas long cycles are subject to semantic ambiguities.

This is the reason why the edges belonging to long cycles (*i.e.*, whose length is greater than 5) are then removed from the core which decomposes into 390 small clusters. A matrix A whose entries A_{ij} are the number of paths of length at most 5 between word i and cluster j is build, and a singular value decomposition (SVD) is performed on it, which leads to associations of semantically related words, in the form of singular vectors.

The authors then go further to study the dynamics of creation of cycles in graphs of dictionary definitions. With the help of the *Online Etimology Dictionary* [66], words are associated with a date and the nodes of the graph get ordered along the time axis. The key idea is that, when a cycle appears in the graph, the definitions of its element words become interdependent and a new concept is created. This is to be contrasted with the case of a local tree, where the leaf nodes (*i.e.*, *definiens*), which appeared earlier in the lexicon, suffice to characterize the word at the root (*i.e.*, *definiendum*).

Interestingly, words within cycles tend to be added to the lexicon around the same period, and two peaks of cycles creation are shown to appear around the 14th and the 19th centuries. These peaks seem to be associated with the development of the printing press, and the progress of modern science and technology respectively. Finally, the author propose a model to explain the growth of the lexicon and the formation of cycles in the lexical network.

In this part, we use a comparable approach to analyze the topology of a directed graph of free associations [67], and show how it provides deep insights into the semantics of words. It is to be noted that, by contrast

to a dictionary whose construction follows a carefully planned process, the dataset of free associations we used is rather the projection of an implicit and decentralized collective semantic memory. Thus, the topology of the graph we obtain emerges naturally, which precisely makes it an interesting subject of study.

3.3.2 Diffusion Fingerprints

In 1905, Karl Pearson posed the so-called "*drunkard's walk*" problem to the readers of *Nature* [68], as its resolution was according to him "of considerable interest". Formally, the problem consists in calculating the probability that a *random walk* on a 2-dimensional lattice returns to the origin. Lord Rayleigh replied within one week by giving a solution to the problem [69]. A few years later, George Pólya generalized the problem to d-dimensional lattices and showed that a random walk is recurrent only if $d < 3$ [70]. The study of random walks have since found numerous fundamental applications in mathematics [71], physics [72], economics [73], genetics [74, 75], ...

Within the framework of computational science, random walks have proved to be of particular interest for exploring the topology of complex networks. Moreover, the theory of random walks on graphs benefits from a well-established mathematical background as it is closely related to the theory of finite *Markov chains* [76, 77].

In November 1990, Tim Berners-Lee launched the *World Wide Web* (Web, WWW or W3) at the European Center for Nuclear Research (CERN) in Geneva [78]. His idea was initially to develop an internet-based infrastructure for enabling the exchange of data in the form of hypertext documents among the scientific community. But after it was made available to the public in August 1991, and even more after the Mosaic web browser was introduced in 1993, the WWW experienced a tremendous success and has become since then a powerful communication medium at the global scale.

Considering the exponential growth of the Web throughout the 90's, it appeared quickly that new methods needed to be developed for efficient search and retrieval of relevant web pages. Interestingly, the hyperlink structure of the Web can be modeled as a directed graph (*i.e.*, the *webgraph*), with the nodes corresponding to the web pages and the edges to the hyperlinks. This led Sergey Brin and Lawrence Page to adopt a graph theoretical approach to the problem of ranking web pages, by devising in 1998 the *Pagerank* algorithm [79, 80]. One year later, Jon Kleinberg proposed *HITS* [81], another link-based ranking algorithm that computes a *hub* and *authority* score for each document.

The Pagerank algorithm computes iteratively a score for each web page

until convergence, and the resulting Pagerank vector can be seen as the stationary distribution of a modified random walk over the webgraph. The intuitive justification for the algorithm is to model the behaviour of a *random surfer* who clicks hyperlinks at random regardless of the content, but get sometimes bored, and then jumps to any page uniformly at random. The probability of jumping at any step is determined by the only parameter of the algorithm: the *jumping constant* α which is related to the *damping factor* d by the simple formula $\alpha = 1 - d$.

The use of the Pagerank for ranking web pages was a key factor which contributed to the early success of the *Google* search engine [82]. The analysis of its properties have thus caught much attention and the algorithm have been thoroughly studied [83, 84, 85, 86, 87]. The Pagerank algorithm has also been successfully applied for ranking authors in co-citation networks [88, 89, 90], for word sense disambiguation in semantic networks [91, 92], for modeling influence in social networks [93, 94], for analyzing protein networks [95, 96], etc...

An important difference between Pagerank and HITS is that the first needs to be computed offline only once, whereas the second has to be computed for each query. Although this makes Pagerank more suitable when achieving a short query time is essential, a user-specific ranking may be desirable in order to personalize the search results. This is what motivated the definition of the *Topic-Sensitive Pagerank* [97], and of the *Personalized Pagerank* [80, 98, 99]. The latter algorithm is a generalization of the Pagerank, in the sense that it allows the random jumps to follow not only the uniform distribution, but any arbitrary distribution over the graph nodes called the *personalized vector*.

The personalized Pagerank (ppr) vector can thus be seen as the trace left in the graph by a diffusing process started from a specific subset of nodes. This lead us to think that diffusion processes may serve to characterize such subgraphs. We know for example that in *Natural Language Processing* (NLP), word contexts (*i.e.*, text surrounding a given word) are used for semantic disambiguation. Thus, when applied to semantic networks, diffusion processes may provide a tool for characterizing semantic contexts, and help for *Word Sense Disambiguation* (WSD) [100].

In this chapter, we formalize these ideas and present a novel method called *Diffusion Fingerprints*. It consists in using diffusion processes over a graph for generating high-dimensional numerical vectors that can be efficiently used for classifying and clustering data exhibiting associative properties.

3.3.3 Sampling Massive Directed Complex Networks

The empirical study of real-world complex networks is made particularly challenging when computationally intensive algorithms become impractical on large graphs, or when the structure of a graph is only partially known. The problem became particularly acute by the end of the 90s, because the Web had grown so fast that its topological properties could only be roughly estimated. Various papers tried to estimate local and global properties of the webgraph by using crawls of different sizes [101, 45, 102].

Thus, the only remaining solution for studying very large graphs is to use samples. This comes with a cost however, as working with a sample necessarily induces an error of approximation. Different sampling strategies aiming at mitigating the problem were investigated in the context of the Web [103, 104], of *Online Social Networks* (OSNs) [105, 106, 107, 108, 109], and of various other complex networks [110, 111].

If the whole graph is known, it is straightforward to select a uniform sample of vertices. However, if the graph is yet unexplored, getting an unbiased sample gets more difficult. A useful workaround consists in scanning the space of vertex identifiers (IDs) at random until a sufficient number of real IDs is collected [112], but in practice, the method is rarely usable, either because testing the validity of randomly generated IDs is not feasible or too costly.

As an alternative, one can resort to random walks to explore the graph, but it is known that a simple random walk is biased toward high-degree nodes [76], which also applies to classic search algorithms like *Breadth First Search* (BFS) and *Depth First Search* (DFS) [113, 114]. In the specific case of undirected graphs, two modified random walk algorithms permit to correct efficiently the bias of a simple random walk and obtain a uniform sample of vertices: the *Metropolis-Hastings Random Walk* (MHRW) explores new nodes with a probability inversely proportional to their degree [115], and the *Re-Weighted Random Walk* (RWRW) resamples the set of nodes discovered by a simple random walk [112].

The problem becomes even more difficult in the case of unexplored directed graphs, as no method exists for correcting the bias of a random walk by using only local properties of the graph. In [103], the authors try to unbias the exploration process by computing local estimates of the stationary distribution with the help of the Pagerank algorithm. In the same vein as the MHRW algorithm for undirected graphs, the bias of a random walk over a directed graph can indeed be corrected by sampling nodes with a probability inversely proportional to the simple random walk stationary probability associated to them. Other studies assume that the node in-bounds are known

or can be queried to a search engine, and transform the directed graph in an undirected one, in order to solve the problem in a simpler setting [116, 104].

In this last chapter, we study the problem of sampling a massive directed graph under strong but realistic constraints. We assume in particular that the whole structure of the graph is unknown, it is unfeasible to explore the graph entirely and too costly to query vertex IDs at random, node in-bounds cannot be queried to a search engine, and we only have a single seed node to start the exploration process. We first define a set of basic measures, and present a few algorithms for computing them deterministically on a massive directed graph. Then we present a heuristic that grows a strongly connected subgraph iteratively around a seed node, and aims at generating a subgraph which mimicks as faithfully as possible the properties of the whole graph, as measured previously. We illustrate our method by using a crawl of the Twitter social network collected in 2009 [117].

Bibliography

[1] L. Euler, "Solutio problematis ad geometriam situs pertinentis," *Commentarii academiae scientiarum Petropolitanae*, vol. 8, pp. 128–140, 1741.

[2] L. Euler, "Solution d'une question curieuse qui ne paroît soumise à aucune analyse," *Mémoires de l'Academie Royale des Sciences et Belles Lettres*, vol. 15, pp. 310–337, 1766.

[3] A.-T. Vandermonde, "L'histoire de l'académie des sciences," vol. 1771, pp. 566–574, 1774.

[4] H. C. von Warnsdorf, "Des rösselsprungs einfachste und allgemeinste lösung," *Th. G. Fr. Varnhagensehen Buchhandlung*, 1823.

[5] W. R. Hamilton, "Lvi. memorandum respecting a new system of roots of unity," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 12, no. 81, pp. 446–446, 1856.

[6] G. Kirchhoff, "Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird," *Annalen der Physik*, vol. 148, no. 12, pp. 497–508, 1847.

[7] J. J. Sylvester, "On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, with three appendices," *American Journal of Mathematics*, vol. 1, no. 1, pp. pp. 64–104, 1878.

[8] A. Cayley, "Xxviii. on the theory of the analytical forms called trees," *Philosophical Magazine Series 4*, vol. 13, no. 85, pp. 172–176, 1857.

[9] A. Cayley, "A theorem on trees," *Quarterly Journal of Mathematics*, vol. 23, pp. 376–378, 1889.

[10] L. Euler, "Demonstratio nonnullarum insignium proprietatum, quibus solida hedris planis inclusa sunt praedita," *Novi Commentarii academiae scientiarum Petropolitanae*, vol. 4, pp. 140–160, 1758. Presented to the St. Petersburg Academy on April 6, 1752. *Opera Omnia* 1(26): 94–108.

[11] L. Euler, "Elementa doctrinae solidarum," *Novi Commentarii academiae scientiarum Petropolitanae*, vol. 4, pp. 109–140, 1758. Presented to the Berlin Academy on November 26, 1750. *Opera Omnia* 1(26): 71–93.

[12] C. Kuratowski, "Sur le problème des courbes gauches en topologie," *Fundamenta Mathematicae*, vol. 15, no. 1, pp. 271–283, 1930.

[13] H. Whitney, "Non-separable and planar graphs," *Trans. Amer. Math. Soc.*, vol. 34, pp. 339–362, 1932.

[14] K. Wagner, "Über eine Eigenschaft der ebenen Komplexe," *Math. Ann.*, vol. 114, 1937.

[15] N. Robertson and P. D. Seymour, "Graph minors. i. excluding a forest," *Journal of Combinatorial Theory, Series B*, vol. 35, no. 1, pp. 39–61, 1983.

[16] N. Robertson and P. D. Seymour, "Graph minors. xx. wagner's conjecture," *Journal of Combinatorial Theory, Series B*, vol. 92, no. 2, pp. 325–357, 2004.

[17] J. B. Kruskal, "Well-quasi-ordering, the tree theorem, and vazsonyi's conjecture," *Transactions of the American Mathematical Society*, pp. 210–225, 1960.

[18] A. B. Kempe, "On the geographical problem of the four colours," *American Journal of Mathematics*, vol. 2, no. 3, pp. pp. 193–200, 1879.

[19] P. J. Heawood, "Map colour theorem," *Quarterly Journal of Pure and Applied Mathematics*, vol. 24, pp. 332–338, 1890.

[20] K. Appel and W. Haken, *Every Planar Map is Four Colorable*. Contemporary mathematics, American Mathematical Society, 1989.

[21] W. T. Tutte, "A ring in graph theory," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 43, pp. 26–40, 1 1947.

[22] W. T. Tutte, "A contribution to the theory of chromatic polynomials," *Canad. J. Math*, vol. 6, no. 80-91, pp. 3–4, 1954.

[23] W. T. Tutte, "On the algebraic theory of graph colorings," *Journal of combinatorial theory*, vol. 1, no. 1, pp. 15–50, 1966.

[24] P. Erdős and A. Rényi, "On random graphs i.," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[25] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[26] B. Bollobás and P. Erdős, "Cliques in random graphs," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 80, pp. 419–427, Cambridge Univ Press, 1976.

[27] N. Biggs, *Algebraic graph theory*. Cambridge university press, 1993.

[28] F. Chung, *Spectral graph theory*, vol. 92. American Mathematical Soc., 1997.

[29] B. Bollobás, *Modern graph theory*, vol. 184. Springer, 1998.

[30] B. Bollobás, *Extremal graph theory*. Courier Dover Publications, 2004.

[31] S. Ulam, R. Richtmyer, and J. Von Neumann, "Statistical methods in neutron diffusion," *LAMS-551, Los Alamos National Laboratory*, pp. 1–22, 1947.

[32] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[33] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[34] B. G. Farley and W. Clark, "Simulation of self-organizing systems by digital computer," *Information Theory, Transactions of the IRE Professional Group on*, vol. 4, no. 4, pp. 76–84, 1954.

[35] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *Information Theory, IRE Transactions on*, vol. 2, no. 3, pp. 80–93, 1956.

[36] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[37] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[38] J. Travers and S. Milgram, "An experimental study of the small world problem," *Sociometry*, pp. 425–443, 1969.

[39] P. S. Dodds, R. Muhamad, and D. J. Watts, "An experimental study of search in global social networks," *science*, vol. 301, no. 5634, pp. 827–829, 2003.

[40] J. Leskovec and E. Horvitz, "Planetary-scale views on a large instant-messaging network," in *Proceedings of the 17th international conference on World Wide Web*, pp. 915–924, ACM, 2008.

[41] D. J. Watts, *Small worlds: the dynamics of networks between order and randomness*. Princeton university press, 1999.

[42] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[43] D. J. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, no. 3683, pp. 510–515, 1965.

[44] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[45] A.-L. Barabási, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A: Statistical Mechanics and its Applications*, vol. 281, no. 1, pp. 69–77, 2000.

[46] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[47] M. E. J. Newman, "Assortative mixing in networks," *Physical review letters*, vol. 89, no. 20, p. 208701, 2002.

[48] D. Garlaschelli and M. I. Loffredo, "Patterns of link reciprocity in directed networks," *Physical Review Letters*, vol. 93, no. 26, p. 268701, 2004.

[49] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[50] E. Ravasz and A.-L. Barabási, "Hierarchical organization in complex networks," *Physical Review E*, vol. 67, no. 2, p. 026112, 2003.

[51] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[52] M. E. J. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[53] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.

[54] P. Hispanus and I. Bochénski, *Summulae logicales*. Marietti, 1947.

[55] S. Strange, *Porphyry: On Aristotle Categories*. A&C Black, 2014.

[56] R. H. Richens, "Preprogramming for mechanical translation," *Mechanical Translation*, vol. 3, no. 1, pp. 20–25, 1956.

[57] M. Masterman, "Semantic message detection for machine translation, using an interlingua," in *Proc. 1961 International Conf. on Machine Translation*, pp. 438–475, 1961.

[58] S. Ceccato and E. Maretti, "Linguistic analysis and programming for mechanical translation (mechanical translation and thought)," tech. rep., DTIC Document, 1960.

[59] R. F. Simmons, *Synthetic language behavior*. System Development Corporation, 1963.

[60] M. R. Quillian, "A notation for representing conceptual information: An application to semantics and mechanical english paraphrasing, sp-1395," *System Development Corporation, Santa Monica*, 1963.

[61] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[62] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.

[63] D. Levary, J.-P. Eckmann, E. Moses, and T. Tlusty, "Loops and self-reference in the construction of dictionaries," *Physical Review X*, vol. 2, no. 3, p. 031018, 2012.

[64] C. K. Ogden, *The basic dictionary*. Kegan Paul Trench, Trubner & Company Limited, 1932.

[65] M. Hart, *Project gutenberg*. Project Gutenberg, 1971.

[66] D. Harper, "Online etymology dictionary. 2001," *Availabe from: www. etymonline. com/index. php*, 2007.

[67] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber, "The university of south florida free association, rhyme, and word fragment norms," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 3, pp. 402–407, 2004.

[68] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[69] L. Rayleigh, "The problem of the random walk," *Nature*, vol. 72, no. 1866, p. 318, 1905.

[70] G. Pólya, "Über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im strassennetz," *Mathematische Annalen*, vol. 84, no. 1, pp. 149–160, 1921.

[71] P. Diaconis, "Group representations in probability and statistics," *Lecture Notes-Monograph Series*, pp. i–192, 1988.

[72] A. Einstein, "Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen," *Annalen der physik*, vol. 322, no. 8, pp. 549–560, 1905.

[73] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *The journal of political economy*, pp. 637–654, 1973.

[74] R. Lande, "Natural selection and random genetic drift in phenotypic evolution," *Evolution*, pp. 314–334, 1976.

[75] D. L. Hartl, A. G. Clark, and A. G. Clark, *Principles of population genetics*, vol. 116. Sinauer associates Sunderland, 1997.

[76] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[77] D. Aldous and J. A. Fill, "Reversible markov chains and random walks on graphs," 2002. Unfinished monograph, recompiled 2014, available at `http://www.stat.berkeley.edu/~aldous/RWG/book.html`.

[78] T. J. Berners-Lee, "The world-wide web," *Computer Networks and ISDN Systems*, vol. 25, no. 4, pp. 454–459, 1992.

[79] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1, pp. 107–117, 1998.

[80] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," 1999.

[81] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[82] D. Vise, "The google story," *Strategic Direction*, vol. 23, no. 10, 2007.

[83] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating pagerank computations," in *Proceedings of the 12th international conference on World Wide Web*, pp. 261–270, ACM, 2003.

[84] M. Bianchini, M. Gori, and F. Scarselli, "Inside pagerank," *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 1, pp. 92–128, 2005.

[85] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet Mathematics*, vol. 1, no. 3, pp. 335–380, 2004.

[86] P. Berkhin, "A survey on pagerank computing," *Internet Mathematics*, vol. 2, no. 1, pp. 73–120, 2005.

[87] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in pagerank computation: When one iteration is sufficient," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 890–904, 2007.

[88] P. Chen, H. Xie, S. Maslov, and S. Redner, "Finding scientific gems with google's pagerank algorithm," *Journal of Informetrics*, vol. 1, no. 1, pp. 8–15, 2007.

[89] N. Ma, J. Guan, and Y. Zhao, "Bringing pagerank to the citation analysis," *Information Processing & Management*, vol. 44, no. 2, pp. 800–810, 2008.

[90] Y. Ding, E. Yan, A. Frazho, and J. Caverlee, "Pagerank for ranking authors in co-citation networks," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 11, pp. 2229–2243, 2009.

[91] R. Mihalcea, P. Tarau, and E. Figa, "Pagerank on semantic networks, with application to word sense disambiguation," in *Proceedings of the 20th international conference on Computational Linguistics*, p. 1126, Association for Computational Linguistics, 2004.

[92] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 33–41, Association for Computational Linguistics, 2009.

[93] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 261–270, ACM, 2010.

[94] B. Hajian and T. White, "Modelling influence in a social network: Metrics and evaluation," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pp. 497–500, IEEE, 2011.

[95] J. Chen, B. J. Aronow, and A. G. Jegga, "Disease candidate gene identification and prioritization using protein interaction networks," *BMC bioinformatics*, vol. 10, no. 1, p. 73, 2009.

[96] G. Iván and V. Grolmusz, "When the web meets the cell: using personalized pagerank for analyzing protein interaction networks," *Bioinformatics*, vol. 27, no. 3, pp. 405–407, 2011.

[97] T. H. Haveliwala, "Topic-sensitive pagerank," in *Proceedings of the 11th international conference on World Wide Web*, pp. 517–526, ACM, 2002.

[98] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th international conference on World Wide Web*, pp. 271–279, ACM, 2003.

[99] F. Chung and W. Zhao, "Pagerank and random walks on graphs," in *Fete of combinatorics and computer science*, pp. 43–62, Springer, 2010.

[100] N. Ide and J. Véronis, "Introduction to the special issue on word sense disambiguation: the state of the art," *Computational linguistics*, vol. 24, no. 1, pp. 2–40, 1998.

[101] R. Albert, H. Jeong, and A.-L. Barabási, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[102] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," *Computer networks*, vol. 33, no. 1, pp. 309–320, 2000.

[103] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, "On near-uniform url sampling," *Computer Networks*, vol. 33, no. 1, pp. 295–308, 2000.

[104] P. Rusmevichientong, D. M. Pennock, S. Lawrence, and C. L. Giles, "Methods for sampling pages uniformly from the world wide web," in *AAAI Fall Symposium on Using Uncertainty Within Computation*, pp. 121–128, 2001.

[105] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *Proceedings of the 16th international conference on World Wide Web*, pp. 835–844, ACM, 2007.

[106] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, ACM, 2007.

[107] S. Ye, J. Lang, and F. Wu, "Crawling online social graphs," in *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pp. 236–242, IEEE, 2010.

[108] S. A. Catanese, P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Crawling facebook for social network analysis purposes," in *Proceedings of the international conference on web intelligence, mining and semantics*, p. 52, ACM, 2011.

[109] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1872–1892, 2011.

[110] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 631–636, ACM, 2006.

[111] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On unbiased sampling for unstructured peer-to-peer networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 2, pp. 377–390, 2009.

[112] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[113] R. E. Korf, "Depth-first iterative-deepening: An optimal admissible tree search," *Artificial intelligence*, vol. 27, no. 1, pp. 97–109, 1985.

[114] M. Kurant, A. Markopoulou, and P. Thiran, "On the bias of bfs," *arXiv preprint arXiv:1004.1729*, 2010.

[115] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, "Introducing markov chain monte carlo," *Markov chain Monte Carlo in practice*, vol. 1, p. 19, 1996.

[116] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz, "Approximating aggregate queries about web pages via random walks," 2000.

[117] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?," in *WWW '10: Proceedings of the 19th international conference on World wide web*, (New York, NY, USA), pp. 591–600, ACM, 2010.

Chapter 4

The Topology of Semantic Knowledge

This work was done in collaboration with Jean-Pierre Eckmann, Christian Scheible and Hinrich Schütze. It appeared in EMNLP 2013: Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA.

4.1 Abstract

Studies of the graph of dictionary definitions (DD) [1, 2] have revealed strong semantic coherence of local topological structures.

The techniques used in these papers are simple and the main results are found by understanding the structure of cycles in the directed graph (where words point to definitions). Based on our earlier work [2], we study a different class of word definitions, namely those of the Free Association (FA) dataset [3]. These are responses by subjects to a cue word, which are then summarized by a directed, free association graph.

We find that the structure of this network is quite different from both the Wordnet and the dictionary networks. This difference can be explained by the very nature of free association as compared to the more "logical" construction of dictionaries. It thus sheds some (quantitative) light on the psychology of free association.

In NLP, semantic groups or clusters are interesting for various applications such as word sense disambiguation. The FA graph is tighter than the DD graph, because of the large number of triangles. This also makes drift of meaning quite measurable so that FA graphs provide a quantitative measure of the semantic coherence of small groups of words.

4.2 Introduction

The computer study of semantic networks has been around since the advent of computers [4] and has been used to study semantic relations between concepts and for analyzing semantic data. Traditionally, a popular lexical database of English is Wordnet [5, 6], which organizes the semantic network in terms of graph theory. In contrast to manual approaches, the automatic analysis of semantically interesting graph structures of language has received increasing attention. For example, it has become clear more recently that cycles and triangles play an important role in semantic networks, see *e.g.*, [7]. These results suggest that the underlying semantic structure of language may be discovered through graph-theoretical methods. This is in line with similar findings in much wider realms than NLP [8].

In this paper, we compare two different types of association networks. The first network is constructed from an English dictionary (DD), the second from a free association (FA) database [3]. We represent both datasets through directed graphs. For DD, the nodes are words and the directed edges point from a word to its definition(s). For FA, the nodes are again words, and each cue word has a directed edge to each association it elicits.

Although the links in these graphs were not constructed by following a rational centralized process, their graph exhibits very specific features and we concentrate on the study of its topological properties. We will show that these graphs are quite different in global and local structure, and we interpret this as a reflection of the different nature of DD vs. FA. The first is an *objective* set of relations between words and their meaning, as explained by other words, while the second reveals the nature of *subjective* reactions to cue words by individuals. This matter of fact is reflected by several quantitative differences in the structure of the corresponding graphs.

The main contribution of this paper is an empirical analysis of the way semantic knowledge is structured, comparing two different types of association networks (DD and FA). We conduct a mathematical analysis of the structure of the graphs to show that the way humans express their thoughts exhibits structural properties in which one can find semantic patterns. We show that a simple graph-based approach can leverage the information encoded in free association to narrow down the ambiguity of meaning, resulting in precise semantic groups. In particular, we find that the main strongly connected component of the FA graph (the so-called **core**) is very cyclic in nature and contains a large predominance of short cycles (*i.e.*, co-links and triangles). In contrast to the DD graph, bunches of triangles form well-delimited lexical fields of collective semantic knowledge. This property may be promising for downstream tasks. Further, the methods developed in this paper may be ap-

plicable to graph representations that occur in other problems such as word sense disambiguation (*e.g.*, [9, 10]) or sentiment polarity induction [11, 12].

To show the semantic coherence of these lexical fields of the FA graph, we perform an experiment with human raters and find that cycles are strongly semantically connected even when compared to close neighbors in the graph.

The reader might wonder why sets of pairwise associations can lead to any interesting structure. One of the deep results in graph theory, [13], is that in sparse graphs, *i.e.*, in graphs with few links per node, the number of triangles is extremely rare. Therefore, if one does find many triangles in a graph, they must be not only a signal of non-randomness, but carry relevant information about the domain of research as shown earlier [8].

4.3 The USF FA dataset

This dataset is one of the largest existing databases of free associations (FA) and has been collected at the University of South Florida since 1973 by researchers in psychology [3]. Over the years, more than 6 000 participants produced about 750 000 responses to 5 019 stimulus words.

The procedure for collecting the data is called discrete association task and consists in asking participants to give the first word that comes to mind (**target**) when presented a stimulus word (**cue**).

For creating the initial set of stimulus words, the Jenkins and Palermo word association norms [14] proved useful but too limited as they consist of only 200 words. For this reason, additional words have been regularly added to the pool of normed words, unfortunately without well established rules being followed. For instance, some were selected as potentially interesting cues, some were added as responses to the first sets of cues and, some others were collected for supporting new studies on verbs. We still work with this database, because of its breadth.

The final pool of stimuli comprises 5 019 words of which 76% are nouns, 13% adjectives, and 7% verbs. A word association is said to be **normed** when the target is also part of the set of norms, *i.e.*, a cue. The USF dataset of free associations contains 72 176 cue-target pairs, 63 619 of which are normed. As an example, the association *puberty-sex* is normed whereas the association *puberty-thirteen* is not, because *thirteen* is not a cue.

4.4 Mathematical definitions

We collect here those notions we need for the analysis of the data.

A **directed graph** is a pair $G = (V, E)$ of a set V of **vertices** and, a set E of ordered pairs of vertices also called **directed edges**. For a directed edge $(u, v) \in E$, u is called the **tail** and v the **head** of the edge. The number of edges incident to a vertex $v \in V$ is called the **degree** of v. The **in-degree** (resp. **out-degree**) of a vertex v is the number of edge heads (resp. edge tails) adjacent to it. A vertex with null in-degree is called a **source** and a vertex with null out-degree is called a **sink**.

A **directed path** is a sequence of vertices such that a directed edge exists between each consecutive pair of vertices of the graph. A directed graph is said to be **strongly connected**, (resp. **weakly connected**) if for every pair of vertices in the graph, there exists a directed path (resp. undirected path) between them. A **strongly connected component**, SCC, (resp. **weakly connected component**, WCC) of a directed graph G is a maximal strongly connected (resp. weakly connected) subgraph of G.

A **directed cycle** is a directed path such that its **start vertex** is the same as its **end vertex**. A **co-link** is a directed cycle of length 2 and a **triangle** a directed cycle of length 3.

The **distance** between two vertices in a graph is the number of edges in the shortest path connecting them. The **diameter** of a graph G is the greatest distance between any pair of vertices. The **characteristic path length** is the average distance between any two vertices of G.

The **density** of a directed graph $G(V, E)$ is the proportion of existing edges over the total number of possible edges and is defined as:

$$d = |E|/(|V|(|V| - 1))$$

The **neighborhood** N_i of a vertex v_i is $N_i = \{v_j : e_{ij} \in E \text{ or } e_{ji} \in E\}$.

The **local clustering coefficient** C_i for a vertex v_i corresponds to the density of its neighborhood subgraph. For a directed graph, it is thus given by:

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{|N_i|(|N_i| - 1)}$$

The **clustering coefficient** of a graph G is the average of the local clustering coefficients of all its vertices.

The **efficiency** Eff of a directed graph G is an indicator of the traffic capacity of a network. It is the harmonic mean of the distance between any two vertices of G. It is defined as:

$$\text{Eff} = \frac{1}{|V|(|V| - 1)} \sum_{i \neq j \in V} \frac{1}{d_{ij}}$$

The linear correlation coefficient between two random variables X and Y is defined as:

$$\rho(X,Y) = (E[XY] - \mu_X \mu_Y)/(\sigma_X \sigma_Y)$$

where μ_X and σ_X are respectively the mean and standard deviation of the random variable X.

The linear degree correlation coefficient of a graph is called **assortativity** and is expressed as:

$$\rho_D = \sum_{xy} xy(e_{xy} - a_x b_y)/(\sigma_a \sigma_b)$$

where e_{xy} is the fraction of all links that connect nodes of degree x and y and where a_x and b_y are respectively the fraction of links whose tail is adjacent to nodes with degree x and whose head is adjacent to nodes with degree y, satisfying the following three conditions:

$$\sum_{xy} e_{xy} = 1, a_x = \sum_{y} e_{xy}, b_y = \sum_{x} e_{xy}$$

When ρ_D is positive, the graph possesses **assortative mixing** and high-degree nodes tend to connect to other high-degree nodes. On the other hand, when ρ_D is negative, the graph features **disassortative mixing** and high-degree nodes tend to connect to low degree nodes.

The **intersection graph** of sets A_i, $i = 1, \ldots, m$, is constructed by representing each set A_i as a vertex $v_i \in V$ and adding an edge for each pair of sets with a non-empty intersection:

$$E = \{(v_i, v_j) : A_i \cap A_j \neq \varnothing\}$$

4.5 Graph topology analysis

4.5.1 Graph generation

Our goal being to study the FA network topology, we first concentrate on the generation of an unweighted directed graph. We generate the corresponding graph by adding a directed edge for each cue-target pair of the dataset. We only consider pairs whose target was normed in order to avoid overloading the graph with noisy data (*e.g.*, a response meaningful only to a specific participant). The graph has 5 019 vertices and 63 619 edges. It is composed of a single WCC and 166 SCCs.

For comparison with dictionary definitions (DD), we construct a graph from the Wordnet2 dictionary (nouns only), following [2]. This graph contains 54 453 vertices and 179 848 edges.

4.5.2 Core extraction

The so-called **core** was defined previously in [1, 2] as that subset of nodes in which a random walker gets trapped after only a few steps.

The shave algorithm was used in [2] to isolate this subset. It consists in recursively removing the source and sink nodes from a weakly connected directed graph and permits to get the subgraph induced by the union of its strongly connected components. Note that the dictionary graph (DD) has no sinks (*i.e.*, words that never get defined) and that it contains a giant SCC whose size is comparable to the one of the initial graph.

It turns out that the FA graph also contains a giant SCC, therefore getting the core consists more simply in extracting the main SCC of the initial graph. We use Tarjan's algorithm [15] for isolating the FA core.

4.5.3 Vertex degree analysis

The FA core has a maximum in-degree of 313, a maximum out-degree of 33 and an average degree of 12.71. The in-degree distribution follows a power law ($\gamma = 1.93$) and the out-degrees are Poisson-like distributed with a peak at 14 [16, 17].

Words having a high in-degree are **targets** that tend to be cited more frequently. On the other hand, words having a high out-degree are **cues** that evoke many different targets.

The most evocative cues are, in decreasing order of out-degree: *field* (33), *body* (31), *condemn* (29), *farmer* (29), *crisis* (28), *plan* (28), *attention* (27), *animal* (27), and *hang* (27).

Interestingly, the most cited targets (*i.e.*, targets with highest in-degree) are in decreasing order: *food* (313), *money* (295), *water* (271), *car* (251), *good* (246), *bad* (221), *work* (187), *house* (183), *school* (182), *love* (179).

4.5.4 Cycle decomposition of the core

We define the **vertex k-cycle multiplicity** (resp. **edge k-cycle multiplicity**) as the number of k-cycles a given vertex (resp. edge) belongs to. We call **core-ER** the set of Erdös-Rényi (ER) random graphs $G(n, M)$ having the same number of nodes and the same number of edges as the FA core. We start by extracting the 2- and 3-cycles by using a customized version of

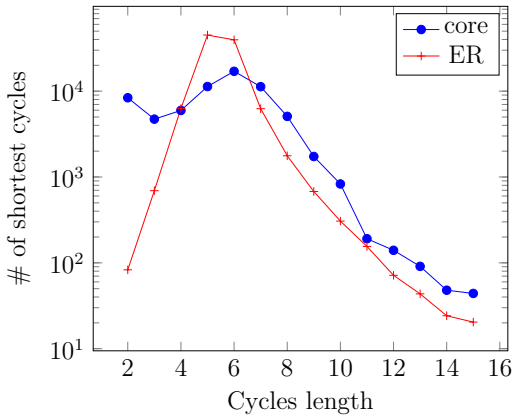

Figure 4.1 – Distribution of shortest cycles lengths in the core compared to equivalent ER models
One should bear in mind that we only consider the set of *shortest* cycles. Thus, a k-cycle is not counted if each of its nodes belongs to a cycle whose length is $< k$. Although the number of 4-shortest cycles is comparable in the core and core-ER graphs for example, there are in reality far more 4-cycles in the core (*i.e.*, 42 738 versus 6 517). We see that when considering shortest cycles, short cycles tend to hide long ones, and, as a large proportion of nodes in the core belong to 2- and 3-cycles, many longer cycles do not get counted at all.

Johnson's algorithm [18]. The first thing we observe is that the core has a very high density of short cycles: the subset of nodes belonging to 2-cycles (*i.e.*, nodes with 2-cycle multiplicities > 0) cover 95% of the core vertices and the 3-cycles cover 88% of the core vertices. The corresponding core-ER graphs have on average about 100 times fewer 2-cycles and almost 20 times fewer 3-cycles.

This shows that the core is very cyclic in nature and that it remains very well connected for short-length cycles: most vertices of the core indeed belong to at least one co-link or triangle.

In order to limit computation times, we only considered shortest cycles for lengths ≥ 3 and analyzed the distribution of the number of shortest cycles in the core compared to equivalent random graphs. Whereas there are many more short cycles in the core, we observe a predominance of 4, 5 and 6-cycles in core-ER graphs. However, we find again a slight predominance of long cycles (length between 7 and 15) in the core (see Fig. 4.1). See [2], Fig. 3, where the cycle distribution is very different, with a minimum at length 5.

4.5.5 Interpretation of cycles

2-cycles are composed of concretely related words (*e.g.*, *drug-coke*, *destiny-fate*, *einstein-genius*, ...). The vertex with highest 2-cycle multiplicity is *music* (22).

Words in 3- and 4-cycles often belong to the same lexical field. Examples of 3-cycles: *protect-guard-defend* or *space-universe-star*. The vertex (resp. edge) with highest 3-cycle multiplicity is *car* (86) (resp. *bad-crime* (11)). Examples of 4-cycles: *monster-dracula-vampire-ghost* or *flu-virus-infection-sick*.

Longer cycles are more difficult to describe: Relations linking words of a given cycle exhibit semantic drift with increasing length (cf. [2]). Examples of 5-cycles: *yellow-coward-chicken-soup-noodles* and *sleep-relax-music-art-beauty*.

The cumulated set of free associations reflects the way in which a group of people retrieved its semantic knowledge. As the associated graph is highly circular, this suggests that this knowledge is not stored in a hierarchical way [16]. The large predominance of short cycles in the core may indeed be a natural consequence of the semantic information being acquired by means of associative learning [19, 20].

4.5.6 FA core clustering

The walktrap community algorithm

Complex networks are globally sparse but contain locally dense subgraphs. These groups of highly interconnected vertices are called **communities** and convey important properties of the network.

Although the notion of community is difficult to define formally, the current consensus establishes that a partition $P = \{C_1, C_2, \ldots, C_k\}$ of the vertex set of a graph G represents a good community structure if the proportion of edges inside the C_i is higher than the proportion of edges between them [21].

Computing such communities in a large graph is generally computationally expensive [22]. We use the so-called 'Walktrap' community detection algorithm [23] for extracting communities from the FA networks. The idea lying behind this algorithm is that random walks on a graph will tend to get trapped in the densely connected subgraphs.

Let P_{ij}^t be the probability of going from vertex i to vertex j through a random walk of length t. The distance between two vertices i and j of the graph is defined as:

$$r_{ij}(t) = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}}$$

where $d(k)$ is the degree of vertex k.

One defines the probability $P_{C,j}^t$ to go from community C to vertex j in t steps: $P_{C,j}^t = \sum_{i \in C} P_{ij}^t / |C|$, and then the distance is easily generalized for two communities C_1, C_2.

The algorithm starts with a partition $P_1 = \{\{v\} \in V\}$ of the initial graph into n communities each of which is a single vertex. At each step, two communities are chosen and merged according to the criterion described below and the distances between communities are updated. The process goes on until we obtain the partition $P_n = \{V\}$.

In order to reduce complexity, only adjacent communities are considered for merging. The decision is then made according to Ward's method [24]: at each step k, the two communities that minimize the mean σ_k of the squared distances between each vertex and its community are merged:

$$\sigma_k = \frac{1}{n} \sum_{C \in P_k} \sum_{i \in C} r_{iC}^2$$

Clustering of the core

We first identify the communities of the FA core using the Walktrap algorithm. We immediately observe that when the path length parameter increases, the number of identified communities decreases (*i.e.*, for a length of 2, we find 35 communities whereas for a length of 9, we only find 8 communities).

For a path length of 2, the algorithm extracts 35 communities, 7 of which contain more than 100 vertices, 3 of which contain between 100 and 50 vertices and 25 of which contain less than 50 vertices.

We observe that for most small communities (*i.e.*, the ones containing less than 50 vertices), there exists a clear relation between the labels of their vertices. Typically, the labels are part of the same lexical field (*e.g.*, all the planets (except *earth*) or related by a common grammatical function (such as *why*, *where*, *what*, ...).

Clustering of the core co-links

We define the **k-cycle induced subgraph** of a graph G as the subgraph of G induced by the set of its vertices with k-cycle multiplicity > 0.

The **co-link graph** of a graph $G(V, E)$ is the undirected graph obtained by replacing each co-link (*i.e.*, 2-cycle) of the 2-cycle induced subgraph of G by a single undirected edge and removing all other edges.

The co-link graph of the FA core has 4 508 vertices and 8 309 edges for a density of 8×10^{-4}. It is composed of a single weakly connected component that can be seen as a projection of the strongest semantic links from the original graph. Extracting the co-link graph is thus an efficient way of selecting the set of most important semantic links (*i.e.*, the set of 2-cycles that appear in large predominance in the core compared to what is found in an equivalent random graph) while filtering out the noisy or negligible ones.

The sets of communities extracted by the Walktrap algorithm exhibit different degrees of granularity depending on the length parameter. For short paths, a large number of very small communities are returned (*e.g.*, 923 communities when length equals 2) whereas for longer paths the average size of the communities increases more and more.

The community detection exhibits thus a far finer degree of granularity for the core co-links graph than for the core itself. The size of the communities being much smaller in average, it is striking to notice to which extent the words of a given community are semantically related.

Examples of communities found in the core co-links graph include (*standards*, *values*, *morals*, *ethics*), (*hopeless*, *romantic*, *worthless*, *useless*), (*thesaurus*, *dictionary*, *vocabulary*, *encyclopedia*) or (*molecule*, *atom*, *electron*, *nucleus*, *proton*, *neutron*).

DD core clustering vs FA core clustering

The clustering of both cores has very different characteristics: We illustrate the neighborhoods of *conflict* for both cases in Fig. 4.2 and 4.3.

On one hand, the words in communities of the DD core are in most cases either synonyms, *e.g.*, *(declaration, assertion, claim)* or an instance-of kind of relation, *e.g.*, *(signal, gesture, motion)* or *(zero, integer)*.

On the other hand, communities of the FA core are generally composed of words belonging to the same lexical field and sharing the same level of abstraction.

Moreover, we notice that it is often difficult to establish the semantic relation existing between words of many small communities (*i.e.*, containing less than 10 words) of the DD core. Two such examples are: (*choice*, *probate*, *executor*, *chosen*, *certificate*, *testator*, *will*) and (*numeral*, *monarchy*, *monarch*, *crown*, *significance*, *autocracy*, *symbol*, *interpretation*).

The comparison of DD and FA reveals, in a quantitative way, fundamental differences between the two realms. The interesting data are shown in

Figure 4.2 – Neighborhood of *conflict* in the FA core
The set of words belonging to the neighborhood of *conflict* are clearly part of the same lexical field. The high density of co-links leads to cyclicity and we see that many directed triangles are present in the local subgraph (*e.g.*, *conflict-trouble-fight, conflict-argument-disagree*). We can even find triangles of co-links that link together words semantically strongly related (*e.g.*, *fight-war-battle, fight-quarrel-argument*). Nodes that are part of the neighborhood of *conflict* in both FA and DD are in empty circles.

table 4.1.

Note that while the FA core is in fact larger than the DD core, its diameter is smaller. This illustrates in a beautiful way the nature of free association as compared to the more neutral dictionary. In particular, the characteristic path length is smaller in the FA graph, because humans use generalized event knowledge [25] in free association, producing semantic shortcuts. For example, FA contains a direct link *mirage→water*, whereas in DD, the shortest path between the two words is *mirage→refraction→wave→water*.

4.6 The Bricks of Meaning

4.6.1 Extraction of the seed

We already saw that most vertices of the core belong to directed 2- and 3-cycles. Whereas 2-cycles establish strong semantic links (*i.e.*, synonymy or antonymy relations) and provide cyclicity to the underlying directed graph, we claim that 3-cycles (*i.e.*, triangles) form the set of elementary concepts of

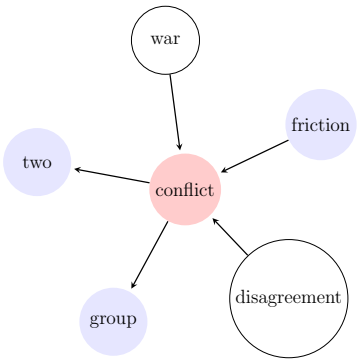

Figure 4.3 – Neighborhood *conflict* in the DD core
First, we note that the neighborhood has a lower density than in the FA core. We also see that there is no cycle and there seems to be a flow going from source nodes to sink nodes. As it generally happens in the neighborhood subgraphs of the DD core, source nodes are rather specific words whereas sink nodes are generic words.

the core.

These structues are common to DD and to FA, but we will see that the links in FA are somehow more direct than in DD.

We call **seed** the subgraph of the core induced by the set V_3 of vertices belonging to directed triangles and **shell** the subgraph of the core induced by the set $V \backslash V_3$ (*i.e.*, the set of vertices with a null 3-cycle multiplicity), see Fig. 4.4.

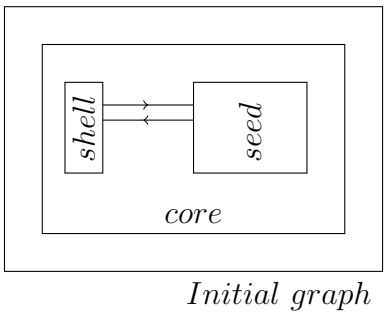

Figure 4.4 – Composition of the FA graph
The graph of FA contains a giant SCC (the core). The subgraph of the core induced by the set of nodes belonging to at least one triangle also forms a giant component we call the 'seed'. The subgraph of the core induced by the set of nodes not belonging to any triangle is called the 'shell' and is composed of many small SCCs, including single vertices. Although the shell has a low density, its nodes are very well connected to the seed.

The shell contains 530 nodes and 309 edges. There are 7 035 edges connecting the shell to the seed. The shell consists of many small SCCs and

	FA core	DD core
# vertices	4 843	1 496
# edges	61 544	4 766
density	2.5×10^{-3}	2.1×10^{-3}
avg degree	12.71	3.21
max in-degree	313	59
directed diameter	10	29
characteristic path length	4.26	10.42
efficiency	2.5×10^{-1}	1.2×10^{-1}
clustering coefficient	8.5×10^{-2}	5.1×10^{-2}
assortativity	5.5×10^{-2}	6.1×10^{-2}

Table 4.1 – Comparison FA vs DD

although its average degree is low (1.17), its vertices have on average many (13.27) connections to the seed.

The seed contains 4 313 vertices (89% of the core) and 54 197 edges. The first thing to notice is that it has 100 times more co-links (7 895) and 20 times more triangles (13 119) than an equivalent random graph. We call **shortcuts** the 32 773 edges of the seed that do not belong to 3-cycles, see Fig. 4.5.

The seed obviously also contains cycles whose length is greater than 3. One can check that there exist only 5 basic motifs involving 2 attached triangles and 1 shortcut for creating 4- and 5-cycles, and that linking 2 isolated triangles with 2 shortcuts also permit to form 4-, 5- and 6-cycles. All longer cycles are simply made of a juxtaposition of these basic motifs.

Furthermore, there is a limit on the number of shortcuts that can possibly be added in the seed before it gets saturated, as all its vertices belong to at least one triangle. We show that at most 16 shortcuts can be added between two isolated triangles, at most 6 between 2 triangles sharing 2 vertex and at most 2 between 2 triangles sharing 2 vertices (see Fig. 4.5).

4.6.2 The elementary lexical fields

Once the seed is isolated, we go on digging into its structure. We focus on the arrangements of triangles as they constitute the set of elementary concepts.

We start by removing all shortcuts from the seed and convert it then to an undirected graph, in order to get a homogeneous simplicial 2-complex.

Let $\mathbf{t}$ be the graph operator which transforms a graph G into the intersection graph $\mathbf{t}G$ of its 2-simplices (*i.e.*, triangles sharing an edge). We

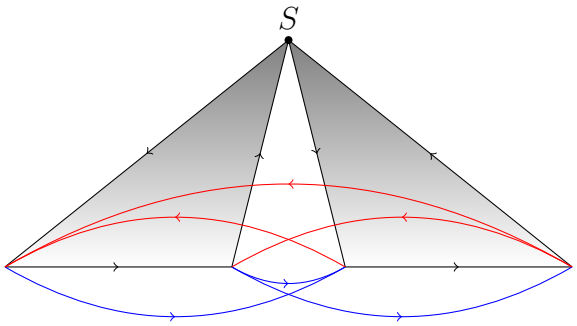

Figure 4.5 – Shortcut edges between two triangles sharing a single vertex S
Two triangles can share 0, 1 or 2 vertices. For each of these three basic motifs, we count the maximum number of shortcut edges (*i.e.*, edges not belonging to 3-cycles) that can be added. By linking two triangles, these shortcuts permit to move two basic semantic units closer together and create longer cycles (*i.e.*, 4, 5, and 6-cycles). Long cycles can be thus considered as groupings of basic semantic units. In the case of two triangles sharing one vertex for example, it is possible to add at most 6 shortcuts, whereas, for two triangles sharing two vertices, at most 2 shortcuts can be added.

apply **t** to the homogeneous simplicial 2-complex found previously. The result represents the links between the basic semantic units of the seed. We call **seed-crux** the giant WCC in the intersection graph.

We enumerate the 8 380 maximal cliques of FA seed-crux and get the list of words composing each clique. By removing the ones that are subsets of bigger lists, we finally obtain 3 577 lists of words .

These lists of words have a rather small and homogeneous size (between 4 and 17) and 95% have a size comprised between 4 and 10. More interestingly, they clearly define well-delimited lexical fields. We will show this through two experiments in the following sections. A few examples include *(honest, trustworthy, reliable, responsible)*, *(stress, problem, worry, frustration)* and *(data, process, computer, information)*.

From a topological perspective, we deduce that bunches of triangles (*i.e.*, cliques of elementary concepts) span the seed in a homogeneous way. These bunches form a set of cohesive lexical fields and constitute essential bricks of semantic knowledge.

4.6.3 Semantic similarity of the lexical fields

In order to quantify the relative meaning of words in the lexical fields of the seed-crux, we define the following semantic similarity metric based on the Wordnet *WUP* metric [26] for a given set of words L:

Distance	Acc	κ	KS
original	–	0.404	30
1	74	0.522	42
2	97	0.899	89
∞	99	0.899	89

Table 4.2 – Accuracy, κ, and $count(p < 0.05)$ for KS

$$S_\ell(L) = 2 \sum_{w_i, w_j \in L, w_i \neq w_j} S_w(w_i, w_j)/(|L|(|L|-1))$$

where $S_w(w_i, w_j) = \max_{S_k \ni w_i \text{and} S_\ell \ni w_j} \{wup(S_k, S_\ell)\}$ and wup is the WUP semantic metric and S_k and S_ℓ are Wordnet synsets.

The average value of S_ℓ for the set of cliques of seed-crux is 0.6 whereas it is only 0.43 for randomly sampled set of words. This suggests the corresponding lists of words are indeed semantically related. We will show the strength of this relation in the following experiment with human raters.

4.6.4 Human evaluation of the lexical fields

To validate our findings, we conducted an empirical evaluation through human annotators. Starting from the 1 204 4-groups, we designed the following experiment: We corrupt the groups by exchanging one of the 4 elements with a randomly chosen word at a distance from the group of 1, 2, and "infinity" (*i.e.*, any word of the whole core). We presented 100 random samples for each of the 3 distances as well as 100 unperturbed groups (*original*) to annotators at Amazon Mechanical Turk [1], asking which word fits the group the least. Intuitively, the closer the randomly chosen words get to the group, the closer the distribution of the votes for each sample should be to the uniform distribution. We collected 10 votes for each of the 4 problems of 100 random samples. We calculated accuracy (*i.e.*, the relative frequency of correctly identified random words) for the 3 random confounder experiments and Fleiss' κ. Further, we used the Kolmogorov-Smirnov (KS) test for how uniform the label distribution is, reporting the relative frequency of samples that are significantly ($p < 0.05$) different from the uniform distribution. The results of this experiment are summarized in Table 4.2 and show clearly that the certainty about the "odd man out" increases together with the distance.

1. `http://www.mturk.com`

4.6.5 Error analysis

If we view our results as a resource for a downstream task, it is important to know about possible downsides. First, we note that there are words which are not in a triangle and will thus be missing in the intersection graph. This is an indication that the corresponding word is less well embedded contextually, so conversely, any prediction made about it from the data may be less reliable. Additionally, semantic leaps caused by generalized event knowledge may lead to lesser-connected groups such as (*steel*, *pipe*, *lead*, *copper*). Jumps like these may or may not be desired in a subsequent application.

4.7 The Case of the EAT FA dataset

The Edinburgh Associative Thesaurus (EAT) [27] is a large dataset of free associations. We extract the EAT FA seed-crux with the previously described methods.

We start by generating the initial graph (23 219 vertices and 325 589 edges), then extract its core (7 754 vertices and 247 172 edges) and its seed (7 500 vertices and 238 677 edges). It is interesting to notice at this stage that the EAT seed contains 74% of the words belonging to the USF seed. After generating the seed-crux which contains 63 363 vertices, 6 825 731 edges, and 342 490 maximal cliques, we finally obtain 40 998 lists of words.

These lists comprise between 4 and 233 words but 80% of them have a relatively small size between 4 and 20. Although we find exceptions for this graph, most of the extracted lists

again form well-delimited lexical fields (*e.g.*, *(health, resort, spa, bath, salts)* or *(god, devil, angel, satan)*.

Comparing the two association experiments, we see that the local topologies are quite similar. Both FA cores have a high density of connected triangles, whereas cycles in the DD graph tend to be longer and most triangles are isolated. This can be attributed to the different ways in which DD and FA are obtained, the former being built rationally by following a humanly-driven process and the latter reflecting an implicit collective semantic knowledge.

4.8 Related Work

A number of metrics like *Latent Semantic Analysis* [28] and *Word Association Spaces* [29] have been recently developed for quantifying the relative meaning of words. As the topological properties of free association graphs reflect key aspects of semantic knowledge, we believe some graph theory

metrics could be used efficiently to derive new ways of measuring semantic similarity between words.

Topological analysis of the Florida Word Associations (FA) was started by [16, 17], who extracted global statistics. We follow the basic methodology of these studies, but extend their approach. First, we conduct deeper analyses by examining the neighborhood of nodes and extracting the statistics of cycles. Second, we compare the properties of FA and DD graphs.

Word clustering based on graphs has been the subject of various earlier studies. Close to our work is [30]. These authors recognize that nearest-neighbor-based clustering of co-occurrence give rise to semantic groups. This type of approach has since been applied in various modified forms, *e.g.*, by [31] who performs label-propagation based on randomized nearest neighbors, or [32] who perform greedy clustering. Hierarchical clustering algorithms (*e.g.*, [33, 34]) are related as well, however, the key difference is that in hierarchical clustering, the granularity of a cluster is difficult to determine.

[7] recognize that triangles form semantically strongly cohesive groups and apply clustering coefficients for word sense disambiguation. Their work focuses on undirected graphs of corpus co-occurrences whereas our work builds on directed associations. Building on this work, we take finer topological graph structures into account, which is one of the main contributions in this paper.

4.9 Conclusion

The cognitive process of discrete free association being an epiphenomenon of our semantic memory at work, the cumulative set of free associations of the USF dataset can be viewed as the projection of a collective semantic memory.

To analyze the semantic memory, we use the tools of graph theory, and compare it also to dictionary graphs. In both cases, triangles play a crucial role in the local topology and they form the set of elementary concepts of the underlying graph. We also show that cohesive lexical fields (taking the form of cliques of concepts) constitute essential bricks of meaning, and span the core homogeneously at the global level; 89% of all words in the core belong to at least one triangle, and 77% belong to cliques of triangles containing 4 words (i.e., pairs of triangles sharing an edge or forming tetrahedras). As the words of a graph of free associations acquire their meaning from the set of associations they are involved in [35], we go a step further by examining the neighborhood of nodes and extracting the statistics of cycles. We further check through human evaluation that the clustering is strongly related to

meaning, and furthermore, the meaning becomes measurably more confused as one walks away from a cluster.

Comparing dictionaries to free association, we find the free association graph being more concept driven, with words in small clusters being on the same level of abstraction. Moreover, we think that graphs of free associations could find interesting applications for *Word Sense Disambiguation* (*e.g.*, [9, 10]), and could be used for detecting psychological disorders (*e.g.*, depression, psychopathy) or whether someone is lying [36, 37].

Finally, we believe that studying the dynamics of graphs of free associations may be of particular interest for observing the change in meaning of certain words [38], or more generally to follow the cultural evolution arising among a social group.

Bibliography

[1] O. Picard, A. Blondin-Massé, S. Harnad, O. Marcotte, G. Chicoisne, and Y. Gargouri, "Hierarchies in dictionary definition space," in *Annual Conference on Neural Information Processing Systems*, 2009.

[2] D. Levary, J.-P. Eckmann, E. Moses, and T. Tlusty, "Loops and self-reference in the construction of dictionaries," *Phys. Rev. X*, vol. 2, p. 031018, 2012.

[3] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber, "The University of South Florida free association, rhyme, and word fragment norms," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 3, pp. 402–407, 2004.

[4] E. Brunet, "Le traitement des faits linguistiques et stylistiques sur ordinateur," *Texte d'application: Giraudoux, Statistique et Linguistique. David, J. y Martin, R.(eds.). Paris: Klincksieck*, pp. 105–137, 1974.

[5] G. A. Miller, "WordNet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[6] G. Miller and C. Fellbaum, *WordNet: An Electronic Lexical database*. Cambridge, MA: MIT Press, 1998.

[7] B. Dorow, D. Widdows, K. Ling, J.-P. Eckmann, D. Sergi, and E. Moses, "Using curvature and Markov clustering in graphs for lexical acquisition and word sense discrimination," in *MEANING-2005, 2nd Workshop organized by the MEANING Project, February 3rd-4th 2005, Trento, Italy.*, 2004.

[8] J.-P. Eckmann and E. Moses, "Curvature of co-links uncovers hidden thematic layers in the World Wide Web," *Proc. Natl. Acad. Sci. USA*, vol. 99, no. 9, pp. 5825–5829 (electronic), 2002.

[9] F. Heylighen, "Mining associative meanings from the web: from word disambiguation to the global brain," in *Proceedings of Trends in Special Language & Language Technology*, pp. 15–44, 2001.

[10] E. Agirre and A. Soroa, "Personalizing pagerank for word sense disambiguation," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, (Stroudsburg, PA, USA), pp. 33–41, Association for Computational Linguistics, 2009.

[11] A. Hassan and D. Radev, "Identifying text polarity using random walks," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 395–403, Association for Computational Linguistics, 2010.

[12] C. Scheible, "Sentiment translation through lexicon induction," in *Proceedings of the ACL 2010 Student Research Workshop*, (Uppsala, Sweden), pp. 25–30, Association for Computational Linguistics, July 2010.

[13] B. Bollobás, *Random graphs*, vol. 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge: Cambridge University Press, second ed., 2001.

[14] D. S. Palermo and J. J. Jenkins, "Word association norms: Grade school through college.," *University of Minnesota Press*, 1964.

[15] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.

[16] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth," *Cognitive Science*, vol. 29, no. 1, pp. 41–78, 2005.

[17] P. Gravino, V. D. Servedio, A. Barrat, and V. Loreto, "Complex structures and semantics in free word association," *Advances in Complex Systems*, vol. 15, no. 03n04, 2012.

[18] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM Journal on Computing*, vol. 4, no. 1, pp. 77–84, 1975.

[19] M. H. Ashcraft and G. A. Radvansky, *Cognition*. Pearson Prentice Hall, 2009.

[20] D. R. Shanks, *The psychology of associative learning*, vol. 13. Cambridge University Press, 1995.

[21] S. Fortunato, "Community Detection in Graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[22] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical review E*, vol. 80, no. 5, p. 056117, 2009.

[23] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Journal of Graph Algorithms and Applications*, pp. 284–293, Springer, 2006.

[24] B. Everitt, S. Landau, and M. Leese, "Cluster analysis. 4th Edition," *Arnold, London*, 2001.

[25] K. McRae and K. Matsuki, "People use their knowledge of common events to understand language, and do so as quickly as possible," *Language and linguistics compass*, vol. 3, no. 6, pp. 1417–1429, 2009.

[26] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational*

Linguistics, pp. 133–138, Association for Computational Linguistics, 1994.

[27] G. R. Kiss, C. Armstrong, R. Milroy, and J. Piper, "An associative thesaurus of english and its computer analysis," *The computer and literary studies*, pp. 153–165, 1973.

[28] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[29] M. Steyvers, R. M. Shiffrin, and D. L. Nelson, "Word association spaces for predicting semantic similarity effects in episodic memory," *Experimental cognitive psychology and its applications: Festschrift in honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*, pp. 237–249, 2004.

[30] D. Widdows and B. Dorow, "A graph model for unsupervised lexical acquisition," in *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, (Stroudsburg, PA, USA), pp. 1–7, Association for Computational Linguistics, 2002.

[31] C. Biemann, "Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, (Stroudsburg, PA, USA), pp. 73–80, Association for Computational Linguistics, 2006.

[32] Y. Matsuo, T. Sakaki, K. Uchiyama, and M. Ishizuka, "Graph-based word clustering using a web search engine," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, (Stroudsburg, PA, USA), pp. 542–550, Association for Computational Linguistics, 2006.

[33] I. Jonyer, D. J. Cook, and L. B. Holder, "Graph-based hierarchical conceptual clustering," *The Journal of Machine Learning Research*, vol. 2, pp. 19–43, 2002.

[34] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge University Press Cambridge, 2008.

[35] J. Deese, "On the structure of associative meaning.," *Psychological review*, vol. 69, p. 161, 1962.

[36] J. T. Hancock, M. T. Woodworth, and S. Porter, "Hungry like the wolf: A word-pattern analysis of the language of psychopaths," *Legal and Criminological Psychology*, vol. 18, no. 1, pp. 102–114, 2013.

[37] G. H. Kent and A. J. Rosanoff, *A study of association in insanity.* American Journal of Insanity, 1910.

[38] J. Deese, "Meaning and change of meaning.," *The American psychologist*, vol. 22, no. 8, p. 641, 1967.

Chapter 5

Diffusion Fingerprints - Theory & Applications

This work was done in collaboration with Jean-Pierre Eckmann and Andrea Agazzi. It was submitted to ACML2015: 7th Asian Conference on Machine Learning, Hong Kong, China.

5.1 Abstract

We introduce, test and discuss a method for classifying and clustering data modeled as directed graphs. The idea is to start diffusion processes from any subset of a data collection, generating corresponding distributions for reaching points in the network. These distributions take the form of high-dimensional numerical vectors and capture essential topological properties of the original dataset. We show how these diffusion vectors can be successfully applied for getting state-of-the-art accuracies in the problem of extracting pathways from metabolic networks. We also provide a guideline to illustrate how to use our method for classification problems, and discuss important details of its implementation. In particular, we present a simple dimensionality reduction technique that lowers the computational cost of classifying diffusion vectors, while leaving the predictive power of the classification process substantially unaltered. Although the method has very few parameters, the results we obtain show its flexibility and power. This should make it helpful in many other contexts.

5.2 Introduction

Our method combines several ideas that can be applied in a flexible way, either for classification or graph mining. By using a graph modeling of associations within a data collection, we generate "fingerprints" for any subset of data items in the form of high-dimensional distribution vectors of a random walk diffusion process. When such a graph is not provided explicitly, we first compute association matrices for each subset of the data collection, and then merge them to form a directed graph, after a threshold is applied. Since the diffusion vectors are in a high-dimensional space (the dimension being, for example, the number of different tokens in a corpus of texts), we propose a simple dimensionality reduction that allows for reasonable computational cost associated with the classification of these vectors.

Various related methods start to build a weighted undirected graph in the form of a similarity matrix, and make use of the spectrum of a derived Laplacian matrix for dimensionality reduction or clustering [1, 2, 3].

Diffusion kernels [4] propose a general method for constructing kernels on undirected graphs, so that such discrete structures can be used with classical kernel-based learning algorithms [5]. Unfortunately, this approach remains in general computationally expensive for large graphs.

Random walks over undirected graphs have also been used as a similarity measure for collaborative recommendation [6], community detection [7, 8], or relational classification [9]. The *ItemRank* algorithm [10], on the other hand, uses biased random walks over a directed graph to compute a score vector for each user, within the framework of a recommender system.

Although related to these previous works, our method combines a set of specific features. First, by operating preferably on directed graphs, it enables to get deeper insights into the topology of the dataset under consideration. Moreover, it can potentially use many types of biased random walks, and does not need the diffusion processes to reach full convergence (this may help to quickly compute estimates of the fingerprints, or to get "snapshots" of the dataset at different times). Finally, it can be used to compute numerical vectors (one may think of "feature vectors") for any subset of data items, including overlapping ones for example.

In the present paper, we start by presenting the basic formalism of our method in section 5.3, then show how it can be successfully applied to the problem of extracting relevant pathways from metabolic networks in section 5.4. In section 5.5, we provide a guideline to illustrate how to use our method for classification problems. Finally, we discuss a few important details more thoroughly in section 5.6.

5.3 Formalism

5.3.1 The association matrices

We consider a *data collection* $\Sigma = \{\sigma(1), \sigma(2), \ldots\}$ of "documents", where each *document* is viewed as a set of data items whose pairs get assigned a value of association. Some data collections such as the USF Free Associations dataset [11, 12] already provide explicit association weights while for others, one needs to define a way to compute these association values. The set of different items appearing in Σ is called $\mathcal{T}$. We denote $|\mathcal{T}|$ the cardinality of $\mathcal{T}$.

An example: computing words collocation

To illustrate our terminology and give an example of how to compute such association values when necessary, we consider a corpus Σ of text documents. A document $\sigma(k)$ will consist here of $N(k)$ tokens, which are typically stemmed words, with some stop words omitted. We define $I(k) = \{I_k(1), \ldots, I_k(N(k))\}$ as the list of tokens of $\sigma(k)$ in the order in which they appear. Each $I(k)$ is thus a map from positions in $\sigma(k)$ to tokens in $\mathcal{T}$.

We next define, for each k, the association matrix $K(k)$, which is a $|\mathcal{T}| \times |\mathcal{T}|$ matrix. We fix k and omit the index k for the moment. The matrix K measures the association of pairs of tokens $u, v \in \mathcal{T}$. For every $u \in \mathcal{T}$, we let $p_u(i)$ be the position of the i^{th} occurrence of token u. For every ordered pair (u, v) with $u \neq v$ of tokens we look for occurrences of the form $p_u(i) < p_v(j) < p_u(i+1)$, that is, occurrences of token v between two successive occurrences of token u (or after the last occurrence of u). We let s_{uv} be the set of all such pairs (i, j). Still omitting the index k, the matrix K is defined by

$$K_{uv} = g(h(u, v)) \sum_{(i,j) \in s_{uv}} f(p_u(i), p_v(j)) \ ,$$

where

$$h(u, v) = \frac{|s_{uv}|}{\sum\limits_{\substack{u', v' \in \mathcal{T}(k) \\ u' \neq v'}} |s_{u'v'}|} \ ,$$

$\mathcal{T}(k)$ are the tokens appearing in $I(k)$ and the functions $g(\cdot)$ and $f(\cdot, \cdot)$ are defined below. Note that $K_{uv} = 0$ if s_{uv} is the empty set.

We conduct our experiments with the following families of functions:

$$f(i, j) = \exp(-\frac{(j - i - 1)^\beta}{\sigma}) \text{ and } g(x) = -\log(x) \ .$$

The rationale behind the use of the function f is that the collocation measure should decrease exponentially with the distance between any two tokens. $g(h(\cdot))$ is a function of the relative frequency of each pair s_{uv} and serves as a normalizing function whose goal is to correct the influence of very frequent pairs. (These functions might be changed somewhat depending on the study one wants to perform.)

5.3.2 The domain graph

Having generated, for each document k, the association matrix $K(k)$ as described above, we next define a matrix $K(\Sigma)$ for the whole data collection Σ by

$$K(\Sigma)_{uv} = \sum_k K(k)_{uv} \ .$$

We now introduce a density parameter γ and define with it an *adjacency matrix* $A(\gamma)$ as follows: we replace the $N \equiv \gamma|\mathcal{T}| \cdot (|\mathcal{T}| - 1)$ largest elements of $K(\Sigma)$ by 1, and the others by 0. This means that the matrix elements of $K(\Sigma)$ above a certain threshold are replaced by 1 and the others by 0. [1] Note that our method does not specifically require the use of a binary adjacency matrix, and can easily be adapted to work with a weighted one, if it brings a clear benefit to do so.

The *domain graph* $G(\gamma)$ is the directed graph whose nodes are the elements of $\mathcal{T}$ and whose adjacency matrix is $A(\gamma)$. The topology of $G(\gamma)$ reflects the N strongest associations in Σ for a given density γ.

5.3.3 The diffusion fingerprints

Having determined the directed graph $G(\gamma)$, we now consider a diffusion process on it. In particular, we are interested in how a given document $\sigma(k)$ fits into this graph.

For a fixed k, there is a set $\mathcal{T}'(k) \subset \mathcal{T}$ of data items which are nodes of the domain graph and which appear in $\sigma(k)$. We want to know how the set $\mathcal{T}'(k)$ diffuses into the domain graph. [2]

We call *diffusion fingerprint* (DF) of document $\sigma(k)$ the distribution vector of the diffusion process started from the subset $\mathcal{T}'(k)$ of nodes in $G(\gamma)$. Note that the smaller the set $\mathcal{T}(k) \backslash \mathcal{T}'(k)$, the better the generated fingerprint represents $\sigma(k)$ within the context of the domain graph.

---

1. In case of multiplicities (for example if all matrix elements of $K(\Sigma)$ are equal), we perform a random choice of the required number of elements.

2. $\mathcal{T}'(k)$ might be smaller than the set $\mathcal{T}(k)$ of tokens in $\sigma(k)$.

Let P be the *probability matrix* defined by

$$P(\gamma) = D^{-1}(\gamma)A(\gamma) \ ,$$

where D is the diagonal matrix of the degrees of G. We compute the DF of document $\sigma(k)$ as the *personalized Pagerank* [13, 14] ppr_k defined recursively by

$$\mathrm{ppr}_k(t+1) = \alpha v_k + (1-\alpha)\,\mathrm{ppr}_k(t)P \ , \tag{5.1}$$

where $\alpha \in (0,1]$ is called the *jumping constant* and $\mathrm{ppr}_k(0) = v_k$. The vector v_k is the *personalized vector* given by

$$v_k(u) = \begin{cases} f_k(u) & \text{if } u \in \mathcal{T}(k) \\ 0 & \text{otherwise} \end{cases} \ ,$$

with $f_k(u)$ the frequency of data item u in document $\sigma(k)$. Note that the parameter $(1-\alpha)$ is the inverse of the expected path length of a random walker before being projected back to $\mathcal{T}'(k)$.
In principle, we define

$$\pi(k) = \lim_{t\to\infty} \mathrm{ppr}_k(t) \ ,$$

and call $\pi_t(k)$ the *DF at time t* of document $\sigma(k)$, and $\pi(k)$ its *stationary DF.*[3] Note that by default, we use the personalized Pagerank algorithm for computing the diffusion vectors as its properties are well-studied and understood, but nothing prevents our method to be used with other types of biased random walks. Considering the DFs at different times may also permit to construct derived feature vectors taking into account the dynamics of the diffusion processes.

5.4 Application to metabolic pathway inference

We will now apply the general method sketched out in section 5.3 to the problem of extracting metabolic pathways from metabolic networks. After describing what the question is, we proceed with the application of the general ideas, adapting them, where needed, to the specificities of the task.

3. Of course, we just compute $\mathrm{ppr}_k(t)$ for some sufficiently large t.

5.4.1 Description of the problem

The understanding of the dynamics underlying the set of metabolic reactions in a living cell has been pushed a step further by the extensive application of high-throughput analysis techniques in cell biology. In particular, the quantity and accuracy of data produced with these methods has dramatically increased in the last decade, and new computational techniques are required in order to interpret the set of these experimental results as a whole.

The metabolomic datasets can be represented by simple or bipartite graphs. In the first case, the nodes of the graphs are the metabolites, and a (directed or undirected) link is drawn between two molecules if there exists a chemical reaction having one of them as substrate and the other as product. In this case, we refer to the graph as the species-species graph (SSG). In the second case, the nodes of the graphs representing chemical interactions are both chemical molecules and reactions, interlinked by in- and out-flow relations. This kind of representation is referred to as species-reactions graph (SRG). An interesting problem in this context is the one of predicting metabolic pathways (chains of interlinked reactions, transforming a set S of source metabolites into a set T of targets) in metabolic databases.

Different approaches have been proposed for solving this problem: [15] have developed a method that ranks paths connecting a source and a target node in a SRG, according to gene expression levels corresponding to enzymes catalyzing reactions on the possible paths. Approaches based on the same intuition have been developed for pathway discovery in protein-protein interaction networks, by applying breadth-first search-based and Steiner tree problem solving algorithms [16, 17].

Other algorithms for metabolic pathway extraction based on shortest-path finding have been developed in [18, 19]. In particular, a method based on random walks on weighted metabolic graph is used in [19] to extract metabolic pathways connecting a given set of nodes in a SRG extracted from the *MetaCyc v11.0* database. This is done by searching for the shortest paths interlinking the set of terminal nodes of a given pathway, where by shortest path one means the path with minimal summed weights.

The weighting algorithm used in this case is based on the expected number of times a random walker transits through a given edge before reaching a terminal node of the pathway. The computational cost of this algorithm is $O(sm^3)$ [20], whereby m is the total number of edges of the domain graph and s the cardinality of $S \cup T$). An arbitrary limit on the path length must be given in order to reduce the complexity of the algorithm. In the following we propose an alternative to this metabolic pathway-finding algorithm, based on the DF method explained above.

5.4.2 Description of the algorithm

The graph G here is the directed domain graph modeling the metabolic network we want to analyze and P is the corresponding transition probability matrix. No filtering procedure has been applied on the domain graph in this case. We call *reverse graph* of G and denote G^* the graph having the same set of vertices as G, with all its edges reversed. We thus have $P^* = P^t$.

The domain graph G can be either simple or bipartite according to the representation of the metabolic network we choose to apply (SSG or SRG respectively).

Let $R \subset V$ be the set of species that participate in a given annotated metabolic pathway. The nodes in R are weakly connected in the domain graph G. We also denote $S \subset R$ the set of sources (*i.e.*, nodes with null in-degree), and $T \subset R$ the set of sinks (*i.e.*, nodes with null out-degree) of the pathway.

For a general pathway, we want to reconstruct the set R of nodes participating in it starting from the sets S of sources and T of sinks.

We start by computing $\pi(S)$, the stationary DF of S in G, and $\pi^*(T)$, the stationary DF of T in G^*. In order to exhibit the set of nodes that are highlighted by both of the fingerprints, we consider the *combined diffusion fingerprint*

$$\pi_{\triangleright}(S,T) = \pi(S) \times \pi^*(T) \ ,$$

where $\times$ represents the Hadamard Product (component-wise multiplication).

5.4.3 Pagerank boosting

Because of the presence of hub nodes (such as H_2O, ADP, $NADH$) in the corresponding graphs, a direct application of the algorithm described above would result in pathways connecting source and target nodes only through such highly connected compounds, since they effectively represent the shortest path to connect any given set of two nodes in the graph. This problem is well known and different methods have been developed to overcome it [21]. We propose here a method which is softer than the direct elimination of hub metabolites having a total degree above an arbitrarily fixed threshold.

In order for the algorithm to find relevant metabolites, *i.e.*, those that characterize a particular pathway and belong to as few other paths as possible, we renormalize our results keeping the centrality of the different nodes into account. In particular, we rescale the values of the combined diffusion fingerprint vector $\pi_{\triangleright}(S,T)$ resulting from the algorithm described above using the Pagerank vector $\pi(G)$ of the full graph G, and the Pagerank vector

$\pi(G^*)$ of the full reverse graph G^* in the following way:

$$\pi_{\triangleright}(S,T)^{boosted} = \frac{\pi_{\triangleright}(S,T)}{\pi(G) \times \pi(G^*)},$$

where both multiplication and division are intended in the component-wise sense.

This boosting procedure effectively penalizes metabolic "hubs" such as H_2O and ATP which have high in- and out-degrees.

5.4.4 Pathway selection

In order for the algorithm to detect a pathway connecting a set S of source nodes to a set T of sink nodes, we consider the n largest entries of the vector $\pi_{\triangleright}(S,T)^{boosted}$, and increase n until the subgraph resulting from the first n^w compounds connects all the elements of S to all the elements of T in the weak sense. The extracted subgraph (the inferred pathway) is then given by the subset of the n^w largest entries of $\pi_{\triangleright}(S,T)^{boosted}$ that belong to the weakly connected component connecting S to T.

5.4.5 Application of the algorithm

We applied the algorithm described above to the problem of finding the set of known pathways in the domain graph extracted from the annotated chemical reactions in the *MetaCyc v18.5* database. The domain graph is an SSG of 9 553 nodes and 75 078 edges. In order to have realistic results, we have applied the algorithm to the search of pathways of length $l \geq 3$, where l is the minimal shortest path length between any source and target node in the annotated pathway. In total, the algorithm has been applied to the reconstruction of 1 981 pathways.

We have tuned the only parameter of our algorithm to the value $\alpha = 0.15$, as generally assumed in the Pagerank literature [22]. However, our results are surprisingly invariant with respect to variations of this parameter. This behavior can be explained as follows. For a jumping constant $\alpha \approx 1$, the expected path length approaches the value 0 and no pathway can be inferred. On the opposite case, for $\alpha \approx 0$, one approaches the limit of a standard random walk, and all pathways, independently on their length, will be indistinguishably found. However, when we are in neither of these two cases, since the node weights in the $\pi_{\triangleright}(S,T)$ vector exponentially decrease as a function of distance from the source and target sets, shorter pathways will *always* be preferred by our algorithm over longer ones. After that, applying pagerank boosting, hub nodes are penalized and only the relevant shortest

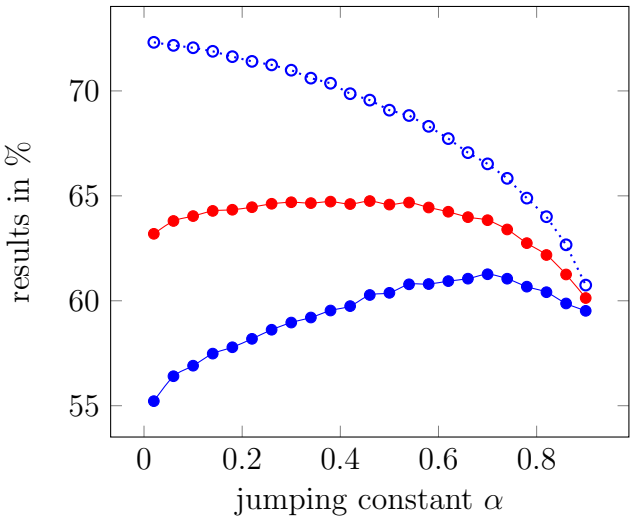

Figure 5.1 – Precision (filled blue circles) and recall (empty blue circles) of pathway inference averaged over 1981 experiments as a function of the jumping constant α. The geometric mean (red filled circles) of these two quantities, later also referred to as the *base score geometric accuracy*, is very weakly affected by the choice of α in the interval $\alpha \in (0.1, 0.6)$.

paths are evidenced. The behavior explained above is displayed in Fig. 5.1. There we see that the geometric mean of precision (PPV) and recall (TPR) of our results is very weakly influenced by the choice of the parameter α, the latter two measures being defined as

$$PPV = \frac{TP}{TP + FP}, \quad TPR = \frac{TP}{TP + FN}$$

where TP is the number of non-seed nodes that are present both in the inferred and in the annotated pathway. FP is the number of nodes that are present in the inferred but not in the annotated pathway, and FN is the number of nodes present in the annotated but not in the inferred pathway. This weak dependency makes the highly parallelizable algorithm presented above virtually parameter-free.

5.4.6 Results and discussion

To our knowledge, the strategy appearing in the bioinformatics literature that relates most closely to ours is the one presented in [19], where a random walk approach is used to infer metabolic pathways in metabolomics databases. There, the parameter chosen to quantify the quality of the results

is the *geometric accuracy*, defined as

$$acc_g = \sqrt{PPV \cdot TPR} \, .$$

In the following we will use the same parameter in order to simplify the comparison of the results, which is nonetheless difficult for several reasons.

The first is that even though the metabolomic database on which the algorithm has been applied is the same, *MetaCyc* has been updated multiple times since 2010, doubling in size. Therefore, the graph on which we have applied our algorithm is different from the one used in the article to which we are comparing.

The second reason is that the research of pathways in the article mentioned above is carried out by diffusion starting from source subgraphs S of increasing cardinality until the whole annotated pathway is covered. The result for the geometric accuracy is then given by the average of the geometric accuracies for each diffusion. In our case the set S is given by the terminal nodes of the pathway only.

Noting these differences, we have compared our results to those obtained in [19] selecting out of all the pathways analyzed therein those that are still present in the latest version of the *Metacyc* database. The comparison has been carried out by applying our DF-based algorithm to the inference of all pathways contained in the *Metacyc v18.5* database.

We can reconstruct the value of the geometric accuracy for the algorithm we are comparing to in the case only terminal nodes are taken as seed nodes. We define this parameter as the *base score geometric accuracy:* acc_g^b. In this case, for the pathways that are still present in the updated version of the database, we find a value of $acc_g^b = 0.61 \pm 0.05$. The uncertainty of $\pm 5\%$ is due to difficult decoding of the published data. The geometric accuracy of the inference of metabolic pathways using the DF-based algorithm described above is $acc_g^b = 0.66$.

5.5 A guideline to use DF for classification

This section is dedicated to further illustrating our method within the context of two classical textual classification problems, and provide a guideline to apply DF to classification tasks.

5.5.1 Gender detection

We first apply our method to a binary textual classification problem. Given a set of about 20 000 blogs [23], our goal is to determine the gender of

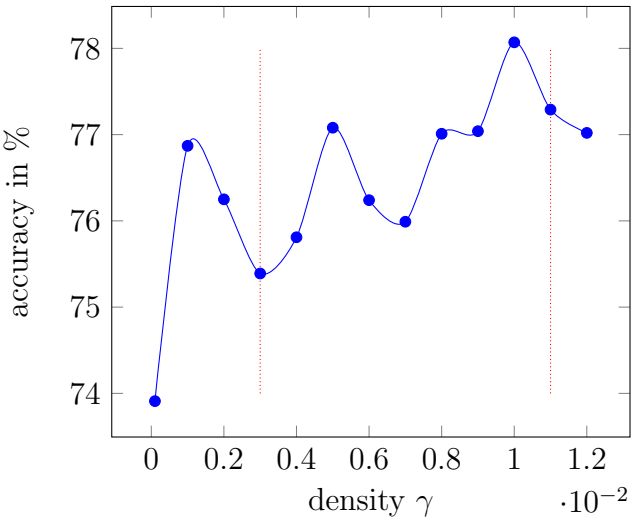

Figure 5.2 – Accuracy of gender detection as a function of the density parameter γ. The left red vertical line marks the minimum density above which the domain graph becomes a strongly connected component (*i.e.*, the diffusion process can reach all parts of the graph). The right red vertical line marks a density above which the accuracy starts decreasing and the computation of fingerprints becomes very costly.

the authors [24].

We randomly select a subset of 1 000 blogs with an equal proportion of male and female authors and use it as a training set. In both examples, we follow the procedure described in section 5.3 for generating the association matrices. Setting the density parameter $\gamma = \times 10^{-2}$ and the function f parameters $\beta = \sigma = 1$, we compute the domain graph G and the set of fingerprints for this subset. The graph G has 23 629 nodes and 5 583 061 edges. At this specific density, the graph forms a single strongly connected component and has a directed diameter equal to 6.

We then perform 10-fold cross validation on 10 random shufflings of the 1000 fingerprints. We get an average accuracy of 79.1% with the AdaBoost meta-algorithm using Decision Tree classifiers [25]. By comparison, we get an average accuracy of 74.8% when we use simple bag-of-words (BOW) vectors on the same set of blogs. Note that we use a BOW model not as a model to compete with, but rather as a null model to compare the effects of our dimensionality reduction heuristic (see details in section 5.6).

Moreover, when we apply our *OPC* dimensionality reduction heuristic to the fingerprint vectors, we observe that the accuracy remains almost constant until we reach a dimension equivalent to 10% of the size of the domain graph.

For instance, the accuracy still only reduces slightly to 77.85% when the dimension d is reduced from 23 629 to 3 000.

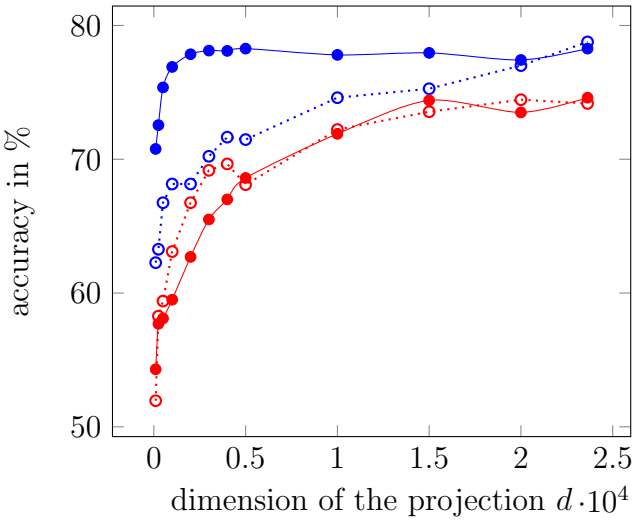

Figure 5.3 – Accuracy of gender detection as a function of the reduced dimension d for $\gamma = \times 10^{-2}$. The curves are: the diffusive fingerprint method with OPC (solid blue) and the diffusive fingerprint method with random projection (dotted blue), bag of words with OPC (solid red), and bag of words with random projection (dotted red). Note the stability of the fingerprint-OPC result when the dimension d is lowered.

5.5.2 Authorship attribution

By using the same set of blogs, we also apply our method to the problem of authorship attribution [26, 27]. We start by selecting at random 500 blogs containing at least 16 posts of more than 8 tokens each, and we split each of them in two equal number of posts.

For the 500 selected blogs (*i.e.*, 500 classes), we get 11 993 posts containing more than 8 tokens. We first use the aggregated list of tokens of the first halves for generating the domain graph. By choosing $\gamma = \times 10^{-2}$ and setting $\beta = \sigma = 1$, we get a graph with 17 036 nodes and 2 902 681 edges. We then generate the DFs for each post, and use the fingerprint vectors of the first halves for training 500 'one-vs-all' Random Forest binary classifiers [28].

We finally use the fingerprint vectors of the posts in the second parts for testing our classifier, which gives us an accuracy of 27.6%. By comparison, the accuracy is reduced to 24.2% when we use simple BOW vectors.

It is to be noted that as the number of classes increases, the computation needed to train the binary classifiers can become very costly. One way to alleviate this problem is to compute instead the mean of the training fingerprint vectors, and use it as a reference fingerprint for each author (*i.e.*, class). When the Manhattan distance is used for classifying the test vectors, we get an accuracy of 22.25% for the DFs and 9.36% for the corresponding BOW vectors.

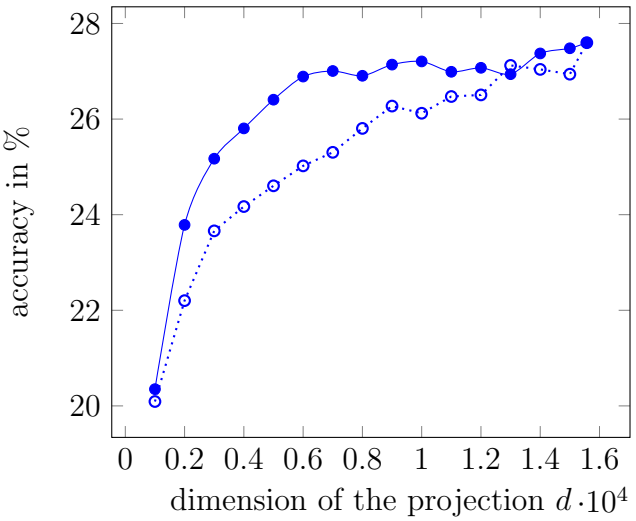

Figure 5.4 – Accuracy of authorship attribution as a function of the reduced dimension d. The curves are: the diffusive fingerprint method with OPC (solid blue) and the diffusive fingerprint method with random projection (dotted blue). Note that BOW results are not indicated here for sake a clarity.

5.5.3 Results and discussion

We emphasize that we use text classification problems as a mere illustration of our method. While this does not lead to state-of-the-art accuracies, it shows that the diffusion fingerprints method is generic enough to achieve quite decent results.

Gender detection

In [23], the authors use a dataset of 37 478 blogs with an equal number of male and female bloggers to perform gender classification. When applying

10-fold cross validation, they report an accuracy of 72% with content based features only, and 80.1% when stylistic features are also included.

By using many more features (*i.e.*, F-measure, stylistic features, gender preferential features, factor analysis, word classes and POS sequence patterns) and a custom *Ensemble Feature Selection* (EFS) algorithm, [29] report a maximum accuracy of 88.56% over a set of 3 100 blogs.

Thus, the accuracy of 79.1% we report seems acceptable, considering that we use no specific tools like n-grams and no additional features.

Authorship attribution

In [26], a dataset of 10 000 blogs is used for testing a method of authorship attribution over a large set of classes. For their first experiment, each blog is split in a reference text of 2 000 words and a snippet of 500 words, both represented as vectors of space-free character 4-gram frequencies. Using cosine similarity, 46% of 1 000 snippets selected randomly get assigned correctly. They also propose an improved algorithm consisting in repeatedly selecting a random fraction of the full feature set, in a similar way to classifier ensemble methods [30].

In our experiment, we obtain a lower accuracy of 27.6%. However, it is to be noted that in our case, the blog posts to be classified can contain as few as 8 tokens, which makes the classification task somewhat harder.

5.6 General discussion

By contrast to some non-linear dimensionality reduction techniques which aim at discovering the structure of a manifold from a set of feature vectors [31, 32, 33, 34], DFs start by extracting the latent topological properties of the provided data under the form of a directed graph. The distributions of the diffusion processes started from the data subsets are then computed as numerical vectors which can subsequently be fed to any classification algorithm.

5.6.1 Choosing the density parameter γ

We discuss here how to determine the density that provides the best accuracy.

We note that when $\gamma = 0$, there is no diffusion and our method amounts to classifying the personalized vectors associated to each subset in a high

dimensional space. In this case, our method simply corresponds to using a bag-of-words model.

On the other hand, when $\gamma = 1$ (then the graph is complete), the initial distributions (*i.e.*, personalized vectors) get flattened, whereas all the other coordinates of the high dimensional fingerprint vectors get assigned the same value. Thus, in this case, the fingerprint stationary distributions resemble an attenuated form of the initial distributions (*i.e.*, the personalized vectors), except that the variance of the data is much lower.

Diffusing in the domain graph from the initial distributions enables to grasp the generic characteristics of the data subsets at the scale of the whole domain graph, and improves average accuracy. A first requirement is thus that the density of the domain graph is large enough (*i.e.*, greater than a critical value γ_c) to enable a diffusion process starting from any subset of nodes to reach all the other parts of the domain graph. This means that γ needs to be chosen to enable the emergence of a giant strongly connected component (SCC). We see for example that the domain graph used previously for gender detection forms a single SCC for $\gamma \geq 3 \cdot \times 10^{-3}$.

We observe that the average entropy of the fingerprint distributions increases monotonically with the graph density, whereas, at the same time, the average variance decreases. Moreover, the average accuracy appears to be approximately a concave function of the domain graph density. In order to reach the best possible accuracy, we thus need to choose the density parameter $\gamma \in [\gamma_c, 1)$ so that the generated fingerprint distributions retain a high variance, but also exhibit a high average entropy: this corresponds to seeking a trade-off between the expressiveness and the genericity of the generated vectors.

5.6.2 Dimensionality reduction

After the construction of the diffusion vector, we use the following procedure for dimensionality reduction to lower the computational cost of classifying the generated DF.

Fix some d and let $\Phi(d) : \mathbb{R}^{|\mathcal{T}|} \to \mathbb{R}^d$ be the orthogonal projection of the $|\mathcal{T}|$-dimensional fingerprint vectors onto the d-dimensional node subspace $\mathbb{R}^d$ ($d \ll |\mathcal{T}|$) spanned by the d most central nodes of the domain graph $G \equiv G(\gamma)$. To find this projection, we apply the *Pagerank* centrality metric [22] to G in order to determine the set of the d most central nodes, corresponding a set $\mathcal{T}_d$ of tokens. The projection $\Phi(d)$ is then the $d \times |\mathcal{T}|$-matrix defined by

$$\Phi(d)_{uv} = \begin{cases} 1 & \text{if } v \in \mathcal{T}_d \\ 0 & \text{otherwise} \end{cases} , \quad u \in \mathcal{T}_d, v \in \mathcal{T} \ .$$

The intuition is that projecting the fingerprint vectors onto the d most central nodes amounts to embedding the data in a d-dimensional hyperplane of maximum variance, as we will see next.

We call *OPC* (Orthogonal Projection on Central nodes) the projection we just described.

5.6.3 OPC dimensionality reduction heuristic

Applying dimensionality reduction by projecting orthogonally on the hyperplane spanned by the set of most central nodes limits the decrease of classification accuracy and is computationally very efficient.

By using the Pagerank metric as a measure of centrality, this heuristic works well because the highest components of the Pagerank vector are highly correlated with the fingerprint coordinates of maximum variance. In the case of the domain graph applied above for gender detection, the Spearman correlation between the Pagerank distribution and the vector of fingerprint variance per coordinate is for example equal to 0.92 for $\gamma = \times 10^{-2}$. This means that projecting on the most central nodes amounts to projecting orthogonally on the hyperplane of maximum variance [35, 36].

The diffusion process plays a fundamental role in allowing a graceful decay of the classification accuracy when the dimension of the fingerprint vectors is reduced by applying the heuristic we just described. By extracting the generic characteristics of the data subsets at the scale of the whole data collection, diffusion fingerprints are in this case far more resilient to dimensionality reduction compared to a bag-of-words model.

5.6.4 Computational considerations

The first step of our method consists in generating the association matrices for the set of documents $\sigma(k)$. Here, the amount of computation highly depends on whether the association weights are provided in the data (*e.g.*, USF Free Associations dataset [11]) or not.

The generation of the domain graph is then straightforward, but we may face a problem if the number of data items is too large for the domain matrix to fit into memory. One potential way to alleviate the problem and to avoid explicitly computing the domain matrix is to generate for each association matrix a labeled unweighted directed graph of density γ in the form of a sparse binary adjacency matrix, and to generate the domain graph by taking the edge-union of the set of labeled subgraphs (*i.e.*, $G = \bigcup E(G_k)$). Note that in this case, it is more difficult to control the overall density γ of the

resulting domain graph, as the intersection of the edge sets $E(G_k)$ is not empty.

One may wonder at this point why we ignore the edge weights when computing the DFs. The reason is simply that experimentally, we observed in our examples that it leads to a decrease in accuracy for a higher computing cost. Thus, it seems sufficient to consider only the topological properties of the domain graph.

Computing the DFs amounts to computing Pagerank vectors and many different efficient algorithms [37] have been developed after the Pagerank algorithm was first described [22]. One of the fastest algorithm currently known uses a Monte Carlo-based incremental approach and has a complexity of $\mathcal{O}(\frac{n \ln m}{\epsilon^2})$, where n is the number of nodes, m the number of selected edges and ϵ the desired precision [38]. We also observe that it may not be necessary to reach full convergence, as we experimentally get a quasi-maximal accuracy when diffusing for a limited number of steps approximately equal to the directed diameter of the domain graph (*e.g.*, 6 in the case of the domain graphs we used for our text classification experiments).

Once the Pagerank vector of the domain graph has been computed, the heuristic we use for reducing the dimension of the fingerprint vectors is very fast, since it only consists in selecting a subset of the vector indices. We note that there is a very high correlation between the Pagerank vector of the domain graph and the component-wise average of the fingerprint vectors (*e.g.*, Spearman correlation of 0.92 for $\gamma = \times 10^{-2}$ in the graph used for gender detection). We can thus get a good approximate of the Pagerank vector by computing the average of the fingerprint vectors. Incidentally, we also find a nearly perfect correlation between the component-wise average and component-wise variance of the fingerprint vectors in the studied graphs. This suggests that it may not be necessary after all to compute the Pagerank vector of the domain graph, in order to quickly identify the most central nodes.

5.7 Conclusion

We have presented a method to generate fingerprint vectors for data exhibiting associative properties. It is based on diffusion processes over a domain graph and shows how to make dimensionality reduction efficient and robust. The numerical vectors that get generated can subsequently be used for classification or clustering.

We applied our method to two classical text classification problems with the same set of blogs. We showed that in both the case of gender detec-

tion and of authorship attribution, *Diffusion Fingerprints* provide a better accuracy and a greater resilience to dimension reduction than equivalent bag-of-words vectors.

The adaption and application of our method to the problem of metabolic subgraph extraction led to results that favorably compare with the state of the art in the field, establishing the power and flexibility of this elegant and conceptually simple method. The hybridization of this algorithm could in principle improve the quality of the results obtained above.

In the context of classification, we believe that DFs may prove useful not just for authorship detection but in many other domains, provided that a domain graph has been constructed. In the case of Free Association datasets for example [11, 12], DF could be used to detect cultural shifts or psychological disorders of certain individuals. When studying social networks (*e.g.*, friendship networks, co-author networks, online social networks, ...), DF could be used to compare the social profiles of different members of a group. DF could also find fruitful application to Word Sense Disambiguation by enabling to generate distinctive contextual fingerprints for the words to be disambiguated.

Bibliography

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[2] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering.," in *NIPS*, vol. 14, pp. 585–591, 2001.

[3] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 21, pp. 7426–7431, 2005.

[4] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *ICML*, vol. 2, pp. 315–322, 2002.

[5] B. Schölkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[6] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *Knowledge and data engineering, ieee transactions on*, vol. 19, no. 3, pp. 355–369, 2007.

[7] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pp. 475–486, IEEE, 2006.

[8] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[9] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, ACM, 2014.

[10] M. Gori, A. Pucci, V. Roma, *et al.*, "Itemrank: A random-walk based scoring algorithm for recommender engines.," in *IJCAI*, vol. 7, pp. 2766–2771, 2007.

[11] D. Nelson, C. McEvoy, and T. Schreiber, "The University of South Florida free association, rhyme, and word fragment norms," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 3, pp. 402–407, 2004.

[12] J. Dubuisson, J.-P. Eckmann, C. Scheible, and H. Schütze, "The topology of semantic knowledge," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (Seattle, Washington, USA), pp. 669–680, Association for Computational Linguistics, October 2013.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," 1999.

[14] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, (Washington, DC, USA), pp. 475–486, IEEE Computer Society, 2006.

[15] A. Zien, R. Küffner, R. Zimmer, and T. Lengauer, "Analysis of gene expression data with pathway scores.," in *Ismb*, vol. 8, pp. 407–417, 2000.

[16] M. S. Scott, T. Perkins, S. Bunnell, F. Pepin, D. Y. Thomas, and M. Hallett, "Identifying regulatory subnetworks for a set of genes," *Molecular & Cellular Proteomics*, vol. 4, no. 5, pp. 683–692, 2005.

[17] D. Rajagopalan and P. Agarwal, "Inferring pathways from gene lists using a literature-derived network of biological relationships," *Bioinformatics*, vol. 21, no. 6, pp. 788–793, 2005.

[18] A. V. Antonov, S. Dietmann, P. Wong, and H. W. Mewes, "Ticl – a web tool for network-based interpretation of compound lists inferred by high-throughput metabolomics," *FEBS Journal*, vol. 276, no. 7, pp. 2084–2094, 2009.

[19] K. Faust, P. Dupont, J. Callut, and J. van Helden, "Pathway discovery in metabolic networks by subgraph extraction," *Bioinformatics*, vol. 26, no. 9, pp. 1211–1218, 2010.

[20] P. Dupont, J. Callut, G. Dooms, J.-N. Monette, Y. Deville, *et al.*, "Relevant subgraph extraction from random walks in a graph," *Université catholique de Louvain, UCL/INGI, Number RR*, vol. 7, 2006.

[21] K. Faust and J. van Helden, "Predicting metabolic pathways by subnetwork extraction," in *Bacterial Molecular Networks* (J. van Helden, A. Toussaint, and D. Thieffry, eds.), vol. 804 of *Methods in Molecular Biology*, pp. 107–130, Springer New York, 2012.

[22] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[23] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging.," in *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, vol. 6, pp. 199–205, 2006.

[24] J. Dubuisson, "Diffusion Fingerprints – Application demo to text classification," 2014.

[25] Y. Freund and R. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, EuroCOLT '95, (London, UK, UK), pp. 23–37, Springer-Verlag, 1995.

[26] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Language Resources and Evaluation*, vol. 45, no. 1, pp. 83–94, 2011.

[27] Y. Seroussi, F. Bohnert, and I. Zukerman, "Authorship attribution with author-aware topic models," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, (Stroudsburg, PA, USA), pp. 264–269, Association for Computational Linguistics, 2012.

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] A. Mukherjee and B. Liu, "Improving gender classification of blog authors," in *Proceedings of the 2010 conference on Empirical Methods in natural Language Processing*, pp. 207–217, Association for Computational Linguistics, 2010.

[30] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern recognition*, vol. 36, no. 6, pp. 1291–1302, 2003.

[31] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[32] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2003.

[34] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators," in *in Advances in Neural Information Processing Systems 18*, pp. 955–962, MIT Press, 2005.

[35] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[36] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[37] P. Berkhin, "A survey on pagerank computing," *Internet Mathematics*, vol. 2, pp. 73–120, 2005.

[38] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 4, no. 3, pp. 173–184, 2010.

Chapter 6

Sampling Massive Directed Complex Networks

6.1 Abstract

Since the birth of the *Web* and the subsequent development of *Online Social Networks* (OSNs), the analysis and characterization of the so-called *complex networks* has drawn considerable attention. When the full graph under consideration remains partially unknown because it cannot be crawled entirely, one needs to get a sample in order to make certain measurements. After reviewing the literature and various existing sampling strategies, we present a heuristic that, under stronger constraints than other methods, aims at sampling iteratively connected subgraphs from unexplored well-connected directed complex networks, and apply our findings to the Twitter social network.

6.2 Introduction

Twitter is an online social network and microblogging service created in March 2006. Its users can publish short messages called "tweets" limited to 140 characters, and follow other users to receive their "tweets".

The entire site was crawled in 2009 and its social graph was collected by [1]. It is a follower-followee directed graph containing 41.7 million vertices and 1.47 billion edges.

Twitter has since sustained a rapid growth, and the number of its active users has reached 271 million in July 2014. Moreover, some rate limits have been applied to its API, which now makes it impossible to crawl the entire social graph in a reasonable amount of time.

Thus, Twitter is a good example where sampling remains the only available option that one can use to discover and measure certain characteristics of the underlying social graph. Another obvious example is the Web graph whose size grows exponentially, and that has become so big as to prevent any serious attempt to get a complete snapshot of it.

In this article, after analyzing some basic properties of the Twitter social graph, we describe and analyze a sampling method which aims at getting under strong constraints a "representative" subgraph of a yet unexplored graph, by exploring it in an iterative way. By representative subgraph, we mean a subgraph induced by the set of sampled vertices whose topology reproduces as faithfully as possible the set of basic properties we describe in the following section. The distributions of in- and out-degrees, the degree correlations, and the organization of 2- and 3-cycles reflect indeed fundamental topological properties of the underlying network [2].

6.3 Formalism

Let $G(V, E)$ be a directed graph with vertex set V and edge set E. And let n and e be the cardinalities of the sets V and E. For any given vertex $u \in V$, we denote $\Gamma(u)$ the set of its children and $\Phi(u)$ the set of its parents. Let c_u, p_u and i_u be the cardinalities of the sets $\Gamma(u)$, $\Phi(u)$ and $\Gamma(u) \cap \Phi(u)$ respectively.

We call *colink* a directed 2-cycle and, *triangle* a directed 3-cycle. For a given node u, the number of potential colinks is given by the total number of neighbors $\eta_u = c_u + p_u - i_u$, and the number of potential triangles is given after a simple calculation by $\tau_u = c_u * p_u - i_u$.

We define the *local colink coefficient* C_{1u} of a vertex u to be the number of colinks passing by u over the potential number of colinks, *i.e.*, if u is not isolated:

$$C_{1u} = \frac{i_u}{\eta_u} \tag{6.1}$$

Let a *triangle* be a triplet $(u, v, w) \in V^3$ such that $v \in \Gamma(u),\ w \in \Phi(u)$, and $(v, w) \in E$. We now define the *local clustering coefficient* C_{2u} of a vertex u to be the number of triangles centered at u over the potential number of triangles, i.e., if $\tau_u > 0$:

$$C_{2u} = \frac{t_u}{\tau_u} \tag{6.2}$$

where t_u denotes the number of triangles centered at node u.

It is to be noted here that the usual definition of the clustering coefficient uses $\eta_u(\eta_u - 1)$ instead of τ_u as the number of potential triangles. Although it comes as a natural extension of the formula used for undirected graphs, we believe it gives an unnecessarily high upper bound on the number of potential triangles. By using the usual denominator, the clustering coefficient can indeed reach its maximum value of 1 only in the specific case where $\Gamma(u) = \Phi(u)$, and the set of neighbors of u form a complete graph.

The *colink coefficient* and *clustering coefficient* of the graph G are the average of the corresponding local coefficients:

$$C_{1G} = \frac{1}{n}\sum_{v \in V} C_{1v}, \quad C_{2G} = \frac{1}{n}\sum_{v \in V} C_{2v} \ .$$

We suppose that there exist a total order relation $\mathfrak{O}_1$ on the set V of vertices. In the case vertex ids are represented by integers, it is for example natural to use them as a mean to compare any two vertices. In the following, $u > v$ means that u is a greater vertex than v according to the total order relation $\mathfrak{O}_1$. We also define a second total order relation $\mathfrak{O}_2$ on the set V as:

$$u \succ v \Leftrightarrow (\tau_u > \tau_v) \vee (\tau_u = \tau_v \wedge u > v)$$

We call *reciprocity* of G the proportion of existing relations that are reciprocated, which corresponds to the number of unique existing colinks over the number of unique linked pairs of vertices. Note that two other interesting options may consist in measuring the proportion of edges belonging to a colink, or the proportion of any pairs of vertices that are attached by a colink. Let $I_G = \{(u, v) \in E : u < v,\ (v, u) \in E\}$ be the set of unique colinks in G. We define the reciprocity coefficient by:

$$\mathfrak{R}_G = \frac{i_G}{\eta_G} \tag{6.3}$$

where $i_G = |I_G|$, and $\eta_G = e - i_G$.

Let a *triangle candidate* be a triplet $(u, v, w) \in V^3$ such that $v \in \Gamma^+(u)$, $w \in \Phi^+(u)$, where $\Gamma^+(u)$ is the set of greater children of u (*i.e.*, $\{v \in \Gamma(u) : u \prec v\}$), and $\Phi^+(u)$ is the set of greater parents of u (*i.e.*, $\{w \in \Phi(u) : u \prec w\}$). A *unique triangle* is thus a closed triangle candidate, that is, a triangle candidate (u, v, w) such that $(v, w) \in E$.

We call *transitivity* of G the number of unique triangles over the number of triangle candidates. Note that as for the colink coefficient, other ways may be used for computing the transitivity, like measuring the proportion of any triples of vertices that are attached by a triangle. By using our definition, the transitivity coefficient is given by:

$$\mathfrak{T}_G = \frac{t_G}{\tau_G} \tag{6.4}$$

where t_G and τ_G denote the cardinalities of the set of unique triangles, and the set of triangle candidates in G respectively.

We finally define the *vertex degree correlation* which measures the Pearson correlation between the vertex in- and out-degrees, and the *(o,i)-assortativity* which measures the tendency of vertices with a certain out-degree to connect to vertices of a certain in-degree.

We denote k_u^{in} and k_u^{out} the in- and out-degree of vertex u. $s(e)$ and $t(e)$ represent the source and target vertices of edge e. The vertex degree correlation is given by:

$$\rho_v(G) = \frac{\sum_u (k_u^{in} - \langle k \rangle)(k_u^{out} - \langle k \rangle)}{\sqrt{\sum_u (k_u^{in} - \langle k \rangle)^2} \sqrt{\sum_u (k_u^{out} - \langle k \rangle)^2}} \tag{6.5}$$

And the (o,i)-assortativity is defined as follows:

$$\rho_e^{(o,i)}(G) = \frac{\sum_e (k_{s(e)}^{out} - \langle k \rangle)(k_{t(e)}^{in} - \langle k \rangle)}{\sqrt{\sum_e (k_{s(e)}^{out} - \langle k \rangle)^2} \sqrt{\sum_e (k_{t(e)}^{in} - \langle k \rangle)^2}} \tag{6.6}$$

Note that the two last equations define Pearson correlation coefficients which are known not to be resistant to outliers in the data. To solve the problem, one can instead compute the more robust Spearman correlation coefficients [3], by simply replacing the variables by their associated rank in the corresponding equations. Thus, considering the highly heterogeneous structure of a complex network like the Twitter follower-followee graph, we use the Spearman version of the correlation coefficients to conduct our analysis in the sequel.

6.4 The Twitter graph of followers

6.4.1 Basic analysis

We start with a graph of followers having 41 652 230 vertices and 1 468 365 182 edges.

The first step of our analysis is to extract its *core* (*i.e.*, its main strongly connected component) [2] by using the Pearce improved version [4] of the classic Tarjan algorithm [5]. It is to be noted that with such a big graph, the typical recursive approach guiding the depth-first exploration is not applicable. We thus use a custom iterative version of the algorithm written in Julia [6].

The core has 33 479 734 vertices, 1 394 440 906 edges, and an average degree of 41.65. The maximum in-degree and out-degree are respectively equal to 2 936 232 and 768 552.

6.4.2 The degree distributions

Unsurprisingly, the in- and out-degree distributions of the core follow power-laws. By applying logarithmic binning as described in [7], we find an exponent of 2.15 for the in-degree distribution, and 2.14 for the out-degree distribution.

Moreover, as noted in [1], clear deviations of the out-degree distribution appear for degrees equal to 20 and 2000. These deviations correspond to external influences exerted by the Twitter system on the natural evolution of the social graph. In the first case, this is due to the fact a set a 20 people gets automatically recommended upon registration, and newcomers can add them easily in a single click. The second deviation appeared because a limit of 2000 on the number of people one could follow was set before 2009.

Concerning the in-degree distribution, we observe that users with more than 10^5 followers have many more followers than the power-law actually predicts. This is a specific feature of the Twitter social network, and basically corresponds to celebrities who get followed by a large number of fans, and who use Twitter as a mean to inform them.

6.4.3 Vertex degree correlation and assortativity

The vertex degree correlation is of 0.69 for the whole core, and reach 0.93 for the set of nodes whose out-degree is greater than 2000. This confirms what we observe on the figure 6.2, where high degree nodes appear to have strongly correlated in- and out-degrees.

The (o,i)-assortativity takes a very high value of 0.97. This indicates that the higher the out-degree of a node, the higher the in-degree of the nodes it connects to. More concretely, this means that users having a similar number of followees tend to follow with a high probability users having a similar number of followers.

6.4.4 Reciprocity and colink distribution

The core contains 265 678 093 unique colinks and has thus 15.87 colinks per node in average. Its reciprocity and colink coefficients are respectively equal to 0.24 and 0.20. The node with the highest number of colinks has 698 112 colinks, with an in-degree of 1 864 061 and an out-degree of 768 552.

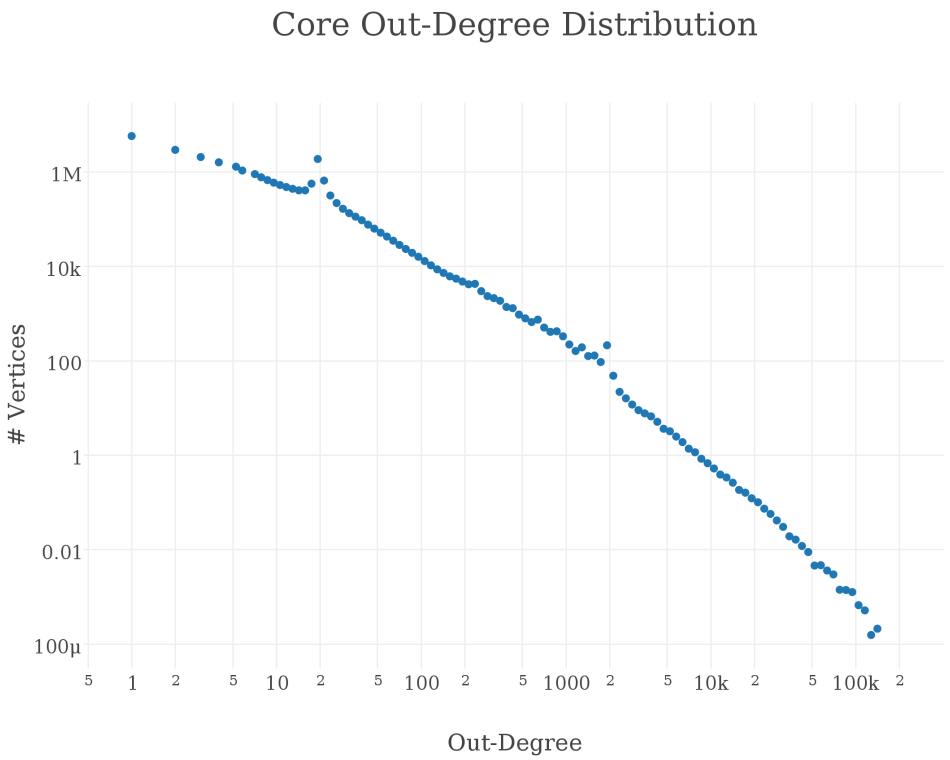

Figure 6.1 – Power law out-degree distribution of the core. We see two clear deviations on the curve for degrees equal to 20 and 2000 (see comments above). Note that for sake of clarity, we do not represent the in-degree distribution.

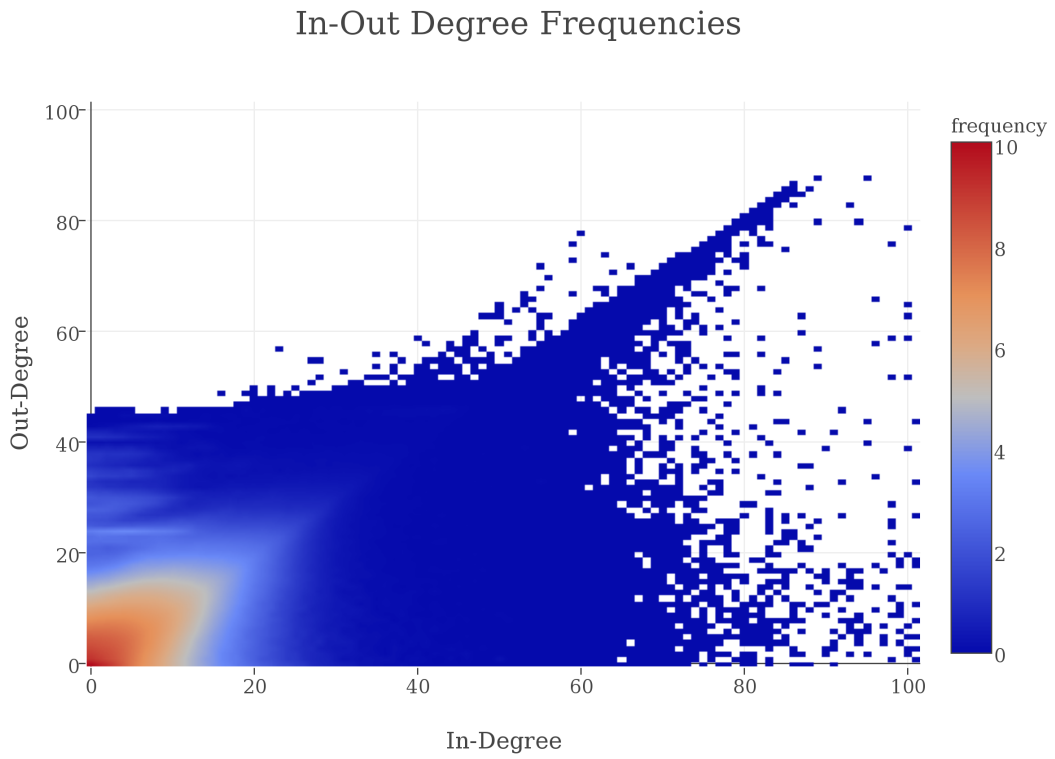

Figure 6.2 – Heatmap of the in-out degree frequencies. We see that above a certain value of the out-degree, the in- and out-degrees seem to be well correlated, which confirms the very high value found for the vertex degree correlation for degrees above 2000. We also observe that some high in-degree nodes have an unusually low out-degree, which explains the deviation appearing for high in-degree nodes in the plot of the in-degree distribution (see comments above).

Concerning the colink distribution, we see it follows a power-law with exponent 2.12.

As noted in [1], the reciprocity and colink coefficients are rather low compared to other social networks. This is due to the fact Twitter has a hybrid status between a media and a social networking site: people may follow other users for receiving news, but only a fraction of them get followed back by their followees.

6.4.5 Transitivity and triangle distribution

The core contains 44 736 505 422 unique triangles, and we observe that the triangle distribution follow a power-law with exponent 1.45. The average number of triangles per node is equal to 4008.68, and the maximum number of triangles per node reaches a value of 104 231 123, for a node having an in-degree of 103 840 and an out-degree of 106 255.

The transitivity and clustering coefficient of the core are equal to 0.28 and 0.04 respectively, which is, as expected, very high compared to an equivalent random graph.

6.4.6 Pagerank correlations with local measures

As our heuristic uses local estimates of the Pagerank values for guiding the exploration process of the graph [8], we look at the Spearman correlations between the Pagerank vector and different local measures. The results for the Twitter graph are summarized in the table 6.1.

It is already known that the Pagerank value of a given node is well correlated to its in-degree [9], but we see here that the number of colinks and the colink coefficient measures exhibit even stronger correlations with the Pagerank vector. These local measures thus appear to be the best candidates for undirectly comparing the Pagerank values of any two nodes, when these values are unknown.

6.5 If the whole graph is known

We start by describing a few algorithms that permit to estimate some basic properties of a massive graph, in the case the whole structure of the graph is known. These are precisely the ones we used to compute the measures described above for the Twitter social network.

Local measure	Correlation with Pagerank
in-degree	0.60
out-degree	0.37
# colinks	**0.82**
# triangles	0.72
colink coeff.	0.68
clustering coeff.	0.61

Table 6.1 – Correlation of the Pagerank vector with local measures of the Twitter social network

6.5.1 Using a deterministic approach

Adopting a deterministic approach may be desirable to get the true values of the properties being studied, but it is often computationally unfeasible when applied to massive graphs. When a graph gets bigger, computing the average degree remains relatively easy, but computing the betweeness centrality [10] quickly becomes intractable for example.

In the following, we describe two algorithms for listing all the colinks, and all the triangles of a massive graph. These algorithms are specifically designed for *MapReduce*, a simple programming model first proposed by [11], and used for processing very large datasets in parallel on computer clusters built from commodity hardware. The technology was first developed at Google for running data analytics jobs, and was later open-sourced.

Listing all colinks

First, we propose a straightforward algorithm for listing all the colinks of a directed graph. In order to avoid considering twice the same colink, we use the total order realtion $\mathfrak{O}_1$ on the set V of vertices, and consider each colink from its least vertex only.

Algorithm 1 MR-ListAllDirectedColinks

```
function MAP(< u; v >)
    # input: set of directed edges (u, v)
    if u < v then
        # v greater child of u
        emit(< u; (v, 1) >)
    else
        # u greater parent of v
        emit(< v; (u, −1) >)

# Γ+(u): set of greater children of u
# Φ+(u): set of greater parents of u
function REDUCE(< u; {(v, 1)|v ∈ Γ+(u)} ∪ {(w, −1)|w ∈ Φ+(u)} >)
    for all (v, w) do
        if v == w then
            # a colink (u, v) has been found
            emit(< u; v >)
```

Listing all triangles

We now present an algorithm for listing all the triangles of a directed graph which is inspired by *NodeIterator++* [12, 13], an algorithm for efficiently counting all the triangles in an undirected graph with a time complexity of $O(|E|^{\frac{3}{2}})$.

As previously, in order to avoid listing three times each triangle, we use the total order relation $\mathfrak{O}_2$ on the set V of vertices, and consider each triangle candidate from its least vertex only.

Algorithm 2 MR-ListAllDirectedTriangles

```
function MAP(< u; v >)
    # input: set of directed edges (u, v)
    if u ≺ v then
        # v greater child of u
        emit(< u; (v, 1) >)
    else
        # u greater parent of v
        emit(< v; (u, −1) >)

# Γ+(u): set of greater children of u
# Φ+(u): set of greater parents of u
function REDUCE(< u; {(v, 1)|v ∈ Γ+(u)} ∪ {(w, −1)|w ∈ Φ+(u)} >)
    # N.B.: loading the graph in memory makes the loop efficient
    for all (v, w) do
        if (v, w) ∈ E then
            # a triangle (u, v, w) has been found
            emit(< u; (v, w) >)
```

It is important to note that to make the reduce phase efficient, the graph needs to be loaded entirely in memory, which might not be possible. In such case, it is possible to revert to a pure streaming version of the algorithm, by adding a second MapReduce round which checks if the pair (v, w) belongs to E:

Algorithm 3 MR-ListAllDirectedTriangles (streaming version - round 2)

function MAP($< u; \{v | v \in \Gamma(u)\} \cup \{(v, w) | v \in \Gamma^+(u), w \in \Phi^+(u)\} >$)
  # 2 input sources: set of edges (u, v) and output of the 1st reducer
  if input of type (u, v) **then**
    # 1st input source: set of directed edges
    emit($< (u, v); 0 >$)
  else
    # 2nd input source: output of the 1st reducer
    emit($< (v, w); u >$)

function REDUCE($< (v, w); \{u | u \in S \subseteq V \cup \{0\}\} >$)
  # check if $(v, w) \in E$
  if $0 \in S$ **then**
    for $u \in V \cap S$ **do**
      # a triangle (u, v, w) has been found
      emit($< u; (v, w) >$)

6.5.2 Estimating with Monte-Carlo methods

When the whole graph is known, it is possible to sample the set of vertices uniformly, and apply Monte Carlo methods to compute estimates of the sought-after graph properties. In the following, we illustrate this approach by slightly adapting an algorithm described in [12] and [14] for computing the clustering coefficient of a directed graph.

Clustering coefficient

The algorithm takes as input a set of s vertices sampled uniformly at random from V, and works as follows:

Algorithm 4 MC-ApproxGraphClusteringCoefficient

Input: s vertices $u_1, \cdots, u_s$ sampled from V uniformly at random
for $i = 1$ to s **do**
  sample a pair $v \in \Gamma(u_i),\ w \in \Phi(u_i),\ v \neq w$ uniformly at random
  if $(v, w) \in E$ **then**
    $X_i \leftarrow 1$
  else
    $X_i \leftarrow 0$
Output: $X = \frac{1}{s} \sum_{i=1}^{s} X_i$

For each $i \in \{1, 2, \cdots, s\}$, we have

$$E[X_i] = \frac{1}{n}\sum_{v \in V} E[X_i | u_i = v] = \frac{1}{n}\sum_{v \in V} C_{2v} = C_{2G}$$

By using the fact that the X_i are 0-1 mutually independent random variables, we can write

$$Var[X_i] \leq E[X_i^2] = E[X_i] = C_{2G}$$

and,

$$Var[X] = Var[\frac{1}{s}\sum_{i=1}^{s} X_i] = \frac{1}{s^2}\sum_{i=1}^{s} Var[X_i] \leq \frac{C_{2G}}{s}$$

Finally, by using the Chebyshev inequality, we get

$$P[|X - E[X]| \geq \epsilon E[X]] \leq \frac{Var[X]}{(\epsilon E[X])^2} \leq \frac{1}{s\ \epsilon^2\ C_{2G}}$$

Thus, the algorithm approximates the clustering coefficient within relative error $(1 \pm \epsilon)$ with probability 2/3, if $s \geq \frac{3}{\epsilon^2 C_{2G}}$ for example. Moreover, by using a standard amplification technique which consists in running the algorithm $O(log\frac{1}{\delta})$ times and returning the median of all results, we can get a $(1 \pm \epsilon)$-approximation of C_{2G} with probability $1 - \delta$.

Note that the accuracy of this algorithm directly depends on the value of the clustering coefficient it tries to estimate. In order to overcome this situation, a modified version of the algorithm is proposed in [14] to provide a guarantee on the accuracy.

Local clustering coefficient

We can proceed in the same manner to estimate the clustering coefficient of a single node:

Algorithm 5 MC-ApproxVertexClusteringCoefficient

Input: set S of s pairs (v, w) of neighbors of vertex u, such that $v \in \Gamma(u),\ w \in \Phi(u),\ v \neq w$ sampled uniformly at random
for $(v, w) \in S$ **do**
  if $(v, w) \in E$ **then**
    $X_i \leftarrow 1$
  else
    $X_i \leftarrow 0$
Output: $X = \frac{1}{s}\sum_{i=1}^{s} X_i$

Like for the previous algorithm, we can show that:

$$P[|X - E[X]| \geq \epsilon E[X]] \leq \frac{1}{s\ \epsilon^2\ C_{2u}}$$

As a consequence, we know that we need $O(log\frac{1}{\delta} \cdot \frac{1}{\epsilon^2\ C_{2u}})$ trials to get a $(1 \pm \epsilon)$-approximation of C_{2u} with probability $1 - \delta$.

On the other hand, a deterministic approach to compute the clustering coefficient of a node u requires to check all τ_u triangle candidates. Thus, depending on the required level of accuracy, it may be more efficient to use a probabilistic approach or to compute the exact value of the clustering coefficient deterministically.

Let us suppose for example it is sufficient to get a (1 ± 10^{-3})-approximation of the local clustering coefficient with a probability of 95%. The randomized algorithm then requires about $3 \cdot 10^8$ trials for a clustering coefficient equal to 10^{-2}. In the Twitter social network we study, the node with the highest number of triangles has 104 231 123 closed triangles for 11 033 424 651 triangle candidates, and its clustering coefficient is equal to $9.4 \cdot 10^{-3}$. In such a context, we see it is much more efficient to estimate the clustering coefficient with a probabilistic approach (*i.e.*, $3 \cdot 10^8$ trials versus $11 \cdot 10^{10}$ triangle candidates to be checked).

Note that in order to estimate the number of necessary trials and to be able to use such a hybrid approach (*i.e.*, decide for each node whether to use a probabilistic or deterministic algorithm), we need to know the value of the clustering coefficient beforehand. As this is obviously not possible, this means we need at least to provide a lower bound to its value. One possible solution may be to consider that each triangle candidate of node u is closed with a probability equal to the graph density $p = \frac{e}{n(n-1)}$, and set the bound to $\tau_u \cdot p$. However, this would give a bound which is far too low, as the clustering coefficient of a complex network is usually orders of magnitude greater than its density. A more pragmatic approach would be to define an adaptive heuristic to estimate the unknown value of the clustering coefficient from the set of values computed so far.

6.6 If the graph is unexplored

In the case the graph remains largely unexplored, it is not possible to sample a subset of nodes uniformly at random. A simple solution consists in performing a *Breadth First Search* (BFS) from a vertex selected randomly [15], but the generated sample is known to be biased toward high-degree nodes [16]. Another common practice with online social networks (OSNs)

(*e.g.*, Myspace or Facebook) is to perform random vertex sampling by querying randomly generated user-ids (UNI) [17]. However, this workaround is in most cases impractical, as the process is costly when the user-id space is sparsely populated [18], and far too slow if the queries are rate-limited [19].

These strong constraints also rule out recent algorithms like *Frontier Sampling* (FS) [20] which starts multiple parallel random walks from a seed of vertices selected uniformly at random, or *Albatross* (AS) [21] which uses random jumps to avoid getting stuck in densely connected parts of the graph.

6.6.1 Sampling with a random walk

The case of sampling techniques based on a random walk deserves a special treatment. In the following, we first detail, in the context of an undirected graph, the flaws of an exploration based on a simple random walk, and a way to overcome the problem . We then explore the question in the context of a directed graph.

The case of an undirected graph

The stationary distribution of a random walk on an undirected graph is known to be [22]:

$$\pi_v = \frac{k_v}{2e}, \forall v \in V$$

The expected observed degree distribution p_k^* of a random walk is hence given by:

$$p_k^* = \sum_{v \in V} \pi_v 1_{\{k_v = k\}} = \frac{k}{2e} \sum_{v \in V} 1_{\{k_v = k\} = \frac{k}{2e} p_k |V| = \frac{k p_k}{\langle k \rangle}}$$

with p_k and $\langle k \rangle$ respectively denoting the true degree distribution and average degree. And we deduce that the observed average degree $\langle k^* \rangle$ of a random walk is

$$\langle k^* \rangle = \sum_k \frac{k^2 p_k}{\langle k \rangle} = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

Like for a BFS exploration, we see that a random walk is biased toward high-degree nodes. One way to get a uniform sample of the vertices is to perform a so-called *Metropolis-Hastings Random Walk* (MHRW) [23] which modifies the transition probabilities of a usual random walk in the following way:

$$P_{uv} = \begin{cases} \frac{1}{k_u}\ min(1, \frac{k_u}{k_v}) & \text{if } (u,v) \in E \\ 1 - \sum_{w \neq u} P_{uw} & \text{if } w = u \\ 0 & otherwise \end{cases}$$

Another solution called Re-Weighted Random Walk (RWRW) [17] consists in resampling a set of nodes discovered via a usual random walk in order to correct its bias.

The case of a directed graph

In the case of a directed graph, it is also known that a random walk is biased toward high in-degree nodes. However, the transition matrix of the corresponding Markov Chain is asymmetric, and there is no analytical form to express the value of the stationary distribution. For this reason, it is not possible to quantify beforehand the bias of a random walk.

Although the true average degree of the Twitter core is equal to 41.65, we find experimentally that the values of the observed average in-degree and average out-degree of a random walk converge toward about 81 000 and 5 150 respectively (see figure 6.3). We note that these values are surprisingly close to $\frac{\langle k_{in}^2 - k_{in}k_{out}\rangle}{2\ \langle k\rangle} = 80070.63$ and $\frac{\langle k_{out}^2 + k_{in}k_{out}\rangle}{2\ \langle k\rangle} = 5163.2$.

To further check those results, we also compute the stationary distribution π of a random walk on the Twitter core (*i.e.*, the principal eigenvector of the transition probability matrix) with the power iteration method [24]. We then obtain the true values of the observed average in- and out-degrees by calculating the dot products of the vector π with the in- and out-degree vectors of the core, and find $\langle k_{in}^* \rangle = 85\,351.11$ and $\langle k_{out}^* \rangle = 5\,336.79$.

So, our problem first consists in trying to correct the bias of a random walk on a directed graph, while ignoring the true value of the stationary distribution π, and the whole structure of the graph. We saw earlier that, in the case of the Twitter core, the values of the vector π are strongly correlated with the number of colinks at a given node. Furthermore, we know we can get a nearly uniform sample of vertices by selecting any node u with a probability which is inversely proportional to π_u [8].

We thus implement a modified random walk inspired by the Metropolis-Hastings random walk seen above (MHRW), and using the number of colinks

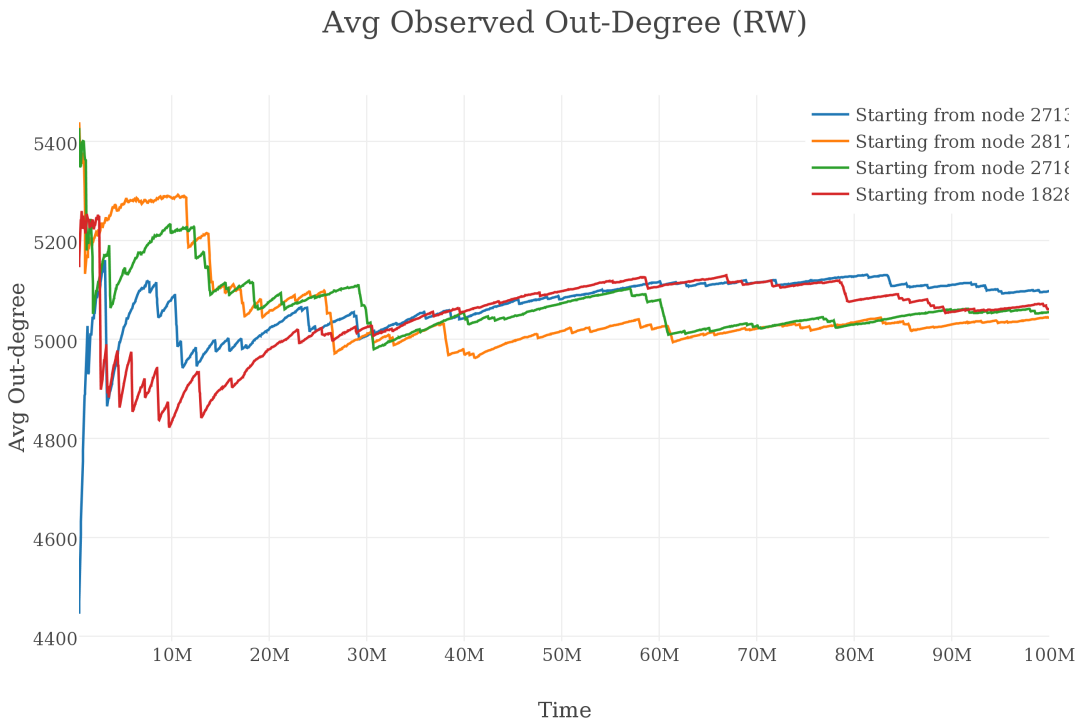

Figure 6.3 – Observed average out-degree of four random walks on the core. The observed values seem to converge toward about 5 150, which is far higher than the true value of the average degree (41.65).

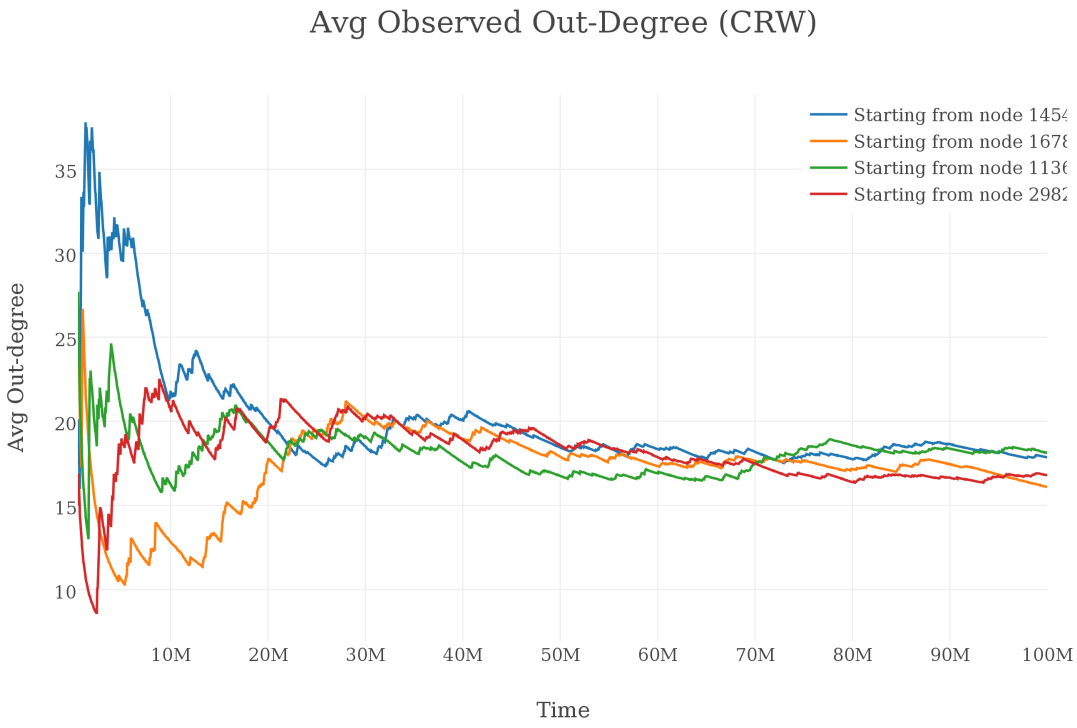

Figure 6.4 – Observed average out-degree of four CRW modified random walks using the number of colinks at each node as their transition criteria.

at each node as its transition criteria (CRW). The goal is to favor the nodes which are visited less frequently during a random walk, by avoiding the nodes having a large number of colinks. We define a Markov Chain whose transition probabilities are given by:

$$P_{uv} = \begin{cases} \frac{1}{k_u} \min(1, \frac{i_u}{i_v}) & \text{if } (u,v) \in E,\ i_v \neq 0 \\ \frac{1}{k_u} \max_{w \neq u} P(u,w) & \text{if } (u,v) \in E,\ i_v = 0 \\ 1 - \sum_{w \neq u} P_{uw} & \text{if } w = u \\ 0 & otherwise \end{cases}$$

In figure 6.4, we observe that our CRW random walk experimentally converges faster, and toward a value which is much closer to the true average degree than a usual random walk (see figure 6.3).

6.6.2 The Icebreaker heuristic

Our *Icebreaker heuristic* (IH) consists in selecting first a random node in the graph to be sampled, and then to explore its neighborhood in order to generate a strongly connected seed subgraph. The exploration process is guided by a CRW random walk which can restart at any step to the node selected initially, with a jumping probability α. The intuition is that, at the beginning of the sampling process, we need to make some room to be able to compute a local estimate of the stationary distribution on a sufficiently large graph. The jumping probability α permits to specify the depth of the exploration from the initial node.

Let s denote the initial node, and P_{CRW} the transition probability matrix of the CRW random walk on the graph G. In matrix form, the seed exploration random walk (IHS) can be written as follows:

$$P_{IHS}(s, \alpha) = (1 - \alpha) P_{CRW} + \alpha J e_s^T$$

where J is the square matrix of all ones, and e_s is the indicator vector of node s.

The details of the seed exploration algorithm are given below in the algorithm 6:

Algorithm 6 IH-1

```
# s: starting node
# k: number of seed nodes to select
# α: restart probability of seed exploration
function GetSeedBall(s,k,α)
    # set of explored nodes
    explored ← ∅
    # current node being explored
    cNode ← s
    while |explored| < k do
        # simulate a node from IHS random walk
        cNode ← IHS_{s,α}(cNode)
        if cNode ∉ explored then
            add cNode to explored
    return explored
```

Now, let's detail the core of the algorithm:

Algorithm 7 IH-2

```
# n: size of the desired sample
# k: number of seed nodes to start with
# α: restart probability of seed exploration
function IceBreakerSampler(k,α, n)
    select a node s ∈ G at random
    seeds ← GetSeedBall(s, k, α)
    # initialize sample with the set of vertices of
    # the strongly connected component containing s
    sample ← vertices(SCC(seeds, s))
    while |sample| < n do
        # compute Pagerank vector of sample nodes
        compute π_sample of G[sample] subgraph
        found ← false
        while ¬found do
            # select node of minimum Pagerank
            m ← argmin π_sample
            # simulate a node from CRW random walk
            nV ← CRW(m)
            if nV ∉ sample then
                # get children of nV in G
                children ← GetChildren(nV, G)
                if children ∩ sample ≠ ∅ then
                    found ← true
                    add nV to sample
    return sample
```

The idea here is to use the Pagerank vector π_{sample} of the subgraph induced by the set of already sampled nodes $G[sample]$, in order to bias the exploration process and sample nodes as uniformly as possible. We thus use the vector π_{sample} as a local estimate of the unknown stationary distribution π_G over the whole graph. Our assumption is that a subgraph induced by a near-uniform sample of connected nodes will reproduce some (hopefully many) of the topological properties of the underlying unexplored graph.

Once we have selected a node candidate to be explored in the sample set, we simulate one of its children with a CRW random walk. Finally, we include this new potential node nV in the sample set if this node has at least one child which is part of the sample set. This way, the subgraph $G[sample]$ is guaranteed to stay strongly connected.

6.6.3 Comparison with other works

In a seminal work presented in [8], a method for sampling URLs near-uniformly from the webgraph is described. It consists in crawling the Web by simulating the behavior of a random surfer that can jump at any step with a fixed jumping probability [25], but only to one of the pages crawled previously, as the rest of the graph is not known. Then, the idea is to sample a web page from the crawled subset with a probability inversely proportional to its Pagerank. As the true value of the Pagerank remains unknown, the *visit ratio*, which measures the number of times a page was visited over the length of the crawl, is used as an estimate instead.

In [26], another algorithm for sampling the webgraph is presented. It starts a random walk at a page selected randomly, and after a burning time of N steps, records the set of pages crawled for K additional steps. The stationary probability $\tilde{\pi}_l$ of each page X_l recorded previously is then estimated, by computing the visit ratio of a M steps random walk started at each page. Similarly to the previous case, the pages are finally sampled with a probability inversely proportional to their estimated stationary probability. As in [27], a more simple algorithm which assumes that the web is an undirected graph (and thus that the inbound links can be queried to a search engine) is also described.

6.6.4 Our results

We test our heuristic on the Twitter core with an initial seed set of 2000 nodes, and analyze the basic properties of the subgraph we obtain for a size of 150 000 nodes, *i.e.*, about 0.45% of the total number of nodes.

The first thing we observe is that the average degree of the IH subgraph stabilizes around a value which is close to the true average degree (*i.e.*, 41.65), once its size has reached 100 000 nodes (see figure 6.5). It is interesting to note that, within the context of the whole graph, the sampled nodes have an average in-degree of approximately 1 800, and an average out-degree of approximately 200.

The results we obtain for two runs of the heuristic on the Twitter core are summarized in table 6.6.4. Although the average degree of the sampled subgraphs gets very close to the true average degree, we notice significant differences for certain measures. We observe in particular that the reciprocity and transitivity values are higher in the subgraphs than in the original graph.

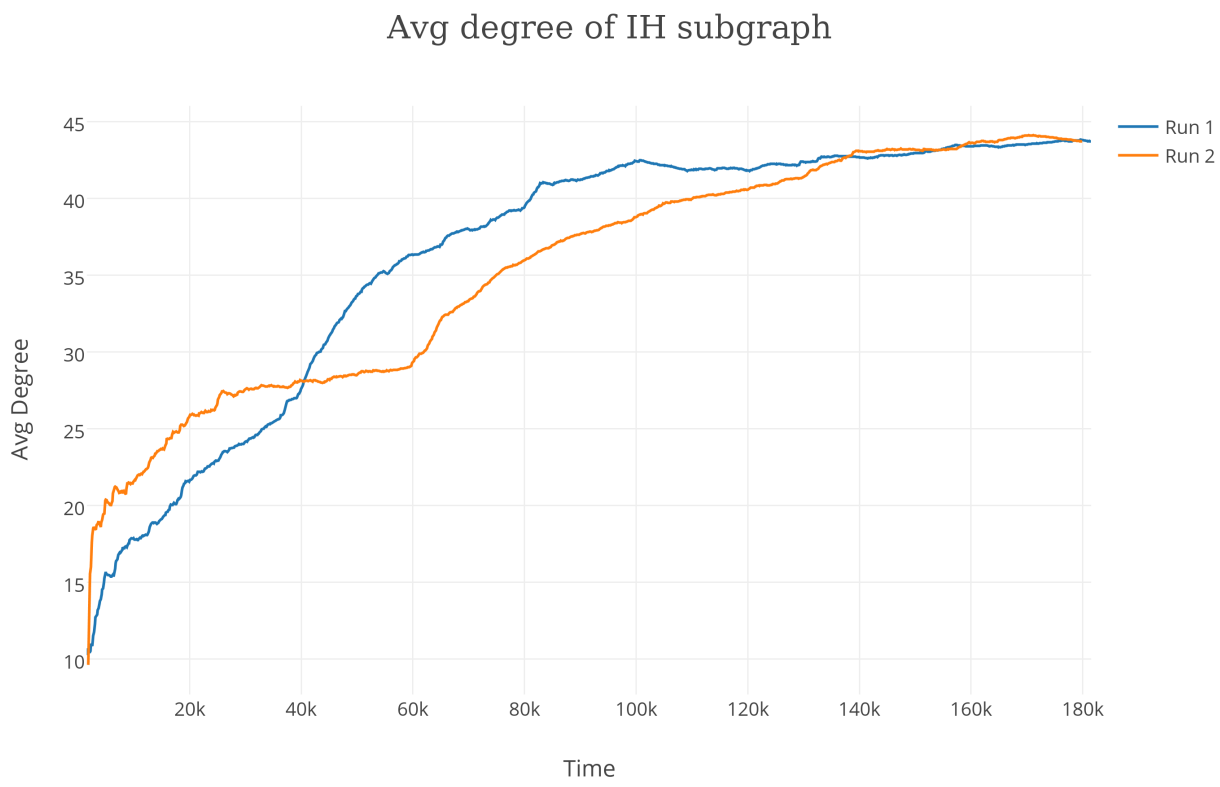

Figure 6.5 – Average degree of the IH subgraph in function of its size. We observe that it stabilizes around the true average degree when the number of nodes is greater than 100 000.

This is to be contrasted with the average colink and average clustering coefficients whose values remain close to their corresponding true values.

It is also interesting to note that, whereas the (o,i)-assortativity is very close to its true value, the degree correlation gets lower values in the sampled subgraphs. This is certainly due to the fact that a large part of the node links is unknown, which tends to make the subgraphs more heterogeneous at the node level than in the original graph.

	core	**run 1**	**run 2**
avg degree	41.65	44.16	43.7
density	1.2e-6	2.3e-4	2.4e-4
reciprocity	0.24	0.76	0.69
avg colink coeff.	0.2	0.12	0.09
transitivity	0.28	0.59	0.54
avg clustering coeff.	0.04	0.04	0.03
degree correlation	0.69	0.24	0.23
(o,i)-assortativity	0.97	0.95	0.95

Table 6.2 – basic measures for two runs of the Icebreaker heuristic on the Twitter core.

6.7 Discussion

6.7.1 Implementation details

A compact graph format

In the case of non streaming graph algorithms, loading the graph being analyzed in memory can improve performance drastically. For conducting our analyses, we use a compact binary format which enables to traverse the graph efficiently and keeps a reasonable memory footprint.

Starting from the adjacency list representation of a graph, we generate two binary files: a data file which contains a list of contiguous children ids, and an index file which associates to each vertex of the graph the starting position of its children in the data file.

We need, at minimum, $\lceil \frac{\lceil log_2(n) \rceil}{8} \rceil$ bytes to represent the n vertices of a graph. This means 4-byte blocks are sufficient to code the vertex ids and vertex positions of the Twitter core. The total amount of memory (in bytes) required to load the Twitter social graph is thus given by

$$m = 4(n + e)$$

which represents approximately 5.3 GB.

Optimization of the algorithm

One of the potential problems when performing a Metropolis-Hastings random walk (MHRW) is to get stuck at a given node during a long time because the transition probabilities associated to its children are very low. As we are not interested to keep track of the time spent at a given node, a simple way to alleviate the problem is to normalize the vector of probabilities for transitions exiting from a given node, so that we are sure to move away at the next step (*i.e.*, the probability to keep at the same place falls down to 0). This is precisely the method we use to simulate nodes efficiently with our custom CRW random walk. Note that in the case of the IHS exploration random walk, we simply need to consider that the starting node is a virtual child of all other nodes with a transition probability equals to the jumping constant.

Furthermore, we need to avoid computing the whole Pagerank vector each time a new node is added to the set of sampled vertices. We suggest for this to use a Monte Carlo approach to allow for the fast incremental computation of the Pagerank vector [28, 29].

Finally, the exploration process can be optimized by keeping track of the nodes in the sample set and at the frontier of the ball that lead to dead-ends, or do not permit to discover new parts of the graph. In our implementation, the nodes which are part of the sample set and that have no child outside of it are added to an ignore list, whereas the nodes at the frontier that have no child in the sample set are temporarily added to a blocked list, until they get unblocked when one of their children is added to the sample set.

6.7.2 Using a null model

In the following, we consider the classic Erdős-Rényi random graph model $G(n, p)$ as our null model to take into account the contribution of edges appearing simply by chance (*i.e.*, with probability $p = \frac{e}{n(n-1)} \simeq \frac{e}{n^2}$). Such a null model is used for example in the definition of *modularity* [30]. This may indeed be desirable to make our measurements independent of the graph density p [31].

By assuming that their values are necessarily higher in a complex network than in an equivalent random graph, we redefine the local colink and the local clustering coefficients in the following way:

$$C_u^{1*} = \frac{i_u - \eta_u p}{\eta_u (1-p)} \tag{6.7}$$

and,

$$C_u^{2*} = \frac{t_u - \tau_u p}{\tau_u (1-p)} \tag{6.8}$$

When computing the reciprocity and the transitivity, we may also be interested to take into account the proportion of colinks and triangles appearing simply by chance in the graph, and adapt the equations 6.3 and 6.4 accordingly.

6.8 Conclusion

We have discussed various important measures of interest on graphs, and described different deterministic and randomized algorithms that can be used to compute them on massive directed complex networks. We have then presented the *Icebreaker heuristic* (IH) which aims at sampling a "representative" connected subgraph from a yet unexplored directed graph. This task is not trivial as it requires to guide an exploration process with a very limited knowledge of the whole structure of the graph. Our heuristic leverages the limited information at disposal by computing local estimates of the unknown stationary distribution values of the subgraph nodes, and by exploring the frontier of the subgraph with a biased random walk inspired by the classical *Metropolis-Hastings random walk* (MHRW) used for undirected graphs.

We have seen that some measures like the average degree, the average clustering coefficient or the (o,i)-assortativity get close to their true value in the sampled graphs, whereas some others like the reciprocity, transitivity or the degreee correlation differ significantly. The very small size of the sampled subgraphs we used for our tests, as well as their high density compared to the one of the original graph certainly play an important role in the observed discrepancies.

Thus, we think that it would be particularly interesting to test our heuristic on other massive complex networks, and for sampled subgraphs whose relative size is comprised between 5% and 10%. We also believe that studying the behavior of the Icebreaker heuristic on various networks may help to understand the evolution of certain measures with respect to the size and density of the sampled subgraphs, and permit to get new insights into their complex topology.

Bibliography

[1] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?," in *WWW '10: Proceedings of the 19th international conference on World wide web*, (New York, NY, USA), pp. 591–600, ACM, 2010.

[2] J. Dubuisson, J.-P. Eckmann, C. Scheible, and H. Schütze, "The topology of semantic knowledge," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (Seattle, Washington, USA), pp. 669–680, Association for Computational Linguistics, October 2013.

[3] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[4] D. J. Pearce, "An improved algorithm for finding the strongly connected components of a directed graph," tech. rep., 2005.

[5] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM journal on computing*, vol. 1, no. 2, pp. 146–160, 1972.

[6] J. Dubuisson, "Massive Graph Sampler." `https://github.com/jimbotonic/mgs`, 2014.

[7] S. Milojević, "Power law distributions in information science: Making the case for logarithmic binning," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2417–2425, 2010.

[8] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, "On near-uniform url sampling," *Computer Networks*, vol. 33, no. 1, pp. 295–308, 2000.

[9] S. Fortunato, M. Boguñá, A. Flammini, and F. Menczer, "Approximating pagerank from in-degree," in *Algorithms and Models for the Web-Graph*, pp. 59–71, Springer, 2008.

[10] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[11] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[12] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," in *Experimental and Efficient Algorithms*, pp. 606–609, Springer, 2005.

[13] S. Suri and S. Vassilvitskii, "Counting triangles and the curse of the last reducer," in *Proceedings of the 20th international conference on World wide web*, pp. 607–614, ACM, 2011.

[14] L. S. Buriol, G. Frahling, S. Leonardi, and C. Sohler, "Estimating clustering indexes in data streams," in *Algorithms–ESA 2007*, pp. 618–632, Springer, 2007.

[15] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *Proceedings of the 16th international conference on World Wide Web*, pp. 835–844, ACM, 2007.

[16] M. Kurant, A. Markopoulou, and P. Thiran, "On the bias of bfs," *arXiv preprint arXiv:1004.1729*, 2010.

[17] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[18] B. Ribeiro, W. Gauvin, B. Liu, and D. Towsley, "On myspace account spans and double pareto-like distribution of friends," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pp. 1–6, IEEE, 2010.

[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, ACM, 2007.

[20] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 390–403, ACM, 2010.

[21] L. Jin, Y. Chen, P. Hui, C. Ding, T. Wang, A. V. Vasilakos, B. Deng, and X. Li, "Albatross sampling: robust and effective hybrid vertex sampling for social graphs," in *Proceedings of the 3rd ACM International Workshop on MobiArch*, pp. 11–16, ACM, 2011.

[22] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[23] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, "Introducing markov chain monte carlo," *Markov chain Monte Carlo in practice*, vol. 1, p. 19, 1996.

[24] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating pagerank computations," in *Proceedings of the 12th international conference on World Wide Web*, pp. 261–270, ACM, 2003.

[25] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1, pp. 107–117, 1998.

[26] P. Rusmevichientong, D. M. Pennock, S. Lawrence, and C. L. Giles, "Methods for sampling pages uniformly from the world wide web," in *AAAI Fall Symposium on Using Uncertainty Within Computation*, pp. 121–128, 2001.

[27] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz, "Approximating aggregate queries about web pages via random walks," 2000.

[28] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in pagerank computation: When one iteration is sufficient," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 890–904, 2007.

[29] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized pagerank," *Proceedings of the VLDB Endowment*, vol. 4, no. 3, pp. 173–184, 2010.

[30] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[31] D. Garlaschelli and M. I. Loffredo, "Patterns of Link Reciprocity in Directed Networks," *Physical Review Letters*, vol. 93, Dec. 2004.

Chapter 7

Conclusion

Throughout this thesis, we have seen that the topology of directed complex networks carries essential information about the data these networks model. Although their analysis is known to be especially hard to handle from a mathematical point of view, it remains possible to carry out empirical studies via a combination of computational and statistical approaches. This is to be contrasted with undirected networks whose study benefits from a set of well-studied mathematical tools, but whose topology also carries much less information than their directed counterparts.

In the first part, we have shown how a thorough analysis of the topology of graphs of free associations enables to get a better understanding of the semantics of words. In the second part, we have developed a new method that leverages diffusion processes over datasets modeled as directed graphs, and can be used efficiently for graph mining and classification. In the final part, we have described various algorithms for analyzing massive complex networks, and presented a heuristic for sampling under strong constraints a "representative" subgraph of a yet unexplored graph.

Thus, in these different studies, we have tried to emphasize the importance of looking carefully into the structure of real directed complex networks, as much remains to be understood regarding their topology. Numerous models have been developed to reproduce some of their usual properties, but we believe a thorough analysis of the real data remains an essential process for getting a better understanding of the problem at stake.

Moreover, we have seen that the topology of datasets modeled as directed graphs can be explored and used to great effect by means of random walks, for getting deep insights into the data. Incidentally, our feeling is that much remains to be discovered in this direction, in particular by leveraging the dynamics of diffusion processes over the data.

The set of mathematical tools that are currently available for studying

directed complex networks is rather limited, especially because much more effort has been devoted to the study of undirected networks until now. Fortunately, some recent works [1, 2, 3, 4] have focused on extending the theoretical framework dedicated to the analysis of directed networks, and have thus paved the way for future research.

Although still in its infancy, the study of directed complex networks is fascinating as it spans many branches of science, and has already been applied successfully to many diverse problems. We believe that this area of research may benefit in the future from careful empirical analysis of the topological properties of the networks under consideration, combined with the development of a standardized formalism and the use of an extended mathematical framework.

Bibliography

[1] F. Chung, "Laplacians and the cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.

[2] R. Agaev and P. Chebotarev, "On the spectra of nonsymmetric laplacian matrices," *Linear Algebra and its Applications*, vol. 399, pp. 157–168, 2005.

[3] F. Chung, "The diameter and laplacian eigenvalues of directed graphs," *Electronic Journal of Combinatorics*, vol. 13, no. 4, pp. 1–19, 2006.

[4] Y. Li and Z.-L. Zhang, "Random walks on digraphs, the generalized digraph laplacian and the degree of asymmetry," in *Algorithms and models for the web-graph*, pp. 74–85, Springer, 2010.