**Work Package 3**
**Deliverable 3.4**

# Multidimensional and Multiscale Analysis of Interactions in Social Systems

Hông-Lan Botterman, Robin Lamarche-Perrin[*], Matthieu Latapy[†], Clémence Magnien[‡], Léonard Panichi, Yiannis Siglidis, Tiphaine Viard, and Audrey Wilmet

*Sorbonne Université (formerly UPMC) and CNRS*

June 29, 2019

**Executive Summary**

This deliverable summarises the work conducted within the ODYCCEUS project about formalisms, methods, and tools for the multidimensional and multiscale analysis of interaction networks. As such, it mainly consists in theoretical research and methodological contributions in computer science, but also contains modest applications to the study of Twitter data for illustration purposes. This document includes in appendix two published and two working papers each addressing a particular research problem and providing theoretical elements of solution. Moreover, we list in this deliverable the implemented tools and pieces of software resulting from this work in order to make the proposed methods operational for social sciences.

[*] robin.lamarche-perrin@lip6.fr

[†] matthieu.latapy@lip6.fr

[‡] clemence.magnien@lip6.fr

# Contents

# 1 Introduction

This deliverable **D3.4**, due at month 30, summarises the work that has been conducted within the ODYCCEUS project about formalisms, methods, and tools for the multidimensional and multiscale analysis of interaction networks. As such, it mainly consists in theoretical research and methodological contributions in computer science (Section 2), but also contains modest applications to the study of Twitter data for illustration purposes. This document includes in appendix two published and two working papers each addressing a particular research problem and providing theoretical elements of solution (Section 4). Moreover, the implemented tools and pieces of software resulting from this work in order to make the proposed methods operational for social sciences (Section 3).

## 1.1 Motivation

Studying the structure of debate in social media and in the public sphere, that is studying the way information and opinions are in practice discussed by individuals, requires sophisticated methods for the analysis of interaction flows between such individuals. Debates are often represented as complex entanglements of social interactions, embedded in space and time, and displaying a multilevel structure: Actors of the debate can be studied at the individual level as well as many mesoscopic collective or institutional levels; Source and destination of interaction flows at several territorial scales; Their dynamics at several temporal scales; And the content of interaction itself can be addressed at many epistemic levels. Computational methods for the data-driven analysis of such systems thus require to cope with these multiple dimensions and multiple scales of analysis. To address such a challenging issue in the case of *structural analysis* of interactions, we are hence developing in the project new graph-theoretical methods for the multidimensional and multilevel analysis of social networks.

In particular, from the point of view of graph theory, we aim at addressing the following theoretical challenges that constitute important barriers to the study of social systems (see summary in Section 2):

- **Dealing with time** for the analysis of interaction dynamics using the *stream graphs* formalism (Appendix A);
- **Dealing with multiple scales of analysis** for the exploration of mesoscopic interaction pattern through the *lossy compression* of stream graphs (Appendix B);
- **Dealing with multiple normalisation models** for the detection of anomalous interaction patterns in stream graphs represented as *multidimensional data cubes* (Appendix C);
- **Dealing with multiple types of nodes and links** for the explanation of complex interaction patterns in *heterogeneous information networks* (Appendix D).

## 1.2 Role in the Project

This deliverable participates to task **T3.3** about "multilevel dynamical networks for the analysis of opinion dynamics and geopolitical conflicts". In this task, we aim at developing methods for analysing the structure of interactions in social media (*e.g.*, polarization, leadership, solidarity effects) and to disentangle diverging representations of geopolitical conflicts, modelled as complex interaction networks between stakeholders embedded in space and time at multiple scales. As stated above, this deliverable provides theoretical and methodological solutions to this end, thus fully participating to **WP3**, as well as associated software solutions that will be integrated within the PENELOPE platform developed in **WP4**.

Note that the four challenges hereabove identified result from conceptual discussions that have been conducted in **WP1** (in particular in **T1.3** and **T1.5**) and the resulting solutions have strong application objectives regarding the various case studies of **WP5** (in particular with respect to **T5.2**, **T5.3**, and **T5.6**).

- First, we aim with this methodological contribution to allow for the verification of hypotheses in *sociology of social media* though the empirical analysis of opinion dynamics, and in particular by studying the structure of different polarisation, leadership, communitarianism, filter bubbles, and solidarity patterns that have been formalised in **T1.5**. This empirical work will be conducted in **T5.6**.
- We also aim at the verification of hypotheses in *sociology of mass media* through the analysis of conflicting world views, and in particular through the concept of "geographic media agenda" which as been formalised in **T1.3** to model international conflicts as territorial information networks. This empirical work will be conducted in **T5.2** and **T5.3**. Note that preliminary work on this matter has already been conducted in the form of a step-by-step tutorial presenting one of our software library on a small geo-media dataset:
  https://lamarche-perrin.github.io/data.cube/tutorial/

- Moreover, to evaluate the potential impact of some of these tools, preliminary experiments have been conducted on a dataset of articles related to climate change debates (**T5.5**). This first application is made available in our online prototype of *Multidimensional Outlier Explorer* (first select the dataset of Guardian articles and comments):
  https://penelope.huma-num.fr/apps/data.cube/outlier-explorer.app/

Finally, methods presented in this deliverable will need to interact with other methodological research conducted in **WP3** when applied to these use cases.

- First, regarding task **T3.3**, this work will be completed by deliverable **D3.3** about algorithms for the extraction of signed social network from textual data. This complementary work will hence deal with a fifth methodological challenge, that is the addition of information about network's interactions to take into account the polarity of views between actors of the network (represented by positive or negative ties).

- More generally, methods presented in this deliverable are dedicated to the *structural analysis* of interactions, but will later need to be integrated with methods dedicated to the *content analysis* of interactions, such as the ones provided by precision language processing of political positions (**T3.1**) and by statistical inference of conceptual spaces with topic models (**T3.2**).

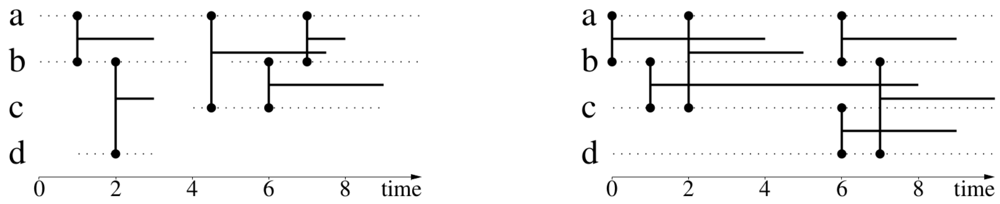
## 2 Work Done

This section provides a summary of the work that has been conducted for this deliverable. It is then presented in much more details in the four appendices (see Section 4).

The starting point of this work on the structure of social interactions is graph theory. In this well-established formal framework for the transverse study of networks, relations, and interactions, a *graph* $G = (V, E)$ is defined by a *set of nodes* $V$ (or vertices), for example representing the system's individuals, and a *set of links* $E \subseteq V \times V$ (or edges), representing their relations or interactions. We will not revisit in this deliverable the many concepts, methods, measures, and algorithms that have been researched in this field during the last decades, but rather focus on the four challenges that have been identified above.

### Dealing with time for the analysis of interaction dynamics

The structure of social systems is constantly evolving under the influence of social processes it organises. To study the bilateral coupling of structure and time, traditional graph theory however seems quite limited. Indeed, methods for the analysis of evolving graphs – that is graphs which nodes and links may appear or disappear through time – are often based on the analysis of sequences of "snapshot" graphs with the usual tools of *static* graph theory: (i) Time is first partitioned into meaningful disjoint intervals; (ii) Independent measurements of the corresponding aggregated structures are then performed; (iii) Results of these measurements are lastly merged to study their evolution through time. Building on signal processing, other approaches advocate for the inverse procedure: (i) Structure is first decomposed into meaningful disjoint units; (ii) Temporal evolution of these units is then independently studied; (iii) Correlation between the resulting time series is lastly highlighted to recover structural dependencies. In both cases, the analysis of *structure* and *time* is broken down into two consecutive steps.



*Two small examples of stream graphs representing lasting interactions between four agents through time (extracted from Appendix A)*

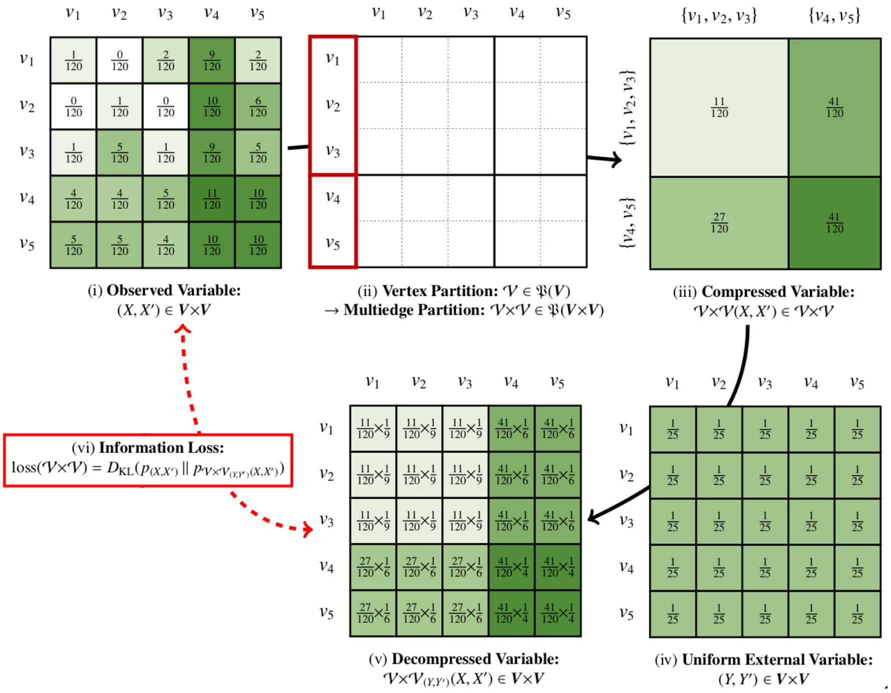
Yet, to fully understand the dynamics of interactions, that is the bilateral coupling of temporal processes and structural patterns, one needs to simultaneously address the system's structural and temporal dimensions. To this end, the "stream graph" framework precisely constitutes a recent attempt to generalise traditional concepts and measures of *static* graph theory – such as graph density, cliques, communities, paths, distances – to the realm of *dynamical* graph theory (see Appendix A). In this framework, a *stream graph* $S = (T, V, W, E)$ is defined by a *time interval* $T = [t_{\text{start}}, t_{\text{end}}] \subset \mathbb{R}$ representing the time span of the graph, a set of *nodes* $V$, a set of *temporal nodes* $W \subseteq T \times V$ representing the time instants when nodes are present in the stream, and a set of *temporal links* $E \subseteq T \times V \times V$ representing the time instants when nodes are interacting in the stream. Hence, $(t, u, v) \in E$ means that node $u$ interacts with node $v$ at time $t$.

From this starting formalism, stream graph theory generalises traditional graph concepts in order to cope with structure and time in a consistent way. Appendix A starts with elementary concepts such as density, clusters, or paths, and then derives more advanced concepts such as cliques, degrees, clustering coefficients, or connected components. One thus builds a consistent language to directly deal with interactions over time, similar to the language provided by graphs to deal with static relations. Indeed, the strength of stream graph theory is that this formalism is also consistent with static graph theory: Graph concepts are special cases of the ones that are introduced by stream graphs, which strongly facilitate the generalisation to higher-level objects such as quotient graphs, line graphs, $k$-cores, and centralities.

### Dealing with multiple scales for the exploration of mesoscopic interaction patterns

The structure and dynamics of real, large-scale, complex networks cannot be grasped at first glance. Indeed, the mere size of such systems – connecting millions of individuals by billions of relations during years – requires a change of perspective. More importantly, the structure and dynamics of social systems are better understood when simultaneously considering them at multiple scales: Interactions are both impacted, on the one hand, by the system's macroscopic structures and global trends and, on the other hand, by crucial microscopic events and individual relations. Hence, abstraction tools are often required to go beyond the mere microscopic analysis of interactions and to unravel other meaningful scales of analysis.

To this end, *graph compression* is a data analysis technique that consists in the replacement of parts of a graph by more general structural patterns in order to reduce its description length. Appendix B presents a framework for the lossy compression of stream graphs. It first builds on a simple and limited scheme, exploiting structural equivalence for the *lossless compression of simple graphs*: The objective is to partition the set of nodes $V$ into several parts $\mathcal{V} = \{V_1, \ldots, V_n\}$ such that, in each part $V_i$, all nodes have the exact same neighbourhood. These equivalent classes thus allow to summarise relational data without losing any information. The appendix then presents several generalisations of this initial problem to address the *lossy compression of stream graphs*.


$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$v_1$: $\frac{1}{120}$ $\frac{0}{120}$ $\frac{2}{120}$ $\frac{9}{120}$ $\frac{2}{120}$
$v_2$: $\frac{0}{120}$ $\frac{1}{120}$ $\frac{0}{120}$ $\frac{10}{120}$ $\frac{6}{120}$
$v_3$: $\frac{1}{120}$ $\frac{5}{120}$ $\frac{1}{120}$ $\frac{9}{120}$ $\frac{5}{120}$
$v_4$: $\frac{4}{120}$ $\frac{4}{120}$ $\frac{5}{120}$ $\frac{11}{120}$ $\frac{10}{120}$
$v_5$: $\frac{5}{120}$ $\frac{5}{120}$ $\frac{4}{120}$ $\frac{10}{120}$ $\frac{10}{120}$
(i) **Observed Variable:** $(X, X') \in V \times V$
(ii) **Vertex Partition:** $\mathcal{V} \in \mathfrak{P}(V)$
→ **Multiedge Partition:** $\mathcal{V} \times \mathcal{V} \in \mathfrak{P}(V \times V)$
$\{v_1, v_2, v_3\}$ $\{v_4, v_5\}$
$\{v_1, v_2, v_3\}$: $\frac{11}{120}$ $\frac{41}{120}$
$\{v_4, v_5\}$: $\frac{27}{120}$ $\frac{41}{120}$
(iii) **Compressed Variable:** $\mathcal{V} \times \mathcal{V}(X, X') \in \mathcal{V} \times \mathcal{V}$
$\frac{1}{25}$
(iv) **Uniform External Variable:** $(Y, Y') \in V \times V$
$v_1$–$v_3$: $\frac{11}{120} \times \frac{1}{9}$ $\frac{11}{120} \times \frac{1}{9}$ $\frac{11}{120} \times \frac{1}{9}$ $\frac{41}{120} \times \frac{1}{6}$ $\frac{41}{120} \times \frac{1}{6}$
$v_4$–$v_5$: $\frac{27}{120} \times \frac{1}{6}$ $\frac{27}{120} \times \frac{1}{6}$ $\frac{27}{120} \times \frac{1}{6}$ $\frac{41}{120} \times \frac{1}{4}$ $\frac{41}{120} \times \frac{1}{4}$
(v) **Decompressed Variable:** $\mathcal{V} \times \mathcal{V}_{(Y,Y')}(X, X') \in V \times V$
(vi) **Information Loss:** $\text{loss}(\mathcal{V} \times \mathcal{V}) = D_{\text{KL}}(p_{(X,X')} \parallel p_{\mathcal{V} \times \mathcal{V}_{(Y,Y')}(X,X')})$

*Compression scheme for static multigraphs using Kullback-Leibler divergence to control information loss (extracted from Appendix B)*

- **Dealing with multigraphs.** A simple stochastic model of multigraphs is introduced using a couple of random variables $(X, X') \in V \times V$ to designate the two nodes that are associated to a given link. These variables follow the empirical distribution of links in the observed graph. In other words, the probability $\Pr((X, X') = (u, v))$ that a given link is randomly associated to nodes $(u, v) \in V \times V$ is the number of links between $u$ and $v$ divided by the total number of links in the observed graph (see matrix (i) in the figure above).

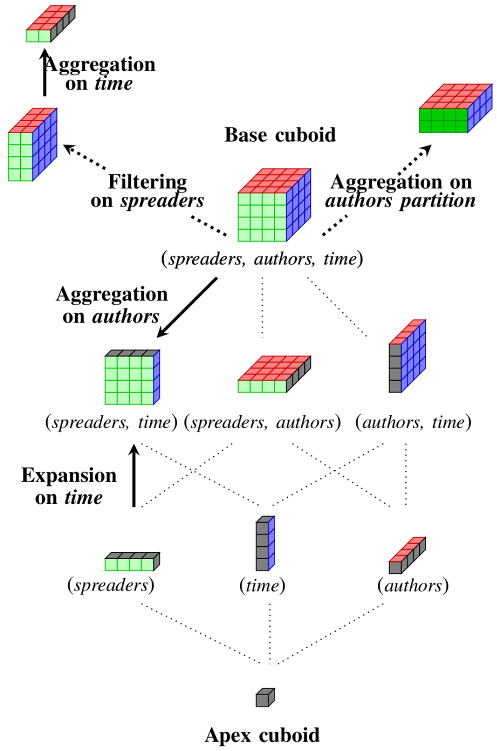
- **Dealing with information loss.** Partition $\mathcal{V} = \{V_1, \ldots, V_n\}$ of $V$ then allows to partition the set of multiedges $V \times V$ (ii) and so to compress variables $(X, X')$ into $\mathcal{V} \times \mathcal{V}$ (iii). The resulting compressed variables are then uniformly "projected back" into $V \times V$ (iv). One ends up with a probability distribution which approximates the initial distribution of links (v). Information about links' position within the observed graph can then be evaluated with measures from information theory, and in particular with Kullback-Leibler divergence which measures and controls the quantity of information that has been lost during this compression/decompression process (vi).

- **Dealing with time.** Lastly, this framework is generalised to stream graphs by adding a third dimension to the formalism (following the generalisation principles of stream graph theory). One now deals with the compression of random variables $(Y, X, X') \in T \times V \times V$.

The resulting combinatorial optimisation problem allows to reduce the description length of interaction datasets by compressing the structural and temporal information they contains. A scaling parameter, expressing the trade-off between information loss and length reduction, finally allows to reveal multiple description scales at which different analyses can be performed. Moreover, Appendix B proposes an algebraic characterisation of the space of partition patterns of $T \times V \times V$ to enhance the expressiveness of the compression scheme. These contributions finally lead to the definition of a combinatorial optimisation problem for which we provide an exact (exponential-time) algorithm.

### Dealing with multiple normalisation models for the detection of anomalous interaction patterns

In social platforms such as Twitter, users may interact with each others, spread information, or exchange opinions via tweets and retweets. Finding salient patterns in such large and complex interaction data requires to first define what constitutes an unexpected observation, a.k.a. an *outlier*. As stream graphs are tridimensional objects, there are numerous ways to define what constitutes an outlier: Are we interested in most active users, that is the ones that spread a lot of content through retweets? Are we instead interested in most popular users, that is the ones whose content is most retweeted? Are we interested in periods of high activity, regardless of who is retweeting whom at that time? Or, are we interested in more sophisticated patterns: Time periods when some users are more popular than average; Users which are more active than normally at a given time compared to their usual behaviour; Relationships between



*Algebraic operations on the data cube of Twitter interactions (extracted from Appendix C)* particular couples of users which appear relatively strong during a given hour when compared to other relationships during the same day; And so on. Many dimensions imply many analysis choices.

To consistently address these various, yet related questions, Appendix C focuses on the detection of statistical outliers in multidimensional data, and in particular on the detection of outliers in stream graphs represented as *data cubes* (see figure on the right). By filtering and aggregating the three dimensions of analysis ($T \times V \times V$), one can define numerous "normalisation contexts" which all lead to different expectation models of the observed variable (*e.g.*, the observed number of retweets $r : T \times V \times V \to \mathbb{N}$). Each model, taking into account marginal totals of the variable according to different dimensions (*e.g.*, $r(.,u,.) = \sum_{(t,v) \in T \times V} r(t,u,v)$ is the total number of retweets published by user $u$) then leads to a particular outlier definition. This approach can hence be seen as a multidimensional generalisation of statistical analysis of contingency tables. For example,

$$r^*(t,u,v) \quad = \quad \frac{r(.,.,.)}{|T|} \, \frac{r(.,u,.)}{r(.,.,.)} \, \frac{r(.,.,c)}{r(.,.,.)}$$

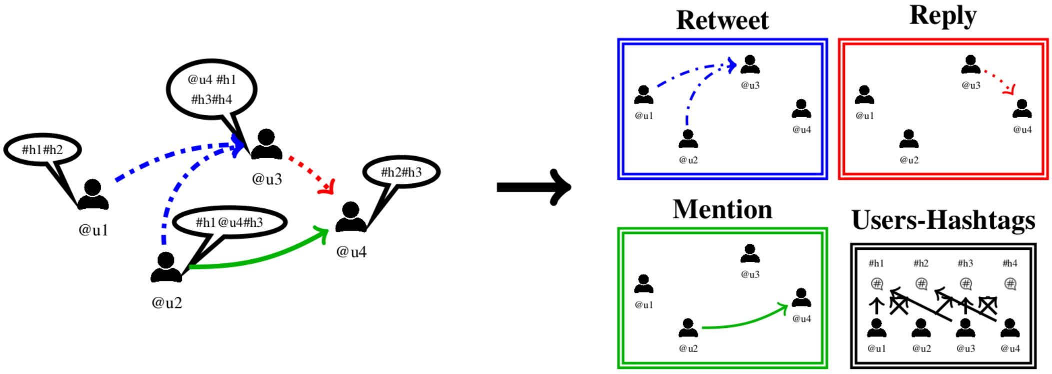
measures the expected number of times user $u$ retweeted user $v$ at time $t$ (term on the left-hand side) knowing the averaged activity of $u$ (second term on the right-hand side) and the averaged popularity of $v$ (third term on the right-hand side).

Focusing on the study of political communication on Twitter, Appendix D then shows how one can browse and select such models in order to identify different structural and temporal objects of interest in stream graphs: Global media events, political figures and political activists, particularly strong interactions between these actors, and so on. These various outliers might then constitute interesting starting points for a more advanced sociological investigation.

### Dealing with multiple types of nodes and links for the explanation of complex interaction patterns

When studying social systems, one has to deal with various types of actors and with various types of relations connecting them. Each of these relations has a particular sociological meaning and analysis methods should address this semantic variety. For example, when studying opinion dynamics on the Web, several intertwined systems could be distinguished: Content exchanges on a particular social platform (such as Twitter) where interactions can take various forms (retweets, replies, mentions); Online media outlets and their audiences, connecting people to different sources of information; Interaction between media outlets themselves, for example through joint references to reports from news agencies (such as AFP reports); As well as more abstract networks such as socio-semantic networks connecting words and concepts depending on their proximity, and connecting people to these concepts depending on their particular interests. Understanding the structure and dynamics of such a social system hence requires to account for the dynamical coupling of all these dependent subsystems.

To this end, Appendix D presents a method to explain (or even to predict) the strength of links in *heterogeneous information networks* (HIN), that is networks made of multiple types of nodes and of multiple types of links. Formally, graph $G = (V, E)$ is enriched with two labelling functions $\phi : V \mapsto \mathcal{V} = \{V_1, \ldots, V_n\}$ and $\psi : V \mapsto \mathcal{E} = \{E_1, \ldots, E_m\}$



*Example of heterogeneous information network with two types of nodes and four types of links for the study of Twitter interactions (extracted from Appendix D)*

respectively associating a type to each node and a type to each link. The objective is then to train a data-based model that will estimate the strength of links of a given type $E_i$ knowing the strength of links of some other types $E_1, E_2 \dots$ For example, in the figure above regarding interactions on Twitter, one could try to estimate the thematic interests $h \in H$ of a given user $u \in U$ (*i.e.*, the links $(u, h) \in E_{UH}$ connecting him or her to different hashtags) from the thematic interests of the people he or she interacts with (*i.e.*, the links $(u, v) \in E_{\text{retweet}}$ or $(u, v) \in E_{\text{mention}}$ connecting him or her to other users, as well as the links $(v, h) \in E_{UH}$ connecting these other users to hashtags). The presented method then exploits random walks in the HIN constrained by *meta-paths*, that is paths defined at the link-type level and expressing a particular interaction semantics. For example, $V \xrightarrow{E_{\text{mention}}} V \xrightarrow{E_{UH}} H$ is the meta-path related to the thematic interests of people that the studied users mention on Twitter.

A simple greedy algorithm is presented to search for the most informative meta-paths and to combine them in a linear regression model. This allows for the study of correlation between the various graph substructures embedded in the HIN, and to eventually exploit these correlations to build a predictive model. More importantly, the resulting model is characterised by a weighted selection of meta-paths which each conveys a precise interaction semantics. We hence believe that this framework is promising to express and test various sociological hypotheses about complex interaction networks.

## 3 Incorporation in PENELOPE

These intertwined lines of methodological research led to various pieces of software that will progressively be incorporated in the PENELOPE platform (**WP4**). These tools will thus ensure that the theoretical work that have been conducted in this deliverable is made available for empirical research that will be conducted by case studies (**WP5**). These pieces of software include:

- **Low-level implementations for algorithmic research.** These mainly aim at prototyping and evaluating algorithmic solutions that have been proposed during the project. They are hence intended for researchers in computer science that would like to study, test, and improve these algorithms.

- **Specialised libraries for empirical research.** This second layer of software allows for a higher-level exploitation of our theoretical work. It consists in interactive libraries for data analysis, developed in R or in Python, that integrate the aforementioned algorithmic solutions within a more accessible framework. The targeted audience of these libraries hence are advanced students and researchers in quantitative social sciences.

- **User-friendly services for a larger audience.** Building on these libraries, we are finally designing ready-to-use components for the PENELOPE platform. They consists in RESTful Web services, along with their API specifications, and in dedicated interfaces providing user-friendly "observatories" to a much broader audience (journalists, citizens, decision makers).

Note that some of theses pieces of software are still under development and will continue evolving until the end of the project. All code that has been produced is (or will soon be) fully documented. This includes developer documentation (internal comments, class architecture, API specifications) and user documentation (tutorials, case studies). Moreover, all source code is freely available on the Web under the GNU General Public License (https://www.gnu.org/licenses/gpl-3.0.html).

### An R library for the processing of multidimensional data

Building on the work presented in Appendix C, we are currently developing data.cube, an R library for the analysis of multidimensional datasets. It is mainly a tool for data exploration intended to researchers in quantitative social sciences. It provides a first glance at large-scale datasets and allowing to identify research hypotheses to be later tested. In practice, this library defines a new data structure – called data.cube – cleverly encoding lists of *observations* according to several *dimensions* and *variables*. Various functions then allow to handle the cube by selecting / joining / filtering / arranging / and then plotting these dimensions and variables, thus generalising the tibble data structure and related standards of the tidyverse (https://www.tidyverse.org/).

- Sources on GitHub:
  https://github.com/Lamarche-Perrin/data.cube

- Step-by-step tutorial:
  https://lamarche-perrin.github.io/data.cube/tutorial/

### Web services for the detection of outliers in multidimensional data

Building on the core functions of data.cube, statistical outliers can then be identified within multidimensional observations (see Appendix C). Depending on the selected dimensions of interest, marginal variables allow for the definition of various normalisation models (similar to the ones usually exploited for the analysis of bidimensional contingency tables). Statistical testing then allows to measure model deviations and to retrieve lists of significant outliers. These particular functions led to the development of a PENELOPE component and of a prototype observatory both aiming at the detection of statistical outliers in multidimensional data.

- R Plumber Web service hosted by Huma-Num (www.huma-num.fr):
  https://penelope.huma-num.fr/tools/

- API specification on SwaggerHub:
  https://app.swaggerhub.com/apis-docs/Lamarche-Perrin/outlier-explorer/1.0.1

- R Shiny Web interface hosted by Huma-Num (www.huma-num.fr):
  https://penelope.huma-num.fr/apps/data.cube/outlier-explorer.app/

### A Python implementation of stream graphs

As tridimensional objects, stream graphs (see Appendix A) could be partly handled by the data.cube library. Yet, this is only true for discrete-time stream graphs and, more importantly, this library is not specialised in graph-theoretic analysis. In order to make all concepts of the stream graph formalism operational, we hence started the development of a dedicated Python library in November 2018, named stream_graph. It firstly provides a consistent class architecture to represent the various types of stream graphs: Discrete-time vs. continuous-time, instantaneous links vs. links with duration, undirected vs. directed links, binary vs. weighted links. Then, it implements most concepts and metrics that have been generalised from static graph theory: *E.g.*, neighbourhood, degree, density, paths, distances. Lastly, it also aims in the future at implementing various algorithms for stream graph analysis: *E.g.*, maximal cliques, shortest paths, pattern enumeration. These algorithms have been developed during the last few years in the "Complex Networks" team of the *Laboratoire d'informatique de Paris 6*: http://www.complexnetworks.fr/papers/

- Sources on GitHub:
  https://github.com/ysig/stream_graph

- Documentation, API, and tutorials:
  https://ysig.github.io/stream_graph/doc/

### C++ algorithms for the lossy compression of stream graphs

Lastly, we worked on a C++ implementation of the algorithm presented in Appendix B for the lossy compression of stream graphs. As this exact algorithm is exponential in the general case, we also began to work on heuristics for the approximate solving of this combinatorial optimisation problem on larger graph instances. While still in development, we expect these algorithms to later be included in the stream_graph library in order to allow for the multiscale representation of interaction networks.

- Exact combinatorial algorithm on GitHub:
  https://github.com/Lamarche-Perrin/multidimensional_compression

- Greedy algorithms on GitHub:
  https://github.com/Lamarche-Perrin/greedy-graph-compression

# 4 Appendices

## A Stream graphs and link streams for the modeling of interactions over time (p. 13–31)

This paper has been published in *Social Network Analysis and Mining*.

M. Latapy, T. Viard, and C. Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8:1–29, 2018.
https://arxiv.org/abs/1710.04073

## B An information-theoretic framework for the lossy compression of link streams (p. 32–55)

This paper has been published in *Theoretical Computer Science*.

R. Lamarche-Perrin. An information-theoretic framework for the lossy compression of link streams. *Theoretical Computer Science*, 2018.
https://arxiv.org/abs/1807.06874

## C Multidimensional outlier detection in temporal interaction networks (p. 56–71)

This paper has been submitted to *Social Network Analysis and Mining*.

A. Wilmet and R. Lamarche-Perrin. Multidimensional outlier detection in temporal interaction networks: An application to political communication on twitter. *ArXiv*, abs/1906.02541, 2019.
https://arxiv.org/abs/1906.02541

## D Link weights recovery in heterogeneous information (p. 72–84)

A short version of this paper has been published in *Complex Networks X*, proceedings of the 10th Conference on Complex Networks (CompleNet'19). This longer version (in progress) will soon be submitted to *Social Network Analysis and Mining*.

H.-L. Botterman and R. Lamarche-Perrin. Link weights recovery in heterogeneous information networks. *ArXiv*, abs/1906.11727, 2019.
https://arxiv.org/abs/1906.11727

# Stream Graphs and Link Streams for the Modeling of Interactions over Time

Matthieu Latapy [1], Tiphaine Viard, Clémence Magnien

### Abstract

Graph theory provides a language for studying the structure of relations, and it is often used to study interactions over time too. However, it poorly captures the both temporal and structural nature of interactions, that calls for a dedicated formalism. In this paper, we generalize graph concepts in order to cope with both aspects in a consistent way. We start with elementary concepts like density, clusters, or paths, and derive from them more advanced concepts like cliques, degrees, clustering coefficients, or connected components. We obtain a language to directly deal with interactions over time, similar to the language provided by graphs to deal with relations. This formalism is self-consistent: usual relations between different concepts are preserved. It is also consistent with graph theory: graph concepts are special cases of the ones we introduce. This makes it easy to generalize higher-level objects such as quotient graphs, line graphs, $k$-cores, and centralities. This paper also considers discrete versus continuous time assumptions, instantaneous links, and extensions to more complex cases.

***Keywords***— stream graphs, link streams, temporal networks, time-varying graphs, dynamic graphs, dynamic networks, longitudinal networks, interactions, time, graphs, networks

[1] Contact author. `Matthieu.Latapy@lip6.fr`

# Contents

# 1 Introduction

Friendship, dependencies, similarities, or connections are typical examples of **relations** modeled by **graphs** or networks, *i.e.* sets of nodes and links: nodes represent individuals and two individuals are linked together if they are friends; nodes represent companies and they are linked together if they signed contracts with each other; nodes represent documents like web pages or articles, and they are linked together if they are similar; nodes represent computer devices and they are linked together if there is a wire between them; etc.

For decades, graph theory, social network analysis and network science have developped a wide set of tools for the study of such graphs. In particular, they developed a *language* for describing networks, with elementary yet powerful concepts such as node degree (their number of links), paths (sequences of links going from one node to another one), density (the fraction of pair of nodes actually linked together), or cliques (sets of nodes all pairwise linked together). This language forms the basis of network studies, and there is a global consensus on a wide set of concepts that are used in the field; with few variations, all courses and reference books on graphs and networks start with them, see for instance [4, 8, 85, 87, 17, 50, 18, 1, 64]. Then, more advanced and specific concepts are defined on this common ground.

Contacts, shopping, travels, or traffic are typical examples of **interactions** that take place over time, *i.e.* **streams** of nodes and links active during specific periods of time: nodes are individuals linked together whenever they call each other; nodes are clients and products linked together when a client buys a product; nodes are places linked together when someone moves from one place to another; nodes are internet devices linked together when they exchange data; etc.

Such sequences of interactions play a key role in many areas, and they have been studied for a long time, see related work in Section 21. Although many variations exist, the most common approach is to model them by sequences of graphs (each graph then aggregates the interactions that occurred during a period of time), by labeled graphs (each link being labeled with its presence times), or other augmented graphs. This makes it possible to use graph theory to study these sequences of graphs, labeled graphs, and other variants. Other works deal directly with higher-level methods for studying graphs, like stochastic block models for instance, and extend them to cope with the dynamics. Finally, a few works define specific properties combining temporal and structural information, such as centrality measures for instance.

In this paper, we propose a different approach: **we develop a formalism to directly cope with interactions over time, in a way similar to what graph theory does for relations**. This means that we do not transform interactions into graphs, but rather transform graph theory into a theory of interactions over time. We model them as *link streams* and *stream graphs* (depending on whether the dynamics is on links only, or on both nodes and links), so named in order to emphasize their streaming nature and the fact that they are *not* graphs or networks. Then, we start with the most elementary graph concepts and we define their equivalent for stream graphs and link streams. Finally, we elaborate on these basic concepts to extend more complex graph concepts. With the aim to make our formalism as intuitive as possible, we put much effort in proposing simple definitions, explaining them with different points of view (especially combinatorial and probabilistic ones), and to provide illustrations and detailed examples of all key concepts we introduce.

In addition to these subjective features, we also put much emphasis on two more objective features to ensure the relevance of our definitions. First, we want our formalism to be a generalization of graph theory in a very precise sense: when the stream has no dynamics, it is equivalent to a classical graph and its properties should be the same as those of this graph (see the end of Section 3). Second, we want the relations that exist between various graph properties (between density and degree for instance) to still hold for stream properties. Similarly, if a graph concept is derived from another one (like clustering coefficient from density for instance) we want the corresponding stream concept to be derived from the corresponding other stream concept. **These features ensure both the self-consistency of our formalism and its consistency with graph theory.**

After Section 2 that introduces a few notations needed in the whole paper, we present our framework from Section 3 to Section 17. Each of these sections is devoted to a key concept of graph theory that we redefine in the stream context. Therefore, they all have the same structure: first we recall the relevant graph concepts and their key properties, in italics; then we introduce equivalent concepts for stream graphs with detailed examples and discuss their properties; we introduce additional related concepts specific to stream graphs; we discuss the case of link streams, *i.e.* when there is no dynamics on nodes; and we show that the newly introduced stream concepts are equivalent to the graph ones, whenever this makes sense. After these core sections, we show how our framework may be used under either discrete and continuous modeling of time in Section 18; we show how it generalizes $\Delta$-analysis and may be used with instantaneous links in Section 19; we show how it may be extended to bipartite streams and other particular cases in Section 20; and we present related work in Section 21. We discuss our contributions and future work in Section 22.

# 2 Preliminary notations

In this paper, we rely on a few notations that we introduce below.

Given two finite sets $X$ and $Y$, one may consider the ordered pairs $(x, y)$ with $x \in X$ and $y \in Y$. Then, $(x, y) \neq (y, x)$ and $(x, x)$ exists if $x \in X$ and $x \in Y$. One may also consider unordered pairs $xy$ with $x \in X$ and $y \in Y$, with $x \neq y$. Then, $xy = yx$ and $xx$ does not exist. The **set of ordered pairs** is denoted by $X \times Y$, and one often uses this notation for the set of unordered pairs too. In this paper, however, we use both notions intensively and need to make a clear distinction between them. We therefore denote the **set of unordered pairs of distinct elements** by $X \otimes Y$.

Throughout this paper, we deal with **set sizes**, denoted by $|X|$ for a given set $X$, but the meaning of this notation depends on the type of $X$. If $X$ is an interval $[\alpha, \omega]$ of $\mathbb{R}$, then $|X| = \omega - \alpha$. If it is an interval $[\alpha, \omega]$ of $\mathbb{N}$ then $|X| = \omega - \alpha + 1$. If $X$ is the union of disjoint intervals of $\mathbb{R}$, then $|X|$ is the sum of these intervals' sizes. The same holds if it is the union of disjoint intervals of $\mathbb{N}$. If $X$ is the product of sets of these types, then its size is the product of their sizes. Notice that, if $X$ contains just one element then depending on the context it may be seen as a (degenerate) interval of $\mathbb{R}$ or $\mathbb{N}$, thus having size 0 or 1, respectively. For instance, the union of the intervals $[1, 2]$ and $[3, 3]$ of $\mathbb{R}$ has size 1, while the union of the same intervals of $\mathbb{N}$ has size 3.

Notice that $|X \times Y| = |X| \cdot |Y|$, and so $|X \times X| = n^2$ if $|X| = n$. This is different from $|X \otimes Y| = |(X \setminus Y) \times Y| + |(Y \setminus X) \times X| - |(X \setminus Y) \times (Y \setminus X)| + \frac{|X \cap Y|^2 - |X \cap Y|}{2}$, leading to $|X \otimes X| = \frac{n \cdot (n-1)}{2}$ if $|X| = n$, and $|X \otimes Y| = |X| \cdot |Y|$ if $X$ and $Y$ are disjoint.

# 3 Stream graphs and link streams

*A (simple undirected[2]) graph $G = (V, E)$ is defined by a finite set of nodes $V$ and a set of links $E \subseteq V \otimes V$: $uv \in E$ means that $u$ and $v$ are linked together in $G$.*

*Graphs model relations between nodes. For instance, nodes may represent individuals and links may represent friendship relations. Nodes may represent computers and links may represent physical connections between them. Examples are countless, making graphs the key formalism for studying network structures.*

We define a (simple undirected[2]) **stream graph** $S = (T, V, W, E)$ by a finite set of nodes $V$, a set of time instants $T$, a set of temporal nodes $W \subseteq T \times V$, and a set of links $E \subseteq T \times V \otimes V$ such that $(t, uv) \in E$ implies $(t, u) \in W$ and $(t, v) \in W$. The set of time instants $T$ may be continuous or discrete, which has little influence on the following, as we explain in Section 18. Until then, all the examples we give assume that $T$ is an interval of $\mathbb{R}^+$.

We define $v_t = 1$ if $(t, v) \in W$ and $v_t = 0$ otherwise, as well as $uv_t = 1$ if $(t, uv) \in E$ and $uv_t = 0$ otherwise. When $v_t = 1$ we say that node $v$ is involved in $S$ at time $t$ or that $v$ is present at time $t$, and when $uv_t = 1$ we say that nodes $u$ and $v$ are linked together at time $t$, or that link $uv$ is present at time $t$. We denote by $T_v$ the set of time instants at which $v$ is present, by $T_{uv}$ the set of time instants at which $uv$ is present, by $V_t$ the set of nodes present at time $t$, and by $E_t$ the set of links present at time $t$: $T_v = \{t, v_t = 1\}$, $T_{uv} = \{t, uv_t = 1\}$, $V_t = \{v, v_t = 1\}$, and $E_t = \{uv, uv_t = 1\}$. Notice that $T_{uv} \subseteq T_u \cap T_v$.

If all nodes are present all the time, *i.e.* $T_v = T$ for all $v$ or, equivalently, $V_t = V$ for all $t$, then we say that $S$ is a **link stream** and we denote it by $L = (T, V, E)$ (with $W = T \times V$ implicitly). Indeed, there is no dynamics on nodes in this case, and $S$ is fully defined by this triplet. Link streams play an important role in many situations, and so we pay special attention to this case in all this paper.

We illustrate these definitions in Figure 1 with **drawings** designed as follows. We display node names on a vertical axis on the left of the figure, and time on an horizontal axis at the bottom of the figure. Each node presence times are represented by an horizontal dotted line in front of its name, whenever the node is present. Each link presence times are

[2] Unless explicitly specified, we always consider simple and undirected graphs and stream graphs; we discuss more general cases in Section 20.

represented by an horizontal solid line parallel to the two dotted lines of involved nodes, and a vertical solid line joining these two dotted lines (marked with bullets) when the two nodes start interacting. In Figure 1, for instance, in $S$ (leftmost example) the node $a$ arrives at time 0 and stays until time 10, and so $[0, 10] \times \{a\} \subseteq W$, *i.e.* $T_a = [0, 10]$. This is represented by a dotted line from time 0 to 10 in front of $a$ in the drawing. Likewise, $b$ arrives at time 0, then leaves at time 4, joins again at time 5 and stays until time 10, and so $([0, 4] \cup [5, 10]) \times \{b\} \subseteq W$, *i.e.* $T_b = [0, 4] \cup [5, 10]$. This is represented by a dotted line from time 0 to 4 and another one from time 5 to 10 in front of $b$. These two nodes interact from time 1 to time 3 and from time 7 to time 8, and so $([1, 3] \cup [7, 8]) \times \{ab\} \subseteq E$, *i.e.* $T_{ab} = [1, 3] \cup [7, 8]$. This is represented by a solid line at time 1 between the dotted lines of $a$ and $b$, with an horizontal line starting from its middle until time 3, and another such solid line at time 7 with an horizontal line until time 8.

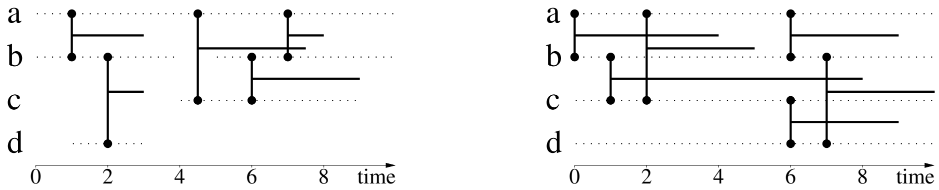


Figure 1: **Simple examples of stream graphs and link streams. Left:** a stream graph $S = (T, V, W, E)$ with $T = [0, 10] \subseteq \mathbb{R}$, $V = \{a, b, c, d\}$, $W = [0, 10] \times \{a\} \cup ([0, 4] \cup [5, 10]) \times \{b\} \cup [4, 9] \times \{c\} \cup [1, 3] \times \{d\}$, and $E = ([1, 3] \cup [7, 8]) \times \{ab\} \cup [4.5, 7.5] \times \{ac\} \cup [6, 9] \times \{bc\} \cup [2, 3] \times \{bd\}$. In other words, $T_a = [0, 10]$, $T_b = [0, 4] \cup [5, 10]$, $T_c = [4, 9]$, $T_d = [1, 3]$, $T_{ab} = [1, 3] \cup [7, 8]$, $T_{ac} = [4.5, 7.5]$, $T_{bc} = [6, 9]$, $T_{bd} = [2, 3]$, and $T_{ad} = T_{cd} = \emptyset$. **Right:** a link stream $L = (T, V, E)$ with $T = [0, 10] \subseteq \mathbb{R}$, $V = \{a, b, c, d\}$, and $E = ([0, 4] \cup [6, 9]) \times \{ab\} \cup [2, 5] \times \{ac\} \cup [1, 8] \times \{bc\} \cup [7, 10] \times \{bd\} \cup [6, 9] \times \{cd\}$. In other words, $T_a = T_b = T_c = T_d = T$ and $T_{ab} = [0, 4] \cup [6, 9]$, $T_{ac} = [2, 5]$, $T_{bc} = [1, 8]$, $T_{bd} = [7, 10]$ and $T_{cd} = [6, 9]$.

Given a stream graph $S = (T, V, W, E)$, we define $G_t = (V_t, E_t)$, the **graph induced by $S$ at time $t$**. In Figure 1, for instance, we obtain for $S$ at time 2 the graph $G_2 = (\{a, b, d\}, \{ab, bd\})$.

We also define $G(S) = (\{v, T_v \neq \emptyset\}, \{uv, T_{uv} \neq \emptyset\}) = (\bigcup_{t \in T} V_t, \bigcup_{t \in T} E_t)$ the **graph induced by $S$**: its nodes are those present in $S$ and they are linked together in $G(S)$ if there exists a time instant in $T$ such that they are linked together in $S$. In other words, it is the graph where there is a link between two nodes if they interacted at least once. In Figure 1, for instance, $G(S) = (\{a, b, c, d\}, \{ab, ac, bc, bd\})$ and $G(L) = (\{a, b, c, d\}, \{ab, ac, bc, bd, cd\})$. One may in addition associate to each node $v$ or link $uv$ a weight capturing a quantity of interest, like for instance their presence duration $|T_v|$ and $|T_{uv}|$.

Stream graphs model interactions between nodes over time, as well as the dynamics of nodes themselves. For instance, nodes may represent individuals present in a given building and links may represent contacts between them. Nodes may represent on-line computers and links may represent data exchanges between them. Examples are countless, and we aim at making stream graphs the key formalism for studying jointly the dynamics and structure of interactions.

Since in a stream graph $S = (T, V, W, E)$ nodes are not present all the time in general, $W$ may differ significantly from $T \times V$. To capture this, we define the **coverage** of $S$ as follows:

$$\text{cov}(S) = \frac{|W|}{|T \times V|}.$$

For instance, in Figure 1 the stream graph $S$ has coverage $\text{cov}(S) = \frac{26}{40} = 0.65$.

Notice that $\text{cov}(S) = 1$ if and only if all nodes are present all the time, and so it is equivalent to saying that $S$ is a link stream.

If in addition for all $u$ and $v$ in $V$, $T_{uv} \in \{\emptyset, T\}$, *i.e.* all existing links are present all the time, then there is no significant distinction between $S$ and $G(S)$, and we say that $S$ is a **graph-equivalent stream**. This gives a formal ground to our wanted feature that stream graphs generalize graphs: we extend graph concepts to stream graphs in a way such that, if a stream graph has a given stream graph property and happens to be a graph-equivalent stream, then this graph has the corresponding graph property. In the following, we systematically check that this feature holds.

# 4 Size, duration, uniformity and compactness

*The number of nodes of a graph $G = (V, E)$ is denoted by $n = |V|$ and its number of links by $m = |E|$.*

Given a stream graph $S = (T, V, W, E)$, we now define its number of nodes and links, as well as its duration. First notice that, unlike in graphs, some nodes may be present for much longer than others. In order to capture this, we define the **contribution of node** $v$ as $n_v = \frac{|T_v|}{|T|}$, which may be seen as the notion of coverage restricted to a node $v$. We then define the **number of nodes** in $S$ as follows:

$$n = \sum_{v \in V} n_v = \frac{|W|}{|T|}.$$

Then, each node contributes to the total number of nodes proportionally to its involvement in $S$: $v$ in $V$ accounts for 1 node only if it is present in $S$ all the time.

We define similarly the contribution of a pair of nodes $uv$ as $m_{uv} = \frac{|T_{uv}|}{|T|}$ and the **number of links** in $S$:

$$m = \sum_{uv \in V \otimes V} m_{uv} = \frac{|E|}{|T|}.$$

Like nodes, each link then contributes to $m$ proportionally to its presence in $S$: $uv$ in $V \otimes V$ accounts for 1 link only if it is present in $S$ all the time.

Finally, we define the **node and link contributions of a time instant** $t$ as $k_t = \frac{|V_t|}{|V|}$ and $l_t = \frac{|E_t|}{|V \otimes V|}$, leading to the following definition of the **node duration** $k$ in $S$ and the **link duration** $l$ in $S$:

$$k = \int_{t \in T} k_t \,\mathrm{d}t = \frac{|W|}{|V|} \quad \text{and} \quad l = \int_{t \in T} l_t \,\mathrm{d}t = \frac{|E|}{|V \otimes V|}.$$

Like the number of nodes $n$ and the number of links $m$, the node duration $k$ may be seen as a duration of $S$ where each time contributes proportionally to the number of nodes present at this time, and the link duration $l$ as a duration of $S$ where each time contributes proportionally to the number of links present at this time.

Notice that $n$ is the expected value of $|V_t|$ when one takes a random time $t$ in $T$. Likewise, $m$, $k$ and $l$ are the expected value of $|E_t|$, $|T_v|$ and $|T_{uv}|$ when one takes a random time $t$ in $T$, a random node $v$ in $V$ or a random pair of nodes in $V \otimes V$, respectively.

The following relation also hold: $\mathrm{cov}(S) = \frac{|W|}{|T \times V|} = \frac{n}{|V|} = \frac{k}{|T|}$, $n \cdot |T| = k \cdot |V| = |W|$, and $m \cdot |T| = l \cdot |V \otimes V| = |E|$.

For the examples in Figure 1, we obtain for $S$ the following values: $n = \frac{|T_a|}{10} + \frac{|T_b|}{10} + \frac{|T_c|}{10} + \frac{|T_d|}{10} = 1 + 0.9 + 0.5 + 0.2 = 2.6$ nodes, $m = \frac{|T_{ab}|}{10} + \frac{|T_{ac}|}{10} + \frac{|T_{bc}|}{10} + \frac{|T_{bd}|}{10} = 0.3 + 0.3 + 0.3 + 0.1 = 1$ link, $k = \frac{26}{4} = 6.5$ time units, and $l = \frac{10}{6} = 1.66...$ time units. For $L$, we obtain $n = 4$ nodes, $m = 0.7 + 0.3 + 0.7 + 0.3 + 0.3 = 2.3$ links, $k = 10$ time units and $l = \frac{23}{6} = 3.833...$ time units.

In a link stream $L = (T, V, E)$, by definition $T_v = T$ for all $v$ in $V$, and so $n_v = 1$ and $n = |V|$. Likewise, for all $t$, $V_t = V$ and so $k_t = 1$ and $k = |T|$. In a graph-equivalent stream, in addition $T_{uv} \in \{\emptyset, T\}$ for all $uv$ in $V \otimes V$ and $E_t$ is the same for all $t$. Then, the number of nodes and links in the stream are equal to the number of nodes and links in the corresponding graph.

Notice now that, in a given stream graph, for two nodes $u$ and $v$ such that $|T_u| = |T_v|$ both $T_u = T_v$ or $T_u \cap T_v = \emptyset$ are possible, as well as all intermediary situations. This has a crucial influence on the possible existence of links between $u$ and $v$, and so on the structure of $S$. In order to capture this, we define the **uniformity** of $S$ as follows:

$$\mathbb{U}(S) = \frac{\sum_{uv \in V \otimes V} |T_u \cap T_v|}{\sum_{uv \in V \otimes V} |T_u \cup T_v|}.$$

If $S$ has uniformity 1, then we say that it is uniform: for all $u$ and $v$ in $V$, $T_u = T_v$, *i.e.* all nodes are present at the same times.

We also define for any pair of nodes $u$ and $v$ in $V$ the uniformity $\mathbb{U}(u, v) = \frac{|T_u \cap T_v|}{|T_u \cup T_v|}$. It measures the overlap between the presence times of $u$ and $v$, thus their ability to be linked together.

Given a stream graph $S = (T, V, W, E)$, we define $S' = (T', V', W, E)$ such that $T' = [\min\{t, \exists (t, v) \in W\}, \max\{t, \exists (t, v) \in W\}]$ and $V' = \{v, \exists (t, v) \in W\}$. We then define the **compactness** of $S$ as follows:

$$c(S) = \frac{|W|}{|T' \times V'|} = \mathrm{cov}(S').$$

If $S$ has a compactness of 1, then we say that it is compact: for all $v$ in $V$, $T_v = [b, e] \subseteq T$, *i.e.* the presence times of all nodes is the same interval of $T$.

For the examples in Figure 1, $S$ has uniformity $\Cup(S) = \frac{\substack{|T_a \cap T_b| + |T_a \cap T_c| + |T_a \cap T_d| \\ + |T_b \cap T_c| + |T_b \cap T_d| + |T_c \cap T_d|}}{\substack{|T_a \cup T_b| + |T_a \cup T_c| + |T_a \cup T_d| \\ + |T_b \cup T_c| + |T_b \cup T_d| + |T_c \cup T_d|}} = \frac{\substack{(4+5)+5+2 \\ +4+2+0}}{\substack{10+10+10+10 \\ +(4+4)+(2+5)}} = \frac{22}{55} = 0.4$ and compactness $c(S) = \mathrm{cov}(S) = \frac{26}{40}$ since on this particular case $T' = T$ and $V' = V$, and so $S' = S$.

If $S$ is a link stream, then its uniformity and compactness are necessarily equal to 1, like $L$ in Figure 1.

# 5 Density

*The density of graph $G = (V, E)$ is the probability when one takes a random element $uv$ in $V \otimes V$ that there is a link between $u$ and $v$ in $E$: $\delta(G) = \frac{2m}{n(n-1)}$. If $n \in \{0, 1\}$ then $\delta(G)$ is defined to be* 0.

We define the **density** of stream graph $S = (T, V, W, E)$ as the probability when one takes a random element $(t, uv)$ of $T \times V \otimes V$ such that $(t, u)$ and $(t, v)$ are in $W$, that $(t, uv)$ is in $E$:

$$\delta(S) = \frac{\sum\limits_{uv \in V \otimes V} |T_{uv}|}{\sum\limits_{uv \in V \otimes V} |T_u \cap T_v|} = \frac{\int\limits_{t \in T} |E_t| \, \mathrm{d}t}{\int\limits_{t \in T} |V_t \otimes V_t| \, \mathrm{d}t}$$

If $\sum_{uv \in V \otimes V} |T_u \cap T_v| = \int_{t \in T} |V_t \otimes V_t| \, \mathrm{d}t = 0$ then we define $\delta(S)$ to be 0.

In other words, the density is the probability when one takes a random time and two random nodes such that a link may exist between them at this time, that the link indeed exists. It is the fraction of possible links that do exist.

Notice that $\sum_{uv \in V \otimes V} |T_{uv}| = \int_{t \in T} |E_t| \, \mathrm{d}t = |E|$. Also, $\sum_{uv \in V \otimes V} |T_u \cap T_v| = \int_{t \in T} |V_t \otimes V_t| \, \mathrm{d}t$ is related to the uniformity $\Cup(S)$ of $S$, but it cannot be directly derived from $|T|$, $|V|$, $|W|$ and $|E|$.

For $S$ defined in Figure 1 (left), $\sum_{uv \in V \otimes V} |T_{uv}| = |T_{ab}| + |T_{ac}| + |T_{bc}| + |T_{bd}| = 3+3+3+1 = 10$, $\sum_{uv \in V \otimes V} |T_u \cap T_v| = |T_a \cap T_b| + |T_a \cap T_c| + |T_a \cap T_d| + |T_b \cap T_c| + |T_b \cap T_d| + |T_c \cap T_d| = 9 + 5 + 2 + 4 + 2 + 0 = 22$, and we obtain $\delta(S) = \frac{10}{22} \sim 0.45$. For $L$ defined in this figure (right), $\sum_{uv \in V \otimes V} |T_{uv}| = 7 + 3 + 7 + 3 + 3 = 23$, $\sum_{uv \in V \otimes V} |T_u \cap T_v| = |V \otimes V| \cdot |T| = 60$ and we obtain $\delta(L) = \frac{23}{60} \sim 0.38$.

Notice that there is in general no relation between the density $\delta$, the number of nodes $n$ and the number of links $m$ in a stream graph, see Figure 2.

However, the classical graph relation $\delta = \frac{2m}{n(n-1)}$ holds for a link stream $L = (T, V, E)$. Indeed, we then have $T_u = T_v = |T|$ for all $u$ and $v$, and $n = |V|$, which leads to:

$$\delta(L) = \frac{\sum_{uv \in V \otimes V} |T_{uv}|}{\sum_{uv \in V \otimes V} |T|} = \frac{2 \cdot \sum_{uv \in V \otimes V} |T_{uv}|}{n \cdot (n-1) \cdot |T|} = \frac{2 \cdot m}{n \cdot (n-1)}$$

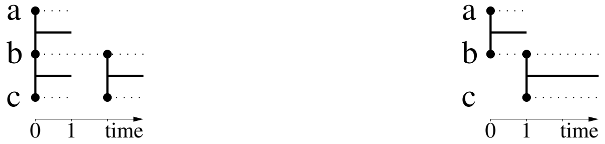


Figure 2: **Two stream graphs with $n = 2$ nodes, $m = 1$ link, but with different densities:** Left: $\delta = 0.75$. Right: $\delta = 1$.

In addition, $\delta(L)$ is equal to the average density of $G_t$: $\frac{1}{|T|}\int_t \delta(G_t)\,\mathrm{d}t = \frac{1}{|T|}\int_t \frac{|E_t|}{|V_t \otimes V_t|}\,\mathrm{d}t = \frac{1}{|T|\cdot|V\otimes V|}\int_t |E_t|\,\mathrm{d}t = \frac{\int_t |E_t|\,\mathrm{d}t}{\int_t |V_t\otimes V_t|\,\mathrm{d}t} = \delta(L)$, since, in $L$, $V_t = V$ for all $t$.

Finally, if we consider a graph-equivalent stream, then its density is equal to the density of the corresponding graph.

In addition to the global concept of density introduced above, we define the **density of a pair of nodes** $uv$ in $V \otimes V$, the **density of a node** $v$ in $V$, and the **density at a time instant** $t$ in $T$ respectively as follows:

$$\delta(uv) = \frac{|T_{uv}|}{|T_u \cap T_v|}, \quad \delta(v) = \frac{\sum_{u\in V, u\neq v} |T_{uv}|}{\sum_{u\in V, u\neq v} |T_u \cap T_v|} \quad \text{and} \quad \delta(t) = \frac{|E_t|}{|V_t \otimes V_t|}.$$

If $|T_u \cap T_v| = 0$, $\sum_{u\in V,u\neq v} |T_u \cap T_v| = 0$ or $|V_t \otimes V_t| = 0$, respectively, then we define $\delta(uv)$, $\delta(v)$ and $\delta(t)$ to be 0.

The density of $uv$ is the probability that there is a link between $u$ and $v$ whenever this is possible, *i.e.* when they are both present. The density of $v$ is the probability that a link between $v$ and any other node exists whenever this is possible, and the density of $t$ is equal to $\delta(G_t)$, the density of the graph $G_t$, *i.e.* the probability that a link exists between any two nodes present at time $t$.

For $S$ defined in Figure 1 (left), for instance, we obtain $\delta(ab) = \frac{|T_{ab}|}{|T_a \cap T_b|} = \frac{3}{9} = \frac{1}{3}$ and $\delta(bd) = \frac{|T_{bd}|}{|T_b \cap T_d|} = \frac{1}{2} = 0.5$. We also obtain $\delta(d) = \frac{|T_{da}|+|T_{db}|+|T_{dc}|}{|T_d\cap T_a|+|T_d\cap T_b|+|T_d\cap T_c|} = \frac{0+1+0}{2+2+0} = 0.25$ and $\delta(2) = \frac{|E_2|}{|V_2 \otimes V_2|} = \frac{2}{3\cdot 2/2} = \frac{2}{3}$.

Notice that $uv_t$ is strongly related to the concept of density: it is the probability that $u$ and $v$ are linked together at time $t$, which is equal to 1 or 0 depending on whether $(t, uv)$ is in $E$ or not. We then have $\delta(uv) = \frac{\int_{t\in T} uv_t\,\mathrm{d}t}{\int_{t\in T} u_t\cdot v_t\,\mathrm{d}t}$, $\delta(v) = \frac{\sum_{u\in V}\int_{t\in T} uv_t\,\mathrm{d}t}{\sum_{u\in V}\int_{t\in T} u_t\cdot v_t\,\mathrm{d}t}$, and $\delta(t) = \frac{\sum_{uv\in V\otimes V} uv_t}{\sum_{uv\in V\otimes V} u_t\cdot v_t}$. Likewise, $\delta(S) = \frac{\sum_{uv\in V\otimes V}\int_{t\in T} uv_t\,\mathrm{d}t}{\sum_{uv\in V\otimes V}\int_{t\in T} u_t\cdot v_t\,\mathrm{d}t}$.

In a link stream $L = (T, V, E)$, $T_v = T$ for all $v$ and $V_t = V$ for all $t$, and so $\delta(uv) = \frac{|T_{uv}|}{|T|} = m_{uv}$, $\delta(t) = \frac{|E_t|}{|V\otimes V|} = l_t$, and, as shown above, $\delta(L)$ is equal to the average of $\delta(t)$. In a graph-equivalent stream, $\delta(uv) \in \{0, 1\}$, and $\delta(t)$ is equal to the density of the induced graph.

The density $\delta(v)$ of node $v$ is strongly related to its degree, that we introduce in Section 8.

For brevity purposes, pages 11 to 38 of this paper are not printed in the deliverable. To get the full document, please go to:

https://arxiv.org/abs/1710.04073

$E$ and $(t, w, v) \in E\}$ and $E_\bot = \cup_{(t,v)\in W_\top}\{(t, uw)$ s.t. $(t, v, u) \in E$ and $(t, v, w) \in E\}$, respectively. In other words, in $S_\top$ two (top) nodes are linked together at a given time instant if they have (at least) a (bottom) neighbor in common in $S$ at this time, and $S_\bot$ is defined symmetrically. See Figure 22 for an illustration. Notice that, if $v \in \top$ (resp. $v \in \bot$) then $N(v)$ always is a (not necessarily maximal) clique in $S_\bot$ (resp. $S_\bot$).

Given a top node $v \in \top$ (the case of bottom nodes is symmetrical), let us denote by $S \setminus v$ the (bipartite) stream graph obtained by removing node $v$ and all its links from $S$: $S \setminus v = (T, \top \setminus \{v\}, \bot, W \setminus (T \times \{v\}), E \setminus (T \times \{v\} \times \bot))$. The redundancy $rc(v)$ of $v \in \top$ is the density of the substream of $(S \setminus v)_\bot$ induced by its neighborhood $N(v)$ in $S$. In other words, it is the fraction of its pairs of neighbors and time instants that have (at least) another neighbor in common at this time.

If $S$ is a graph-equivalent bipartite stream, then its corresponding graph also is bipartite. Moreover, the projections of $S$ are also graph-equivalent streams, and their corresponding graphs are the projections of the graph corresponding to $S$. In addition, the bipartite properties of $S$ are equivalent to the bipartite properties of its corresponding bipartite graph.

# 21 Related work

Studying interactions over time is crucial in a wide variety of contexts, leading to a huge number of papers dealing with various cases of interest. We cite for instance studies of phone calls [37, 6], contacts between individuals [2, 45, 83], cattle exchanges [20, 54], messaging [26, 23], or internet traffic [29, 82], but we could cite hundreds more. In each practical context, researchers and engineers face the challenge of analyzing the both temporal and structural nature of interactions, and they develop ad-hoc methods and tools to do so. Several surveys of these works are available from various perspectives [46, 66, 78, 68, 31, 25, 32, 19].

The most classical approach consists in splitting time into slices and then building a graph, often called snapshot, for each time slice: its nodes and links represent the interactions that occurred during this time slice. One obtains a sequence of snapshots (one for each slice), and may study the time-evolution of their properties, see for instance [65, 43, 61, 27, 7, 79], among many others. In [3], the authors even design a general framework to combine and aggregate wide classes of temporal properties, thus providing a unified approach for snapshot sequence studies. However, these approaches need time slices large enough to ensure that each snapshot captures significant information. But large slices lead to losses of temporal information, since all interactions within a same slice are merged. In addition, several or even varying slice durations may be relevant. As a consequence, choosing appropriate time slices is a research topic in itself [41, 58, 38, 63, 10]. More importantly, key concepts like paths make little sense in this framework: paths within a slice do not respect the dynamics of interactions, and paths over several time slices are difficult to handle [41].

To avoid these issues, several authors propose to encode the full information into various kinds of augmented graphs. In [12, 3, 61] for instance, authors consider the graph of all nodes and links occurring within the data, and label each node and link with its presence times. In [86, 35, 48, 73], the authors duplicate each node into as many copies as its number of occurrences (they assume discrete time steps); then, an interaction between two nodes at a given time is encoded by a link between the copies of these nodes at this time, and each copy of a node is connected to its copy at the next time step. In [88, 52] and others, the authors build reachability graphs: two nodes are linked together if they can reach each other in the stream. With such encodings, some key properties of the stream are equivalent to properties of the obtained graph, and so studying this graph sheds light on the original data. However, concepts like density or clusters make little sense on such objects, and authors then resort to the time slicing approach [61].

All these approaches have a clear advantage: once the data is transformed into one or several graphs, it is possible to use graph tools and concepts to study the interactions under concern. In the same spirit, various powerful methods for graph studies are extended to cope with the dynamics. This leads for instance to algebraic approaches for temporal network analysis [3, 57], dynamic stochastic block models [90, 47, 14, 15], dynamic Markovian models [69, 70, 67, 68], signals on temporal networks [28], adjacency tensors [72, 24], temporal networks studies with walks [71, 59, 62], dynamic graphlets [33, 29] and temporal motif counting approaches [36, 53]. Clearly, these works extend higher-level methods to the temporal setting, whereas we focus here on the most basic graph concepts, in the hope that they will form a unifying ground to such works.

Complementary to these approaches that extend methods, some works extend various graph concepts to deal with time, in a way similar to what we do here [3, 32, 51].

In particular, path-related concepts received much attention because of their importance for spreading phenomena and communication networks, see for instance [31, 88, 75, 54]. Interestingly, although paths defined in these papers are similar to those we consider here, most derived concepts remain node-oriented. For instance most authors define the centrality of a given node and connected components as sets of nodes (without time information) [3, 51, 61, 88, 52, 75]. In [16], the authors introduce a centrality for time instants. Since the centrality of nodes may greatly change over time [44], it is important to define centralities of each node at each time instant. Some authors did so for various kinds of centralities [77, 21, 73, 76, 66] but, up to our knowledge, we are the first ones to consider paths from all nodes at all time instants to all other nodes at all other time instants. This has the advantage of fully capturing the dynamics of the data, in particular the fact that nodes are not always present.

Some works go beyond path-related notions and study dynamics of node and link presence, link repetitions, instantaneous degree, and triadic closure [92, 30, 69, 13, 74, 56, 42, 49, 81, 3]. However, up to our knowledge, there exists no previous generalization of density, neighborhood, or clustering coefficient that avoids time slicing. Interestingly, a notion of degree very close to the one we propose here was introduced in the context of medical studies [80]. A notion close to average degree is introduced in [60] for dense dynamic sub-graphs searching. We also studied preliminary notions of density, cliques, quotient streams, and dense substreams in our own previous work [84, 22, 23, 83, 82].

Finally, although there is a very rich body of works on temporal networks, dynamic graphs, longitudinal networks, time-varying graphs, relational event models, etc, none of these works aims at extending the basic graph theoretic language to the situation where time and structure are equally important, like we try to do here.

# 22 Conclusion

**In this paper, we introduce a formalism to deal directly with the both temporal and structural nature of interactions over time.** We first define elementary concepts like numbers of nodes and links, density, clusters, and paths (Sections 3 to 6 and Section 14). From them, we derive more advanced concepts like cliques, neighborhoods, degrees, clustering coefficients, and connected components (Sections 7 to 10 and Section 15), and we show how to go further by introducing quotient streams, line streams, $k$-cores and centralities (Sections 11 to 13 and Section 17). Our formalism is able to cope with both discrete and continuous time (Section 18), with both instantaneous links and links with durations (Section 19), and we also consider the case where nodes have no dynamic, that we call link streams. Last but not least, our formalism may be extended to incorporate various features of the data, and we illustrate this with bipartite streams in Section 20.

The strength of our approach is to rely on very basic (but non-trivial) innovations like non-integer numbers of nodes and links, symmetric roles for time instants and nodes, a simple and intuitive concept of density, an elementary definition of clusters, and paths that connect a node at a given time to a node at a given time. **These basic concepts make it easy to define more advanced objects**: neighborhoods are clusters, degrees are fractional numbers of nodes in the neighborhoods, clustering coefficients are densities of neighborhoods, betweenness centralities are fractions of paths from any node at any time to any node at any time, etc. We demonstrate the strength of this approach by extending more advanced graph concepts such as quotient graph, trees, line graph, and $k$-cores, among others. Their definitions are mere retranscriptions of classical graph definitions into our formalism for stream graphs and link streams, and one may easily extend many other notions in this way.

In addition to this self-consistency, **our formalism is consistent with graph theory** in a very strong and precise way: if one considers a stream graph with no dynamics (nodes are present all the time, and two nodes are either linked all the time or not at all), then the stream graph is equivalent to a graph and its stream properties are equivalent to the properties of the corresponding graph. As a consequence, our formalism is a generalization of graph theory, which provides a solid ground for generalizing other graph notions.

With our formalism, one is equipped with a wide set of concepts for describing data modeled as a stream graph or a link stream. It is natural to start with the description of how elementary metrics like $k_t$ (the fraction of nodes present at time $t$) evolve over time, and of distributions of values of $n_v$ (the fraction of time at which $v$ is present) for all nodes. One may then study the instantaneous degree distribution, the degree distribution of nodes, and the time-evolution of the time degree. More advanced metrics and properties, such as connectedness, clustering coefficient or centralities, give finer insight on the data. Finally, just like graph concepts do for relations, **our formalism provides a language for describing interactions over time** in an intuitive way, both at global and more local levels. Importantly, it does not require to choose a specific time scale for conducting such studies.

**Data that would benefit from such an approach are countless,** but we believe that analysis of network traffic, mobility traces, and financial transactions are among the most promising ones, and we are working on such applications. Indeed, modeling such data with (directed, weighted) stream graphs and link streams captures most of their features, and progress in these fields is currently limited by the lack of appropriate modeling.

In order to conduct such real-world applications, it is crucial to design and implement convenient software able to efficiently compute the properties of large stream graphs. Work in this direction is in progress for the properties presented in this paper. However, it must be clear that some concepts raise serious algorithmic challenges. We worked for instance on clique and dense substream computations [84, 23], and previous work exists on various problems, see for instance [9, 11, 12, 5, 89]. In particular, the authors of [12] define a first complexity hierarchy for stream graphs. Still, most remains to be done in the design of efficient algorithms for stream graphs and the understanding of their complexity.

Another important direction is the design of models of stream graphs and link streams, which play a crucial role for simulations and proofs. In particular, an important approach in graph studies consists in generating uniformly at random graphs that have a prescribed set of properties. For instance, the Erdös-Renyi model generates graphs with prescribed size and density, while the configuration model generates graphs with prescribed size and degree distribution. The definitions we introduce in this paper (in particular for density and degree) open the way to the definition of models for generating stream graphs with prescribed properties, and to a more unified understanding of already existing models, like the ones defined in [91, 40, 67, 42, 68, 27, 34] for instance.

Last but not least, one may notice that stream graphs are not only generalizations of graphs. They actually lie at the crossroad of two very rich and powerful scientific areas: graph theory, as we have seen, and time series analysis. Indeed, if a stream graph has no dynamics then it is equivalent to a graph; if it has no structure then it is equivalent to a time series. As a consequence, we consider that a very promising direction for future work is to generalize time series concepts to stream graphs, in a way similar to what we did with graph concepts in this paper.

**Acknowledgments.** This work is funded in part by the European Commission H2020 FET-PROACT 2016-2017 program under grant 732942 (ODYCCEUS), by the ANR (French National Agency of Research) under grants ANR-15-CE38-0001 (AlgoDiv) and ANR-13-CORD-0017-01 (CODDDE), by the French program "PIA - Usages, services et contenus innovants" under grant O18062-44430 (REQUEST), and by the Ile-de-France program FUI21 under grant 16010629 (iTRAC). We warmly thank the many colleagues and friends who read preliminary versions

of this work and provided invaluable feedback.

# References

[1] Albert-László Barabási and Márton Pósfai. *Network science.* Cambridge University Press, Cambridge, 2016.

[2] Alain Barrat and Ciro Cattuto. *Temporal Networks of Face-to-Face Human Interactions*, pages 191–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[3] Vladimir Batagelj and Selena Praprotnik. An algebraic approach to temporal network analysis based on temporal quantities. *Social Netw. Analys. Mining*, 6(1):28:1–28:22, 2016.

[4] Claude Berge. *The Theory of Graphs and Its Applications.* Methuen, 1962.

[5] Sandeep Bhadra and Afonso Ferreira. Computing multicast trees in dynamic networks and the complexity of connected components in evolving graphs. *J. Internet Services and Applications*, 3(3):269–275, 2012.

[6] Vincent D. Blondel, Adeline Decuyper, and Gautier Krings. A survey of results on mobile phone datasets analysis. *EPJ Data Science*, 4(1):10, Aug 2015.

[7] Benjamin Blonder, Tina W. Wey, Anna Dornhaus, Richard James, and Andrew Sih. Temporal dynamics and network analysis. *Methods in Ecology and Evolution*, 3(6):958–972, 2012.

[8] John Adrian Bondy. *Graph Theory With Applications.* Elsevier Science Ltd., Oxford, UK, UK, 1976.

[9] Binh-Minh Bui-Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *Int. J. Found. Comput. Sci.*, 14(2):267–285, 2003.

[10] Rajmonda Sulo Caceres and Tanya Berger-Wolf. *Temporal Scale of Dynamic Networks*, pages 65–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[11] Arnaud Casteigts, Paola Flocchini, Bernard Mans, and Nicola Santoro. Shortest, fastest, and foremost broadcast in dynamic networks. *Int. J. Found. Comput. Sci.*, 26(4):499–522, 2015.

[12] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *IJPEDS*, 27(5):387–408, 2012.

[13] Vania Conan, Jérémie Leguay, and Timur Friedman. Characterizing pairwise intercontact patterns in delay tolerant networks. In *Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems*, Autonomics '07, pages 19:1–19:9, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[14] Marco Corneli, Pierre Latouche, and Fabrice Rossi. Modelling time evolving interactions in networks through a non stationary extension of stochastic block models. In Pei et al. [55], pages 1590–1591.

[15] Marco Corneli, Pierre Latouche, and Fabrice Rossi. Block modelling in dynamic networks with non-homogeneous poisson processes and exact icl. *Social Network Analysis and Mining*, 6(1):55, Aug 2016.

[16] Eduardo Chinelate Costa, Alex Borges Vieira, Klaus Wehmuth, Artur Ziviani, and Ana Paula Couto da Silva. Time centrality in dynamic complex networks. *Advances in Complex Systems*, 18(7-8), 2015.

[17] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA, 2010.

[18] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[19] Patrick Doreian and Frans Stokman. *Evolution of Social Networks*. 1997.

[20] Bhagat Lal Dutta, Pauline Ezanno, and Elisabeta Vergu. Characteristics of the spatio-temporal network of cattle movements in France over a 5-year period. *Preventive Veterinary Medicine*, 117(1):79–94, 2014.

[21] Julio Flores and Miguel Romance. On eigenvector-like centralities for temporal networks: Discrete vs. continuous time scales. *Journal of Computational and Applied Mathematics*, 2017.

[22] Noé Gaumont, Clémence Magnien, and Matthieu Latapy. Finding remarkably dense sequences of contacts in link streams. *Social Netw. Analys. Mining*, 6(1):87:1–87:14, 2016.

[23] Noé Gaumont, Tiphaine Viard, Raphaël Fournier-S'niehotta, Qinna Wang, and Matthieu Latapy. *Analysis of the Temporal and Structural Features of Threads in a Mailing-List*, pages 107–118. Springer International Publishing, Cham, 2016.

[24] Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach. *PLOS ONE*, 9(1):1–13, 01 2014.

[25] Betsy George and Sangho Kim. *Spatio-temporal Networks: Modeling and Algorithms*. 2013.

[26] Luiz H. Gomes, Virgilio A. F. Almeida, Jussara M. Almeida, Fernando D. O. Castro, and Luís. A. Bettencourt. Quantifying social and opportunistic behavior in email networks. *Advances in Complex Systems*, 12(01):99–112, 2009.

[27] László Gulyás, George Kampis, and Richard O. Legendi. Elementary models of dynamic networks. *The European Physical Journal Special Topics*, 222(6):1311–1333, Sep 2013.

[28] Ronan Hamon, Pierre Borgnat, Patrick Flandrin, and Céline Robardet. Duality between temporal networks and signals: Extraction of the temporal network structures. *CoRR*, abs/1505.03044, 2015.

[29] Christopher R. Harshaw, Robert A. Bridges, Michael D. Iannacone, Joel W. Reed, and John R. Goodall. Graphprints: Towards a graph analytic method for network anomaly detection. In *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, CISRC '16, pages 15:1–15:4, New York, NY, USA, 2016. ACM.

[30] Enrique Hernández-Orallo, Juan Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. New approaches for characterizing inter-contact times in opportunistic networks. *Ad Hoc Networks*, 52:160 – 172, 2016. Modeling and Performance Evaluation of Wireless Ad Hoc Networks.

[31] Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):234, Sep 2015.

[32] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519(3):97 – 125, 2012. Temporal Networks.

[33] Yuriy Hulovatyy, Huili Chen, and Tijana Milenkovic. Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*, 32(15):2402, 2016.

[34] M. Karsai, M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki. Small but slow world: How network topology and burstiness slow down spreading. *Phys. Rev. E*, 83:025102, Feb 2011.

[35] Vassilis Kostakos. Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007 – 1023, 2009.

[36] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(11):P11005, 2011.

[37] Lauri Kovanen, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs reveal homophily, gender-specific patterns and grouptalk in mobile communication networks. 02 2013.

[38] Gautier Krings, Márton Karsai, Sebastian Bernhardsson, Vincent D. Blondel, and Jari Saramäki. Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Science*, 1(1):4, May 2012.

[39] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio. Basic notions for the analysis of large two-mode networks. *Social Networks*, 30(1):31–48, 2008.

[40] Guillaume Laurent, Jari Saramäki, and Márton Karsai. From calls to communities: a model for time-varying social networks. *The European Physical Journal B*, 88(11):301, Nov 2015.

[41] Yannick Léo, Christophe Crespelle, and Eric Fleury. Non-altering time scales for aggregation of dynamic networks into series of graphs. In Felipe Huici and Giuseppe Bianchi, editors, *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT 2015, Heidelberg, Germany, December 1-4, 2015*, pages 29:1–29:7. ACM, 2015.

[42] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 462–470. ACM, 2008.

[43] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1):2, 2007.

[44] Clémence Magnien and Fabien Tarissan. Time evolution of the importance of nodes in dynamic networks. In Pei et al. [55], pages 1200–1207.

[45] Lucie Martinet, Christophe Crespelle, and Eric Fleury. *Dynamic Contact Network Analysis in Hospital Wards*, pages 241–249. Springer International Publishing, Cham, 2014.

[46] Naoki Masuda and Renaud Lambiotte. *A Guide to Temporal Networks*, volume 4 of *Series on Complexity Science.* 2016.

[47] Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages n/a–n/a, 2016.

[48] Othon Michail. *An Introduction to Temporal Graphs: An Algorithmic Perspective*, pages 308–343. Springer International Publishing, Cham, 2015.

# An Information-theoretic Framework for the Lossy Compression of Link Streams

Robin Lamarche-Perrin

*Centre national de la recherche scientifique*
*Institut des systèmes complexes de Paris Île-de-France*
*Laboratoire d'informatique de Paris 6*

## Abstract

Graph compression is a data analysis technique that consists in the replacement of parts of a graph by more general structural patterns in order to reduce its description length. It notably provides interesting exploration tools for the study of real, large-scale, and complex graphs which cannot be grasped at first glance. This article proposes a framework for the compression of temporal graphs, that is for the compression of graphs that evolve with time. This framework first builds on a simple and limited scheme, exploiting structural equivalence for the lossless compression of static graphs, then generalises it to the lossy compression of link streams, a recent formalism for the study of temporal graphs. Such generalisation relies on the natural extension of (bidimensional) relational data by the addition of a third temporal dimension. Moreover, we introduce an information-theoretic measure to quantify and to control the information that is lost during compression, as well as an algebraic characterisation of the space of possible compression patterns to enhance the expressiveness of the initial compression scheme. These contributions lead to the definition of a combinatorial optimisation problem, that is the Lossy Multistream Compression Problem, for which we provide an exact algorithm.

*Keywords:* Graph compression, link streams, structural equivalence, information theory, combinatorial optimisation.

## Table of Definitions

*Email address:* Robin.Lamarche-Perrin@lip6.fr (Robin Lamarche-Perrin)
*URL:* https://www-complexnetworks.lip6.fr/~lamarche/ (Robin Lamarche-Perrin)

*Preprint submitted to Theoretical Computer Science* *July 19, 2018*

**Contents**

**Table of Notations**

| | |
|---|---|
| $v \in \boldsymbol{V}$ / $t \in \boldsymbol{T}$ | a vertex / a time instance |
| $V \in \mathcal{P}(\boldsymbol{V})$ / $T \in \mathcal{P}(\boldsymbol{T})$ | a vertex subset / a time subset |
| $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ / $\mathcal{T} \in \mathfrak{P}(\boldsymbol{T})$ | a vertex partition / a time partition |
| $\mathcal{V}(v) \in \mathcal{V}$ / $\mathcal{T}(t) \in \mathcal{T}$ | the vertex subset in $\mathcal{V}$ that contains $v$ / the time instance in $\mathcal{T}$ that contains $t$ |
| $(v, v', t) \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$ | a multiedge |
| $V\times V'\times T \in \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | a Cartesian multiedge subset |
| $\mathcal{V}\times\mathcal{V}\times\mathcal{T} \in \mathfrak{P}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | a grid multiedge partition |
| $\mathcal{V}\mathcal{V}\mathcal{T} \in \mathfrak{P}^{\times}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | a Cartesian multiedge partition |
| $\mathcal{V}\mathcal{V}\mathcal{T}(v, v', t) \in \mathcal{V}\mathcal{V}\mathcal{T}$ | the multiedge subset in partition $\mathcal{V}\mathcal{V}\mathcal{T}$ that contains $(v, v', t)$ |
| $e : \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T} \to \mathbb{N}$ | the edge function of a multistream |
| $e : \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{T}) \to \mathbb{N}$ | the additive extension of the edge function |
| $e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})$ | the total number of edges |

| | |
|---|---|
| $(X, X', X'') \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$ | the observed variable associated with the empirical distribution of edges in a multistream |
| $\mathcal{V}\times\mathcal{V}\times\mathcal{T}(X, X', X'') \in \mathcal{V}\times\mathcal{V}\times\mathcal{T}$ | the compressed variable resulting from the compression of the observed variable $(X, X', X'')$ by a given multiedge partition $\mathcal{V}\times\mathcal{V}\times\mathcal{T}$ |
| $(Y, Y', Y'') \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$ | the external variable which distribution is used to decompress the compressed variable $\mathcal{V}\times\mathcal{V}\times\mathcal{T}(X, X', X'')$ |
| $\mathcal{V}\times\mathcal{V}\times\mathcal{T}_{(Y,Y',Y'')}(X, X', X'') \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$ | the decompressed variable obtained from the decompression of the compressed variable $\mathcal{V}\times\mathcal{V}\times\mathcal{T}(X, X', X'')$ according to the external variable $(Y, Y', Y'')$ |
| $\mathrm{loss}(\mathcal{V}\times\mathcal{V}\times\mathcal{T})$ | the information loss induced from the compression of the observed variable $(X, X', X'')$ by a given multiedge partition $\mathcal{V}\times\mathcal{V}\times\mathcal{T}$ and its decompression according to the external variable $(Y, Y', Y'')$ |

| | |
|---|---|
| $V \in \hat{\mathcal{P}}(\boldsymbol{V})$ / $T \in \hat{\mathcal{P}}(\boldsymbol{T})$ | a feasible vertex subset / a feasible time subset |
| $\mathcal{H}(\boldsymbol{V})$ / $\mathcal{I}(\boldsymbol{T})$ | a vertex hierarchy / a set of time intervals |
| $V\times V'\times T \in \hat{\mathcal{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | a feasible Cartesian multiedge subset |
| $\mathcal{V}\mathcal{V}\mathcal{T} \in \hat{\mathfrak{P}}^{\times}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | a feasible Cartesian multiedge partition |
| $\hat{\mathfrak{R}}(\mathcal{V}\mathcal{V}\mathcal{T}) \subset \hat{\mathfrak{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | the set of feasible multiedge partitions that refine $\mathcal{V}\mathcal{V}\mathcal{T}$ |
| $\hat{\mathfrak{C}}(\mathcal{V}\mathcal{V}\mathcal{T}) \subset \hat{\mathfrak{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$ | the set of feasible multiedge partitions that are covered by $\mathcal{V}\mathcal{V}\mathcal{T}$ |

## 1. Introduction

*Graph abstraction* is a data analysis technique aiming at the extraction of salient features from relational data to provide a simpler, and hence more useful representation of the graph under study. Such a process generally relies on a controlled information reduction suppressing redundancies or irrelevant parts of the data [1]. Abstraction techniques are hence crucial to the study of real, large-scale, and complex graphs which cannot be grasped at first glance. First, they provide tools for an optimised storage and data treatment by reducing memory requirements and running times of analysis algorithms. Second, and more importantly, they constitute valuable exploration tools for domain experts who are looking for preliminary macroscopic insights about their graphs' topology or, even better, a multiscale representation of their data.

Among abstraction techniques, *graph compression* [2, 3], also known as *graph simplification* or *graph summarisation* [4, 5, 6], consists in replacing parts of the graph by more general structural patterns in order to reduce its description length. For example, one "can replace a dense cluster by a single node, so the overall structure of the network becomes clearer" [1], or more generally replace any frequent subgraph pattern (*e.g.*, cliques, stars, loops) by a label of that pattern. Such techniques hence range from those building on collections of domain-specific patterns, such as *graph rewriting* techniques in which patterns of interest are specified according to expert knowledge [7], to those relying on more generic patterns, such as *power graph* techniques in which any group of vertices with identical interaction profiles is a candidate for summarisation [8, 9]. Because they provide more general approaches to graph analysis, we will focus on the latter.

In this article, we are more particularly interested in the compression of *temporal graphs*, that is the compression of graphs that evolve with time. Many research studies are indeed interested in the dynamics of relations, as for example the evolution of friendship relations in social sciences, or even in the dynamics of interaction events [10], as for example contact or communication networks, such as mail exchanges, financial transactions, physical meetings, and so on. Having to deal with an additional dimension – that is the temporal dimension – challenges compression techniques that have initially been developed for the study of static graphs. A traditional approach to generalise such techniques preliminary consists in the construction of a sequence of static graphs, by slicing the temporal dimension into distinct periods of interest, then in independently applying classical compression schemes to each graph of this sequence. However, such a process introduces an asymmetry in the way structural and temporal information is handled, the latter being compressed prior to – and independently from – the former.

To this extent, recent work on the *link stream* formalism proposes to deal with time as a simple addition to the graph's structural dimensions [11, 12]. Considering temporal graphs and interaction networks as genuine tridimensional data, the arbitrary separation of structure and time is therefore prohibited. Following this line of thinking, the compression scheme we present in this article aims at the natural generalisation of the bidimensional compression of static graphs to the tridimensional compression of link streams, thus participating in the development of this emerging framework. Similar generalisation objectives have been addressed in previous work on graph compression, as for example the application of bidimensional *block models* to multidimensional matrices [13] or the application of *biclustering* to triplets of variables [14], which has then been exploited for the statistical analysis of temporal graphs [15]. The particular interest of such approaches also consists in the fact that they provide a unified compression scheme in which structural and temporal information is simultaneously taken into account.

In order to present our compression framework, this article starts canonical and specific, then increase in generality and in sophistication. Section 2 introduces the *Graph Compression Problem* (GCP), a first compression scheme that relies on a most classical combinatorial problem in graph theory: Finding classes of structurally-equivalent vertices [16] to summarise the adjacency-list and the adjacency-matrix representations of a given graph. This approach to graph compression is canonical in the sense that it only builds on the primary, first-order information that is contained in relational data, that is the information encoded in vertex adjacency. It is also specific in the sense that it only applies to simple graphs (that is graphs for which at most one edge is allowed between two vertices) with no temporal dimension (that is static graphs). Moreover, this first scheme is lossless (it does not allow for any information loss during compression) and its solution space is both strongly constrained (only vertex partitions are considered, whereas edge partitions would allow for much more compression choices) and weakly expressive (any vertex subset is feasible, whereas interesting structural properties preliminarily defined by the expert domain might need to be preserved during compression).

In order to address such limitations, Section 3 consists in a step-by-step generalisation of the GCP to make it suitable for the lossy compression of temporal graphs. First, we show how to deal with the compression of *multigraphs* (that is graphs for which multiple edges are allowed between two vertices) by generalising the notion of structural equivalence to the case of multiple edges (3.1). Second, we allow for a *lossy* compression scheme by formalising a proper measure of information loss building on the entropy of the adjacency information contained in the compressed graph relative to the one contained in the initial graph (3.2). Third, we allow for a less constrained compression scheme by generalising from vertex partitions to edge partitions (3.3). Fourth, we allow for a more expressive scheme by driving compression according to a predefined set of feasible aggregates (3.4). Fifth and last, we generalise the resulting framework to the compression of temporal multigraphs, that is what we later call *multistreams*, by adding a temporal dimension to the compression scheme (3.5). These five contributions finally define a general and flexible scheme for link stream compression, that we call the *Multistream Compression Problem* (MSCP).

Section 4 then presents a combinatorial optimisation algorithm to solve the MSCP. It relies on the reduction of the problem to the better-known *Set Partitioning Problem* (SPP) arising as soon as one wants to organise a set of objects into covering and pairwise disjoint subsets such that an additive objective is minimised [17]. Building on a generic algorithmic framework proposed in previous work to solve special versions of the SPP [18, 19], this article derives an algorithm to the particular case of the MSCP. This algorithm relies on the acknowledgement of a principle of optimality, showing that the problem's solution space has an optimal substructure allowing for the recursive combination of locally-optimal solutions. Applying classical methods of dynamic programming and providing a proper data structure for the MSCP, we finally derive an exact algorithm which is exponential in the worst case, but polynomial when the set of feasible vertex aggregates is assumed to have some particular structure (*e.g.*, hierarchies of vertices and sets of intervals).

Section 5 discusses the outcomes of this new compression scheme and provides some research perspectives, notably to propose in the future tractable approximation algorithms for the lossy compression of large-scale temporal graphs.

## 2. Starting Point: The Lossless Graph Compression Problem

The starting point to build our compression scheme is a well-known combinatorial problem: Find the quotient set of the structural equivalence relation applying to the vertices of a graph. As the resulting equivalence classes form a partition of the vertex set by grouping together vertices with an identical (first-order) structure – that is with identical neighbourhoods – one can exploit such classes to compress the graph representation, as illustrated in Figure 1. Structural equivalence can thus be used for the lossless compression of static graphs, and we later list the improvements one needs in order to generalise this first simple scheme to the lossy compression of link streams.

### *2.1. Preliminary Notations*

Given a set of vertices $\boldsymbol{V} = \{v_1, \dots, v_n\}$, we mark:

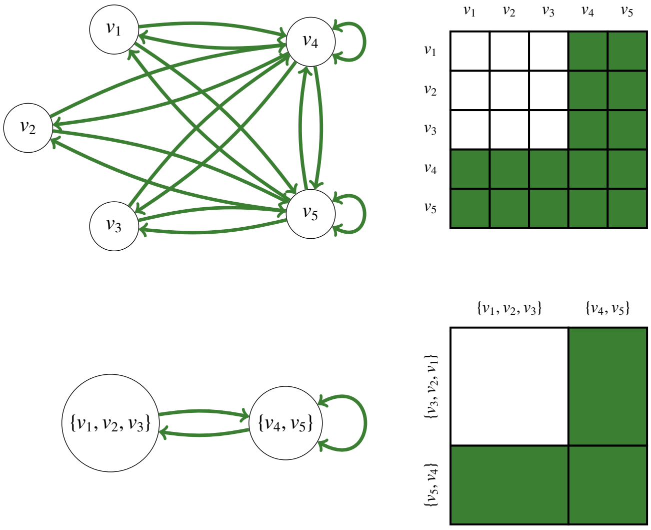
- $\mathcal{P}(\boldsymbol{V})$ the set of all vertex subsets: $\mathcal{P}(\boldsymbol{V}) = \{V \subseteq \boldsymbol{V}\}$;
- $\mathfrak{P}(\boldsymbol{V})$ the set of all vertex partitions:

$$\mathfrak{P}(\boldsymbol{V}) \;=\; \{\{V_1, \dots, V_m\} \subseteq \mathcal{P}(\boldsymbol{V}) \;:\; \cup_i V_i = \boldsymbol{V} \;\wedge\; \forall i \neq j,\; V_i \cap V_j = \emptyset\};$$

- Given a vertex $v \in \boldsymbol{V}$ and a vertex partition $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$, we mark $\mathcal{V}(v)$ the unique vertex subset in $\mathcal{V}$ that contains $v$.

More generally, this article uses a consistent system of capitalization and typefaces to properly formalise the compression problem and its solution space:

- Vertices are designated by lowercase letters: $v$, $v'$, $u$, $u'$;
- Vertex sets and vertex subsets by uppercase letters: $\boldsymbol{V}$, $V$, $V'$;
- Vertex partitions and sets of vertex subsets by calligraphic letters: $\mathcal{V}$, $\mathcal{V}'$, $\mathcal{P}(\boldsymbol{V})$, $\mathcal{H}(\boldsymbol{V})$, $\mathcal{I}(\boldsymbol{V})$;
- Sets of vertex partitions by Gothic letters: $\mathfrak{P}(\boldsymbol{V})$, $\mathfrak{H}(\boldsymbol{V})$, $\mathfrak{I}(\boldsymbol{V})$.



|  | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $v_1$ |  |  |  | 🟩 | 🟩 |
| $v_2$ |  |  |  | 🟩 | 🟩 |
| $v_3$ |  |  |  | 🟩 | 🟩 |
| $v_4$ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| $v_5$ | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |

|  | $\{v_1, v_2, v_3\}$ | $\{v_4, v_5\}$ |
|---|---|---|
| $\{v_3, v_2, v_1\}$ |  | 🟩 |
| $\{v_5, v_4\}$ | 🟩 | 🟩 |

Figure 1: Lossless compression of a 5-vertex, 16-edge graph (above) into a 2-vertex, 3-edge graph (below). The *adjacency-list* representation is given on the left and the *adjacency-matrix* representation on the right.

### *2.2. The Lossless GCP*

To begin with, we consider a simple case: Directed static graphs, with possible self-loops on the vertices.

**Definition 1** (Directed Graph)**.**
*A* directed graph $G = (\boldsymbol{V}, \boldsymbol{E})$ *is characterised by:*

- *A set of* vertices $\boldsymbol{V}$*;*
- *A set of* directed edges $\boldsymbol{E} \subseteq \boldsymbol{V}\times\boldsymbol{V}$.

*For all vertex* $v \in \boldsymbol{V}$*, we respectively mark* $N_{in}(v) = \{v' \in \boldsymbol{V} : (v', v) \in \boldsymbol{E}\}$ *and* $N_{out}(v) = \{v' \in \boldsymbol{V} : (v, v') \in \boldsymbol{E}\}$ *the* in-coming *and the* out-going neighbourhoods *of* $v$*.*

The upper part of Figure 1 gives an example of directed graph made of $|\boldsymbol{V}| = 5$ vertices and $|\boldsymbol{E}| = 16$ edges. It is represented in the form of *adjacency lists* (on the left), where each edge is represented as an arrow going from a source vertex $v \in \boldsymbol{V}$ to a target vertex $v' \in \boldsymbol{V}$, as well as in the form of an *adjacency matrix* (on the right), where edges are represented within a binary matrix of size $|\boldsymbol{V}| \times |\boldsymbol{V}|$.

The combinatorial problem we now formalise builds on the classical relation of *structural equivalence* applying to the vertex set of a graph [16].

**Definition 2** (Structural Equivalence)**.**
*The* structural equivalence relation $\sim\, \subseteq \boldsymbol{V}^2$ *is defined on directed graphs by the equality of neighbourhoods: Two vertices* $(v, v') \in \boldsymbol{V}^2$ *are* structurally equivalent *if and only if they are connected to the same vertices. Formally:*

$$v \sim v' \quad \Leftrightarrow \quad N_{in}(v) = N_{in}(v') \quad \textit{and} \quad N_{out}(v) = N_{out}(v').$$

*A vertex subset* $V \in \mathcal{P}(\boldsymbol{V})$ *is* structurally consistent *if and only if all its vertices are structurally equivalent with each others, and a vertex partition* $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ *is* structurally consistent *if and only if all its vertex subsets are structurally consistent. We respectively mark* $\widetilde{\mathcal{P}}(\boldsymbol{V})$ *and* $\widetilde{\mathfrak{P}}(\boldsymbol{V})$ *the sets of structurally-consistent vertex subsets and vertex partitions:*

$$V \in \widetilde{\mathcal{P}}(\boldsymbol{V}) \quad \Leftrightarrow \quad \forall (v, v') \in V^2, \quad v \sim v'.$$

$$\mathcal{V} \in \widetilde{\mathfrak{P}}(\boldsymbol{V}) \quad \Leftrightarrow \quad \forall V \in \mathcal{V}, \quad V \in \widetilde{\mathcal{P}}(\boldsymbol{V}).$$

The lower part of Figure 1 uses the fact that $v_1 \sim v_2 \sim v_3$ and that $v_4 \sim v_5$ to define a structurally-consistent vertex partition $\mathcal{V} = \{V_1, V_2\}$ made of two structurally-consistent vertex subsets $V_1 = \{v_1, v_2, v_3\}$ and $V_2 = \{v_4, v_5\}$.

Because all vertices belonging to a structurally-consistent vertex subset have the exact same neighbourhoods, one can use this structural redundancy to simplify the graph representation. Such a compression first consists in aggregating all vertices in structurally-consistent subsets to form *compressed vertices*, then in aggregating all edges between couples of structurally-consistent subsets to form *compressed edges*. The resulting *compressed graph* provides a smaller, yet complete description of the initial one.

**Definition 3** (Compressed Directed Graph)**.**
*Given a directed graph $G = (\boldsymbol{V}, \boldsymbol{E})$ and a structurally-consistent vertex partition $\mathcal{V} \in \widetilde{\mathfrak{P}}(\boldsymbol{V})$, the* compressed directed graph $\mathcal{V}(G) = (\mathcal{V}, \mathcal{E})$ *is the graph such that:*

- *$\mathcal{V}$ is the set of* (compressed) vertices*;*
- *$\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the set of* (compressed) directed edges *such that:*

$$\forall (V, V') \in \mathcal{V}\times\mathcal{V}, \quad (V, V') \in \mathcal{E} \quad \begin{array}{cl} \Leftrightarrow & \forall v \in V,\ \forall v' \in V',\ (v, v') \in \boldsymbol{E} \\ \Leftrightarrow & \exists v \in V,\ \exists v' \in V',\ (v, v') \in \boldsymbol{E}. \end{array}$$

*Note that both conditions are equivalent since V and V′ are structurally consistent.*

The lower part of Figure 1 shows the effect of such a compression on the graph's representations. Regarding adjacency lists, no more than one compressed edge is encoded between two given compressed vertices. Regarding the adjacency matrix, cells are merged into "rectangular tiles" containing only one binary value for each couple of compressed vertices.

These definitions lead to a well-known combinatorial problem that we call here the *Lossless Graph Compression Problem* (Lossless GCP). It simply consists in finding the quotient set of the structural equivalence relation, that is the smallest structurally-consistent partition of $\boldsymbol{V}$.

**Definition 4** (The Lossless Graph Compression Problem)**.**
*Given a directed graph $G = (\boldsymbol{V}, \boldsymbol{E})$, find a structurally-consistent vertex partition $\mathcal{V}^* \in \widetilde{\mathfrak{P}}(\boldsymbol{V})$ with minimal size $|\mathcal{V}^*|$:*

$$\mathcal{V}^* \;=\; \underset{\mathcal{V}\in\widetilde{\mathfrak{P}}(\boldsymbol{V})}{\arg\min}\ |\mathcal{V}|.$$

In Figure 1, the represented structurally-consistent vertex partition is the smallest: One cannot find such another partition that contains fewer vertex subsets. This is hence the most optimal lossless compression of the graph.

### 2.3. Related Problems

Note that *structural* equivalence is the stricter form of vertex equivalence one might consider for graph analysis [20, 21]. Yet, other equivalence relations are traditionally used in the literature in social sciences for the detection of other kinds of structural patterns, such as *automorphic* equivalence (two vertices are equivalent if there is an isomorphic graph such that these vertices are interchanged) and *regular* equivalence [13] (two vertices are equivalent if they are equally related to other equivalent classes). Because these two latter equivalence relations are less strict, they induce smaller vertex partitions with bigger classes. But more importantly, and contrary to structural equivalence, the resulting compression scheme is not reversible in the sense that one cannot find back the initial graph from the equivalent classes and their compressed edges.

Structural equivalence, and so the GCP, is also related to *community detection* [22], also known as *graph clustering*, that is a classical problem for graph analysis which consists in finding groups of vertices that are strongly connected with each others while being loosely connected to other groups. However, dense and isolated clusters are only particular examples of structurally-consistent classes. They correspond to dense diagonal blocks in the adjacency matrix. The notion of structural equivalence is more generally interested in groups of vertices with similar relational patterns, that is in any block of equal-density within the adjacency matrix (not necessarily dense and not necessarily on the diagonal, as in other work focusing on *block compression* [6, 2, 3]). Hence, the GCP is more strongly related to the family of *edge compression techniques* [8] such as *modular decomposition*[1] [3], *matching neighbours*, and *power graph analysis* [9]. In the latter, one is searching for groups of vertices that have similar relation patterns of any sort. Because it is more generally interested in the compression of equal-density blocks, the GCP can lastly be seen as a strict instance of *block modelling* [16, 20, 13, 15], another classical method of network analysis that relies on structural equivalence to discover roles and positions in social networks.

### *2.4. Possible Generalisations*

This first formulation of the GCP is restricted to static simple graphs. Moreover, it only allows lossless compression, that is compression of vertices with *identical* neighbourhoods, which is a quite stringent and unrealistic condition for empirical research. In what follows, we list the requirements to formulate a more general and more flexible optimisation problem allowing for the lossy compression of temporal graphs.

**From simple graphs to multigraphs (see 3.1).** This first version of the GCP is restricted to *simple graphs* (no more than one edge between two given vertices). Yet, it is easily generalisable to *multigraphs* (multiple edges are allowed between two given vertices). Such a generalisation has two advantages. First, multigraphs are strictly more general than simple graphs since simple graphs can be considered as a particular cases of multigraphs. Second, multigraphs are more consistent with the lossy compression scheme later presented since the end result of lossy compression is not necessarily a simple graph (as edges are aggregated into multiedges during compression).

**From lossless to lossy compression (see 3.2).** This first version of the GCP is *lossless* in the sense that the result of compression contains all the information that is required to errorlessly build back the initial graph. However, such a lossless compression – relying on *exact* equivalence – is quite inefficient in the case of real graphs within which *identical* neighbourhoods are quite unlikely. One hence needs a measure of *information loss* to allow for a more flexible compression scheme.

**From vertex to edge partitions (see 3.3).** This first version of the GCP consists in finding an interesting *vertex partition* to compress the graph, thus inducing a partition of its edges. This relates to classical approaches such as *modular decomposition* where subsets of vertices (modules) that have similar neighbourhoods are exploited to compressed the graph's structure. However, this can be generalised to the direct search for *edge partitions*, that is the search for interesting edge subsets that do not all necessarily rest on similar vertex subsets. This relates to less known approaches such as *power-graph decomposition* that allows for a more subtle analysis of the graph's structure.

**Adding constraints to the set of feasible vertex subsets (see 3.4).** In this first version of the GCP, one considers any possible vertex subset as a potential candidate for compression, thus leading to an *unconstrained* compression scheme. However, in order to represent and to preserve *additional constraints* that might apply on the vertex structure, one might want to only consider "feasible" vertex subsets when searching for an optimal partition. This requires to integrate such additional constraints within the compression scheme.

**From static graphs to link streams (see 3.5).** Our last generalisation step consists in integrating a temporal dimension within the optimisation problem in order to deal with the compression of *link streams*. The structural equivalence relation hence needs to be redefined with respect to this additional dimension and equivalent classes will then be only valid on given time intervals. In this context, one is hence searching for aggregates that partition the Cartesian product of the vertex set and of the temporal dimension.

[1] Not to be confused with modularity-based clustering, which is a form of community detection.

## 3. Generalisation: From Lossless Static Graphs to Lossy Mutlistreams

### 3.1. From Graphs to Multigraphs

Most approaches in the domain of graph theory focus on the analysis of *simple graphs*, that is graphs for which at most one edge is allowed between two vertices, thus represented as binary adjacency matrices. This is also the case when it comes to the field of graph compression (see for example [13, 4, 5, 6, 8]). Yet, in the scope of this article, we aim at the compression of *multigraphs*, that is graphs for which multiple edges are allowed between two vertices, thus represented as integer adjacency matrices. As simple graphs are special cases of multigraphs, the resulting approach is necessarily more general.

In some articles on graph compression, the generalisation to multigraphs would be quite straightforward as the result of compression – that is the compressed graph – already is, in fact, a multigraph (see for example [6]). Even if not explicitly formalised, research perspectives in that direction are sometimes provided [9]. Yet, other approaches natively deals with multigraph compression by directly taking into account, within the compression scheme, the presence of multiple edges [2, 3]. Statistical methods for *variable co-clustering* also offers compression frameworks that are designed for numerical (non-binary) matrices [23, 15, 14]. This is the approach we choose here by directly working with multigraphs.

**Definition 5** (Directed Multigraph)**.**
*A* directed multigraph $MG = (\boldsymbol{V}, e)$ *is characterised by:*

- *A set of* vertices $\boldsymbol{V}$*;*
- *A multiset of* directed edges $(\boldsymbol{V}{\times}\boldsymbol{V}, e)$
  *where* $e : \boldsymbol{V}{\times}\boldsymbol{V} \to \mathbb{N}$ *is the* edge function*, that is the multiplicity function counting the number of edges* $e(v, v') \in \mathbb{N}$ *going from a given source vertex* $v \in \boldsymbol{V}$ *to a given target vertex* $v' \in \boldsymbol{V}$*.*

  *We also define the additive extension of the edge function on couples of vertex subsets:*

  $$e : \mathcal{P}(\boldsymbol{V}){\times}\mathcal{P}(\boldsymbol{V}) \to \mathbb{N} \quad \textit{such that} \quad e(V, V') \;=\; \sum_{(v,v')\in V\times V'} e(v, v').$$

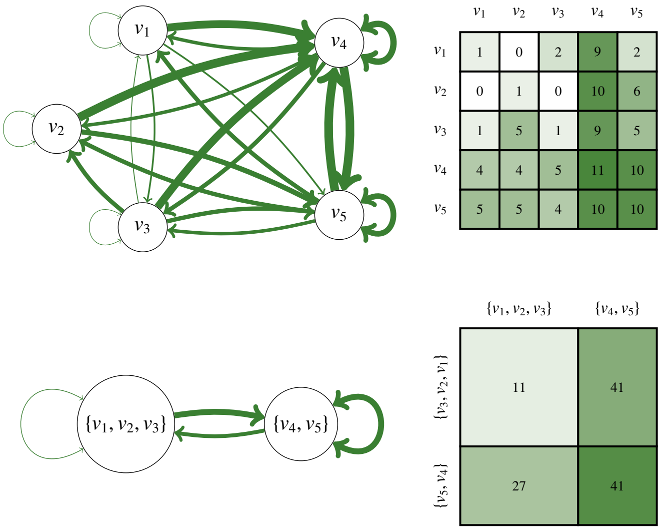
  *It simply counts the number of edges going from any vertex of a given source subset* $V \in \mathcal{P}(\boldsymbol{V})$ *to any vertex of a given target subset* $V' \in \mathcal{P}(\boldsymbol{V})$*. In particular,* $e(\boldsymbol{V}, \boldsymbol{V})$ *is the total number of edges in the multigraph,* $e(\boldsymbol{V}, v)$ *is the in-coming degree of* $v$ *and* $e(v, \boldsymbol{V})$ *its out-going degree.*

The upper part of Figure 2 gives an example of directed multigraph made of $|\boldsymbol{V}| = 5$ vertices, that is $|\boldsymbol{V}{\times}\boldsymbol{V}| = 25$ multiedges, and $e(\boldsymbol{V}, \boldsymbol{V}) = 40$ edges distributed within $\boldsymbol{V}{\times}\boldsymbol{V}$. It is represented in the form of *adjacency lists* (on the left), where each multiedge is represented as an arrow which width is proportional to the number of edges $e(v, v')$ going from a source vertex $v$ to a target vertex $v'$, as well as in the form of an *adjacency matrix* (on the right), where the edge function is represented as an integer matrix of size $|\boldsymbol{V}|{\times}|\boldsymbol{V}|$.

Structural equivalence could then be generalised to multigraphs in order to define, as done previously for simple graphs, a Lossless Multigraph Compression Problem (MGCP). In few words, the structural equivalence relation would be defined on directed multigraphs by the equality of the edge function: Two vertices are hence structurally equivalent if and only if they are each connected the same number of times to the different graph's vertices. However, as we are interested in this article in lossy compression, we directly consider an alternative version of the MGCP that relies on a stochastic relaxation of the structural equivalence relation and on an appropriate measure of information loss.

### 3.2. From Lossless to Lossy Compression

Information-theoretic compression first requires a stochastic model of the data, that is a model of the multigraph to be compressed. The measure of information loss that we present in this subsection has been previously introduced for the aggregation of geographical data [24] and for the summarisation of execution traces of distributed systems [19]. The first contribution of this article in this regard is the application of this measure to graph compression. Moreover, the underlying stochastic models were not made explicit in previous work. Our second contribution is hence the



|  | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $v_1$ | 1 | 0 | 2 | 9 | 2 |
| $v_2$ | 0 | 1 | 0 | 10 | 6 |
| $v_3$ | 1 | 5 | 1 | 9 | 5 |
| $v_4$ | 4 | 4 | 5 | 11 | 10 |
| $v_5$ | 5 | 5 | 4 | 10 | 10 |

|  | $\{v_1, v_2, v_3\}$ | $\{v_4, v_5\}$ |
|---|---|---|
| $\{v_3, v_2, v_1\}$ | 11 | 41 |
| $\{v_5, v_4\}$ | 27 | 41 |

Figure 2: Lossy compression of a 5-vertex, 40-edge multigraph (above) into a 2-vertex, 40-edge multigraph (below). In the *adjacency-list representation* (on the left), the width of arrows is proportional to the edge function, that is to the number of edges going from a source vertex to a target vertex.

thorough formalisation of the graph model we use, in order to properly justify and interpret the resulting measure. In few words, we model a multigraph as a set of edges that are stochastically distributed within the two-dimensional set of multiedges $\boldsymbol{V}\times\boldsymbol{V}$: Each edge has a particular location within this space, characterised by its source vertex and its target vertex. The edge function $e : \boldsymbol{V}\times\boldsymbol{V} \to \mathbb{N}$ hence characterises the *empirical distribution* of the edges within the multigraph, thus allowing to model the data as a discrete random variable $(X, X')$ taking on $\boldsymbol{V}\times\boldsymbol{V}$.

**Definition 6** (Observed Variable)**.**
*The* observed variable $(X, X') \in \boldsymbol{V}\times\boldsymbol{V}$ *associated with a multigraph* $MG = (\boldsymbol{V}, e)$ *is a couple of discrete random variables having the empirical distribution of edges in* $MG$*:*

$$\Pr((X, X') = (v, v')) \quad = \quad \frac{e(v, v')}{e(\boldsymbol{V}, \boldsymbol{V})} \quad \overset{\text{def}}{=} \quad p_{(X,X')}(v, v').$$

In other terms, $p_{(X,X')}(v, v')$ represents the probability that, if one chooses an edge at random among the $e(\boldsymbol{V}, \boldsymbol{V})$ edges of the multigraph, it will go from the source vertex $v$ to the target vertex $v'$. For example, matrix (i) in Figure 3 represents the distribution of the observed variable associated with the multigraph of Figure 2.

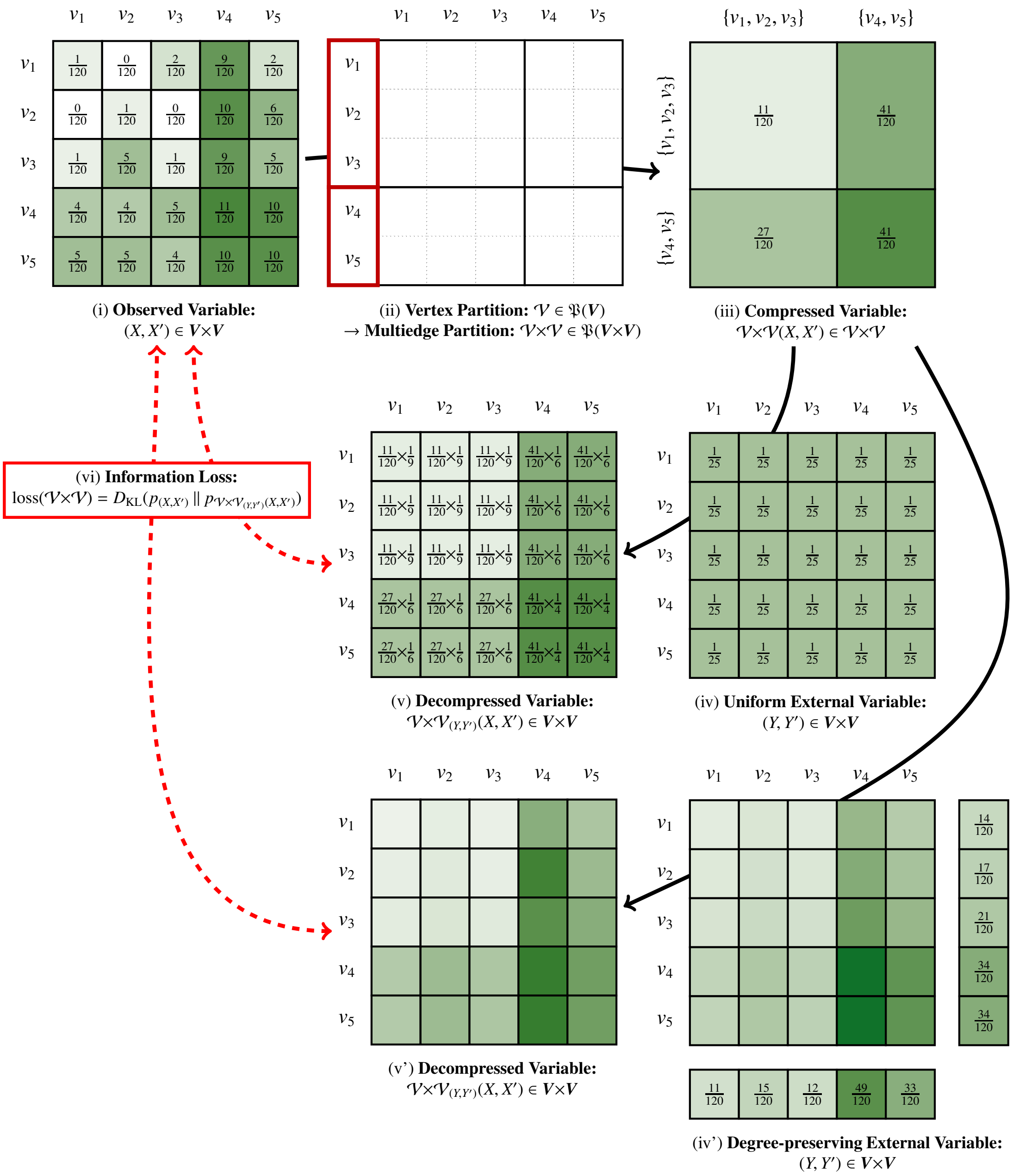
We then define the edge distribution of a multigraph that have been compressed according to a vertex partition $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ by defining a second random variable taking on the multiedge partition $\mathcal{V}\times\mathcal{V} \in \mathfrak{P}(\boldsymbol{V}\times\boldsymbol{V})$.

**Definition 7** (Compressed Variable)**.**
*The* compressed variable $\mathcal{V}\times\mathcal{V}(X, X') \in \mathcal{V}\times\mathcal{V}$ *associated with an observed variable* $(X, X') \in \boldsymbol{V}\times\boldsymbol{V}$ *and a vertex partition* $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ *is the unique couple of vertex subsets in* $\mathcal{V}\times\mathcal{V}$ *that contains* $(X, X')$*:*

$$\mathcal{V}\times\mathcal{V}(X, X') \quad = \quad (\mathcal{V}(X), \mathcal{V}(X')) \quad \in \quad \mathcal{V}\times\mathcal{V}.$$

*It hence has the following distribution*[2]*:*

$$\Pr(\mathcal{V}\times\mathcal{V}(X, X') = (V, V')) \quad = \quad \frac{e(V, V')}{e(\boldsymbol{V}, \boldsymbol{V})} \quad \overset{\text{def}}{=} \quad p_{\mathcal{V}\times\mathcal{V}(X,X')}(V, V').$$


$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$v_1$: $\frac{1}{120}$ $\frac{0}{120}$ $\frac{2}{120}$ $\frac{9}{120}$ $\frac{2}{120}$
$v_2$: $\frac{0}{120}$ $\frac{1}{120}$ $\frac{0}{120}$ $\frac{10}{120}$ $\frac{6}{120}$
$v_3$: $\frac{1}{120}$ $\frac{5}{120}$ $\frac{1}{120}$ $\frac{9}{120}$ $\frac{5}{120}$
$v_4$: $\frac{4}{120}$ $\frac{4}{120}$ $\frac{5}{120}$ $\frac{11}{120}$ $\frac{10}{120}$
$v_5$: $\frac{5}{120}$ $\frac{5}{120}$ $\frac{4}{120}$ $\frac{10}{120}$ $\frac{10}{120}$
$\{v_1, v_2, v_3\}$ $\{v_4, v_5\}$
$\frac{11}{120}$ $\frac{41}{120}$
$\frac{27}{120}$ $\frac{41}{120}$
$\frac{11}{120}\times\frac{1}{9}$ $\frac{41}{120}\times\frac{1}{6}$ $\frac{27}{120}\times\frac{1}{6}$ $\frac{41}{120}\times\frac{1}{4}$
$\frac{1}{25}$
$\frac{14}{120}$ $\frac{17}{120}$ $\frac{21}{120}$ $\frac{34}{120}$ $\frac{34}{120}$
$\frac{11}{120}$ $\frac{15}{120}$ $\frac{12}{120}$ $\frac{49}{120}$ $\frac{33}{120}$

(i) **Observed Variable:** $(X, X') \in V{\times}V$

(ii) **Vertex Partition:** $\mathcal{V} \in \mathfrak{P}(V)$ → **Multiedge Partition:** $\mathcal{V}{\times}\mathcal{V} \in \mathfrak{P}(V{\times}V)$

(iii) **Compressed Variable:** $\mathcal{V}{\times}\mathcal{V}(X, X') \in \mathcal{V}{\times}\mathcal{V}$

(vi) **Information Loss:** $\mathrm{loss}(\mathcal{V}{\times}\mathcal{V}) = D_{\mathrm{KL}}(p_{(X,X')} \parallel p_{\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X,X')})$

(v) **Decompressed Variable:** $\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X, X') \in V{\times}V$

(iv) **Uniform External Variable:** $(Y, Y') \in V{\times}V$

(v') **Decompressed Variable:** $\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X, X') \in V{\times}V$

(iv') **Degree-preserving External Variable:** $(Y, Y') \in V{\times}V$

Figure 3: Lossy compression consists in (i) modelling the multigraph as a random variable $(X, X')$ having the empirical distribution of edges in $\boldsymbol{V}{\times}\boldsymbol{V}$, (ii) choosing a vertex partition $\mathcal{V}$, and so a multiedge partition $\mathcal{V}{\times}\mathcal{V}$ that is used to compress $(X, X')$, (iii) computing the distribution of the resulting compressed variable $\mathcal{V}{\times}\mathcal{V}(X, X')$ by applying partition $\mathcal{V}{\times}\mathcal{V}$ onto $(X, X')$, (iv) taking an external variable $(Y, Y')$ (for example (iv) uniformly distributed or (iv') preserving the degree profile of vertices) to project back the distribution of $\mathcal{V}{\times}\mathcal{V}(X, X')$ into $\boldsymbol{V}{\times}\boldsymbol{V}$, (v) computing the distribution of the resulting decompressed variable $\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X, X')$ by first choosing $(X, X')$, then choosing $(Y, Y')$ conditioned on $\mathcal{V}{\times}\mathcal{V}(Y, Y') = \mathcal{V}{\times}\mathcal{V}(X, X')$, and (vi) comparing the distribution of $(X, X')$ with the distribution of $\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X, X')$ by using information-theoretic measures such as the entropy of $(X, X')$ relative to $\mathcal{V}{\times}\mathcal{V}_{(Y,Y')}(X, X')$.

In other terms, $p_{\mathcal{V}\times\mathcal{V}(X,X')}(V,V')$ represents the probability that, if one chooses an edge at random among the $e(\boldsymbol{V},\boldsymbol{V})$ edges of the multigraph, it will go from a vertex of the source subset $V$ to a vertex of the target subset $V'$. For example, matrix (ii) in Figure 3 represents the multiedge partition $\mathcal{V}\times\mathcal{V}$ induced by the vertex partition $\mathcal{V} = \{\{v_1, v_2, v_3\}, \{v_4, v_5\}\}$ and matrix (iii) then represents the distribution of the resulting compressed variable $\mathcal{V}\times\mathcal{V}(X,X')$.

In order to quantify the information that has been lost during this compression step, we propose to compare the information that is contained in the initial multigraph (that is in the observed variable $(X,X')$) with the information that is contained in the compressed multigraph (that is in the compressed variable $\mathcal{V}\times\mathcal{V}(X,X')$). To do so, we project back the compressed distribution onto the initial value space $\boldsymbol{V}\times\boldsymbol{V}$ by defining a third random variable, the *external variable* $(Y,Y') \in \boldsymbol{V}\times\boldsymbol{V}$, that models additional information that one might have at his or her disposal when trying to decompress the multigraph. It is hence assumed to have a distribution that is somehow "informative" of the initial distribution. It then induces a fourth variable, the *decompressed variable* $\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')$, that models an approximation of the initial multigraph inferred from the combined knowledge of the compressed variable $\mathcal{V}\times\mathcal{V}(X,X')$ and of the external variable $(Y,Y')$. This last variable is hence defined according to the distribution of the external variable within the multiedge subsets.

**Definition 8** (Decompressed Variable)**.**
*The* decompressed variable $\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X') \in \boldsymbol{V}\times\boldsymbol{V}$ *associated with an observed variable* $(X,X') \in \boldsymbol{V}\times\boldsymbol{V}$*, a vertex partition* $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$*, and an external variable* $(Y,Y') \in \boldsymbol{V}\times\boldsymbol{V}$*, is the result of this external variable* $(Y,Y')$ *conditioned by its compression* $\mathcal{V}\times\mathcal{V}(Y,Y')$ *being equal to the result of the compressed variable* $\mathcal{V}\times\mathcal{V}(X,X')$*:*

$$\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X') \quad = \quad (Y,Y') \mid \mathcal{V}\times\mathcal{V}(Y,Y') = \mathcal{V}\times\mathcal{V}(X,X') \quad \in \quad \boldsymbol{V}\times\boldsymbol{V}.$$

*It hence has the following distribution*[3]*:*

$$\Pr(\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X') = (v,v')) \quad = \quad \frac{e(\mathcal{V}\times\mathcal{V}(v,v'))}{e(\boldsymbol{V},\boldsymbol{V})} \; \frac{p_{(Y,Y')}(v,v')}{p_{\mathcal{V}\times\mathcal{V}(Y,Y')}(\mathcal{V}\times\mathcal{V}(v,v'))} \quad \overset{\text{def}}{=} \quad p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}(v,v').$$

In other terms, $p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}(v,v')$ represents the probability that, if one (i) chooses an edge $(u,u')$ at random among the $e(\boldsymbol{V},\boldsymbol{V})$ edges of the multigraph, (ii) considers its compressed multiedge subset $\mathcal{V}\times\mathcal{V}(u,u') = (V,V') \in \mathcal{V}\times\mathcal{V}$, and (iii) chooses a source vertex within $V$ and a target vertex within $V'$ according to the distribution $p_{(Y,Y')}$ of the external variable within this multiedge subset, then one will result with an edge going from the source vertex $v$ to the target vertex $v'$. For example, matrix (iv) in Figure 3 represents a uniformly-distributed external variable $(Y,Y') \in \boldsymbol{V}\times\boldsymbol{V}$ that is used to decompress $\mathcal{V}\times\mathcal{V}(X,X')$ (see *Blind Decompression* below). Matrix (iv') represents another such external

[2] By applying the law of total probability:

$$\begin{aligned}
\Pr(\mathcal{V}\times\mathcal{V}(X,X') = (V,V')) &= \sum_{(v,v')\in\boldsymbol{V}\times\boldsymbol{V}} \underbrace{\Pr(\mathcal{V}\times\mathcal{V}(X,X') = (V,V') \mid (X,X') = (v,v'))}_{=1 \text{ if } (v,v')\in V\times V'\text{, and } 0 \text{ else}} \Pr((X,X') = (v,v')) \\
&= \sum_{(v,v')\in V\times V'} \Pr((X,X') = (v,v')) = \sum_{(v,v')\in V\times V'} \frac{e(v,v')}{e(\boldsymbol{V},\boldsymbol{V})} = \frac{e(V,V')}{e(\boldsymbol{V},\boldsymbol{V})}
\end{aligned}$$

Note that, more generally, compression could be defined for any (possibly stochastic) function of the observed variable $(X,X')$, thus modelling what is sometimes called a "soft partitioning" of the vertices. The information-theoretic framework presented in this article, along with all the measures it contains, are straightforwardly generalisable to such setting. However, because it is often much easier to interpret the result of compression when it is based on "hard partitioning", especially in the case of vertex partitioning, we focus in this article on this simpler setting.

[3] By applying the law of total probability:

$$\begin{aligned}
\Pr(\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X') = (v,v')) &= \Pr((Y,Y') = (v,v') \mid \mathcal{V}\times\mathcal{V}(Y,Y') = \mathcal{V}\times\mathcal{V}(X,X')) \\
&= \sum_{(V,V')\in\mathcal{V}\times\mathcal{V}} \underbrace{\Pr((Y,Y') = (v,v') \mid \mathcal{V}\times\mathcal{V}(Y,Y') = (V,V'))}_{=0 \text{ if } (V,V') \neq \mathcal{V}\times\mathcal{V}(v,v')} \Pr(\mathcal{V}\times\mathcal{V}(X,X') = (V,V')) \\
&= \Pr((Y,Y') = (v,v') \mid \mathcal{V}\times\mathcal{V}(Y,Y') = \mathcal{V}\times\mathcal{V}(v,v')) \; \Pr(\mathcal{V}\times\mathcal{V}(X,X') = \mathcal{V}\times\mathcal{V}(v,v')) \\
&= \frac{\Pr((Y,Y') = (v,v'))}{\Pr(\mathcal{V}\times\mathcal{V}(Y,Y') = \mathcal{V}\times\mathcal{V}(v,v'))} \; \frac{e(\mathcal{V}\times\mathcal{V}(v,v'))}{e(\boldsymbol{V},\boldsymbol{V})}.
\end{aligned}$$

variable (see *Degree-preserving Decompression* below). Matrices (v) and (v') then represent the distribution of the resulting uniformly decompressed variable $\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')$.

*Blind Decompression.* When the external variable $(Y,Y')$ is uniformly distributed on $\boldsymbol{V}\times\boldsymbol{V}$, the decompression step is done without any additional information about the initial edge distribution:

$$p_{(Y,Y')}(v,v') \;=\; \frac{1}{|\boldsymbol{V}\times\boldsymbol{V}|} \quad\Rightarrow\quad p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}(v,v') \;=\; \frac{e(\mathcal{V}\times\mathcal{V}(v,v'))}{e(\boldsymbol{V},\boldsymbol{V})}\,\frac{1}{|\mathcal{V}\times\mathcal{V}(v,v')|}.$$

In this case, only the knowledge of the compressed variable hence is exploited. The decompressed variable is hence the result of a uniform trial among the multiedges contained in $\mathcal{V}\times\mathcal{V}(X,X')$, that is a "maximum-entropy sampling" guarantying that no additional information has been injected during decompression.

*Reversible Decompression.* To the contrary, when the external variable $(Y,Y')$ has the same distribution than the observed variable $(X,X')$, then the decompression step is done with a full knowledge of the initial edge distribution:

$$p_{(Y,Y')}(v,v') \;=\; p_{(X,X')}(v,v') \quad\Rightarrow\quad \mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')(v,v') \;=\; \frac{e(v,v')}{e(\boldsymbol{V},\boldsymbol{V})} \;=\; p_{(X,X')}(v,v').$$

In this case, the decompressed variable also has the same distribution than the observed variable, meaning that one fully restores the initial multigraph when decompressing.

*Degree-preserving Decompression.* An intermediary example of external information can be derived from the knowledge of the vertex degrees in the initial multigraph:

$$p_{(Y,Y')}(v,v') \;=\; p_X(v)\; p_{X'}(v') \;=\; \frac{e(v,\boldsymbol{V})}{e(\boldsymbol{V},\boldsymbol{V})}\,\frac{e(\boldsymbol{V},v')}{e(\boldsymbol{V},\boldsymbol{V})}$$

$$\Rightarrow\quad p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}(v,v') \;=\; \frac{e(\mathcal{V}\times\mathcal{V}(v,v'))}{e(\boldsymbol{V},\boldsymbol{V})}\,\frac{e(v,\boldsymbol{V})\;e(\boldsymbol{V},v')}{e(\mathcal{V}(v),\boldsymbol{V})\;e(\boldsymbol{V},\mathcal{V}(v'))}.$$

In this case, the decompression step takes into account the initial vertex degrees and the resulting multigraph hence has the same degree profile than the initial one. The corresponding generative model is hence similar to the one of a *configuration model* [25]: Multigraphs are sampled according to the compressed variable, while also preserving the initial degree profile.

Now that we have defined compression and decompression as sequential operations on stochastic variables, we exploit a classical measure of information theory to quantify the information that is lost during such a process. Intuitively, it consists in comparing the initial edge distribution (the one of the observed variable $(X,X')$) with the approximated edge distribution (the one of the decompressed variable $\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')$). In this article, we propose to do so by using the *relative entropy* of these two distributions – also known as the *Kullback-Leibler divergence* [26, 27] – as it is the most canonical measure of dissimilarity provided by information theory to compare an approximated probability distribution to a real one.

**Definition 9** (Information Loss)**.**
*The* information loss *induced by a vertex partition $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ on an observed variable $(X, X') \in \boldsymbol{V}\times\boldsymbol{V}$, and according to an external variable $(Y, Y') \in \boldsymbol{V}\times\boldsymbol{V}$, is given by the entropy of $(X, X')$ relative to $\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X, X')$:*

$$\begin{aligned}\text{loss}(\mathcal{V}\times\mathcal{V}) \quad &= \quad \sum_{(v,v')\in\boldsymbol{V}\times\boldsymbol{V}} p_{(X,X')}(v, v') \log_2\left(\frac{p_{(X,X')}(v, v')}{p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}(v, v')}\right)\\ &= \quad \sum_{(v,v')\in\boldsymbol{V}\times V} \frac{e(v, v')}{e(\boldsymbol{V}, \boldsymbol{V})} \log_2\left(\frac{e(v, v')}{e(\mathcal{V}\times\mathcal{V}(v, v'))}\bigg/\frac{p_{(Y,Y')}(v, v')}{p_{\mathcal{V}\times\mathcal{V}(Y,Y')}(\mathcal{V}\times\mathcal{V}(v, v'))}\right)\end{aligned}$$

*Note that information loss is* additively decomposable*. It can be expressed as a sum of information losses defined at the subset level instead of at the partition level:*

$$\text{loss}(\mathcal{V}\times\mathcal{V}) \;=\; \sum_{(V,V')\in\mathcal{V}\times\mathcal{V}} \text{loss}(V, V') \quad \textit{with} \;\; \text{loss}(V, V') \;=\; \sum_{(v,v')\in V\times V'} \frac{e(v, v')}{e(\boldsymbol{V}, \boldsymbol{V})} \log_2\left(\frac{e(v, v')}{e(V, V')}\bigg/\frac{p_{(Y,Y')}(v, v')}{p_{\mathcal{V}\times\mathcal{V}(Y,Y')}(V, V')}\right).$$

Intuitively, this measure considers the following reconstruction task: Imagine that all the edges of a multigraph have been "detached" from their vertices and put into a bag. An observer is now taking one edge out of the bag and tries to guess its initial location, that is its source and target vertices. We then compare two peoples trying to do so: One having a perfect knowledge of the distribution $p_{(X,X')}$ of the edges in the initial multigraph (*e.g.*, matrix (i) in Figure 3); The second only having an approximation $p_{\mathcal{V}\times\mathcal{V}_{(Y,Y')}(X,X')}$ of this distribution, obtained through the compression, then the decompression of the initial distribution (*e.g.*, matrices (v) and (v') in Figure 3). Relative entropy then measures the average quantity of *additional* information (in bits per edge) that the second observer needs in order to make a guess that is as informed as the guess of the first observer. In other words, relative entropy quantifies the information that has been lost during compression, that is no longer contained in the compressed graph, and that cannot be retrieved from the knowledge of the external variable.

*Blind Decompression.* When the external variable is uniformly distributed on $\boldsymbol{V}\times\boldsymbol{V}$, relative entropy simply quantifies the information that has been lost during compression, without the help of any additional information:

$$p_{(Y,Y')}(v, v') \;=\; \frac{1}{|\boldsymbol{V}\times\boldsymbol{V}|} \quad\Rightarrow\quad \text{loss}(V, V') \;=\; \sum_{vv\in V\times V'} \frac{e(v, v')}{e(\boldsymbol{V}, \boldsymbol{V})} \log_2\left(\frac{e(v, v')}{e(V, V')}|V\times V'|\right).$$

*Reversible Decompression.* When the external variable $(Y, Y')$ has the same distribution than the observed variable $(X, X')$, compression then induces no information loss – whatever the chosen vertex partition $\mathcal{V}$ – since all the information required to reconstruct the initial multigraph is reinjected during the decompression step:

$$p_{(Y,Y')}(v, v') \;=\; p_{(X,X')}(v, v') \quad\Rightarrow\quad \text{loss}(V, V') \;=\; 0.$$

*Degree-preserving Decompression.* In this intermediary context, relative entropy quantifies the information that has been lost during compression, and that cannot be retrieved from the additional knowledge of the vertex degrees in the initial multigraph:

$$\begin{aligned}p_{(Y,Y')}(v, v') \;=\; p_X(v)\, p_{X'}(v') \;&=\; \frac{e(v, \boldsymbol{V})}{e(\boldsymbol{V}, \boldsymbol{V})}\frac{e(\boldsymbol{V}, v')}{e(\boldsymbol{V}, \boldsymbol{V})}\\ &\Rightarrow\quad \text{loss}(V, V') \;=\; \sum_{(v,v')\in V\times V'} \frac{e(v, v')}{e(\boldsymbol{V}, \boldsymbol{V})} \log_2\left(\frac{e(v, v')}{e(V, V')}\bigg/\frac{e(v, \boldsymbol{V})\, e(\boldsymbol{V}, v')}{e(V, \boldsymbol{V})\, e(\boldsymbol{V}, V')}\right).\end{aligned}$$

*Related Measures.* Relative entropy is one among many measures that can be found in the literature to quantify information loss in graph compression. Given an initial multigraph and a decompressed one, which is described by the approximated edge distribution, any measure of weighted graph similarity may be relevant to quantify the impact of compression [1]: *E.g.*, the percentage of edges in common, the size of the maximum common subgraph or of the minimum common supergraph, the edit distance, that is the insertion and removal of vertices and edges needed to go from one graph to another [28], or any measure aggregating the similarities of vertices from graph to graph (*e.g.*, Jaccard index, Pearson coefficient, cosine similarity on vertex neighbourhoods).

More sophisticated graph-theoretical measures go beyond the mere level of edges by taking into account paths within the two compared graphs [2]: "[T]he best path between any two nodes should be approximately equally good in the compressed graph as in the original graph, but the path does not have to be the same." More generally, query-based measures aim at quantifying the impact of compression on the results of goal-oriented queries regarding the graph structure: *E.g.*, queries about shortest paths, about degrees and adjacency, about centrality and community structures (see for example [29, 30, 31]). The expected difference between the results of queries on the initial graph and the results of queries on the decompressed graph thus provides a reconstruction error that serves as a goal-oriented information loss [6]. To some extent, we are interested in this article in adjacency-oriented queries, that is the most canonical ones, taking the perspective of the weighted adjacency matrices of the two compared graphs: *What is the weight of the multiedge located between two given vertices of the initial multigraph?* Relative entropy measures the expected error when answering this query from the only knowledge of the compressed graph.

More generally, this perspective is related to the density profile of edges within the graph. Hence, density-based measures [5] seems more relevant than other traditional connectivity measures: *E.g.*, the Euclidean distance or the mean squared error between the two density matrices [2, 3], the average variance within matrix blocks [13], and many other measures inherited from traditional block modelling methods [20]. Note that, when it comes to the latter, the stochastic model underlying our compression framework is similar, but not equivalent to the one of block modelling. The compressed matrix describes in our case the parameters of a multinomial distribution from which the graph's edges are sampled, whereas it describes in the case of block modelling the parameters of $|\boldsymbol{V}\times\boldsymbol{V}|$ independent Bernouilli distributions. In other words, our model gives the probability that an edge – taken at random – is located between two given vertices, and not the probability that an edge exists between two given vertices (see for example block model compression in [6]).

Because of this particular stochastic model, we chose in this article an information-theoretic approach to measure information loss. This allows to derive a measure that is clearly in line with the defined model and which can be easily interpreted within the realm of information theory. Among tools provided by this theory, other approaches use the principles of *minimum description length* [4, 6] to compress a graph using the density-based model in an optimal fashion. In the same line of thinking, traditional tools for Bayesian inference propose to interpret the compressed graph as a generative model and the initial graph as observed data, then computes the likelihood of the data given the model as a measure of information loss [15]. A similar Bayesian interpretation of relative entropy could be given, as it measures the difference of likelihood between two generative models of the multigraph: One corresponding to $e(\boldsymbol{V}, \boldsymbol{V})$ independent trials with the empirical distribution of the multigraph's edges; The second corresponding to $e(\boldsymbol{V}, \boldsymbol{V})$ independent trials with the distribution obtained through compression, and then decompression. Similarly, *co-clustering* [23] interprets the graph's adjacency matrix as the joint probability distribution of two random variables, then finds two vertex partitions that minimise the loss in mutual information between these two variables [27] from the initial graph to the compressed graph. This is shown to be equivalent to minimising the relative entropy between the initial distribution and a decompressed distribution that preserves the marginal values. It is hence equivalent to our measure of information loss in the particular case of a decompression scheme that takes into account additional information regarding the vertex degrees.

### *3.3. From Vertex to Edge Partitions*

By providing a measure of information loss, previous subsection focuses on the *objective function* of the GCP, that is on the quality measure to be optimised. We now focus on the *solution space* of this problem, that is the set of partitions that one actually consider for compression. The original GCP presented in Section 2 consists in using a vertex partition $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$ to then determine a multiedge "grid" partition (see top-left matrix of Figure 4):

$$\mathcal{V}\times\mathcal{V} \quad = \quad \{V\times V' \;:\; V\in\mathcal{V} \;\wedge\; V'\in\mathcal{V}\} \quad \in \quad \mathfrak{P}(\boldsymbol{V}\times\boldsymbol{V}),$$

such that two multiedges $(v, v')$ and $(u, u')$ are in the same multiedge subset $V\times V'$ if and only if their source vertices are both in $V$ and their target vertices are both in $V'$:

$$\mathcal{V}\times\mathcal{V}(v,v') = \mathcal{V}\times\mathcal{V}(u,u') \qquad \Leftrightarrow \qquad \mathcal{V}(v)=\mathcal{V}(u) \quad \text{and} \quad \mathcal{V}(v')=\mathcal{V}(u').$$

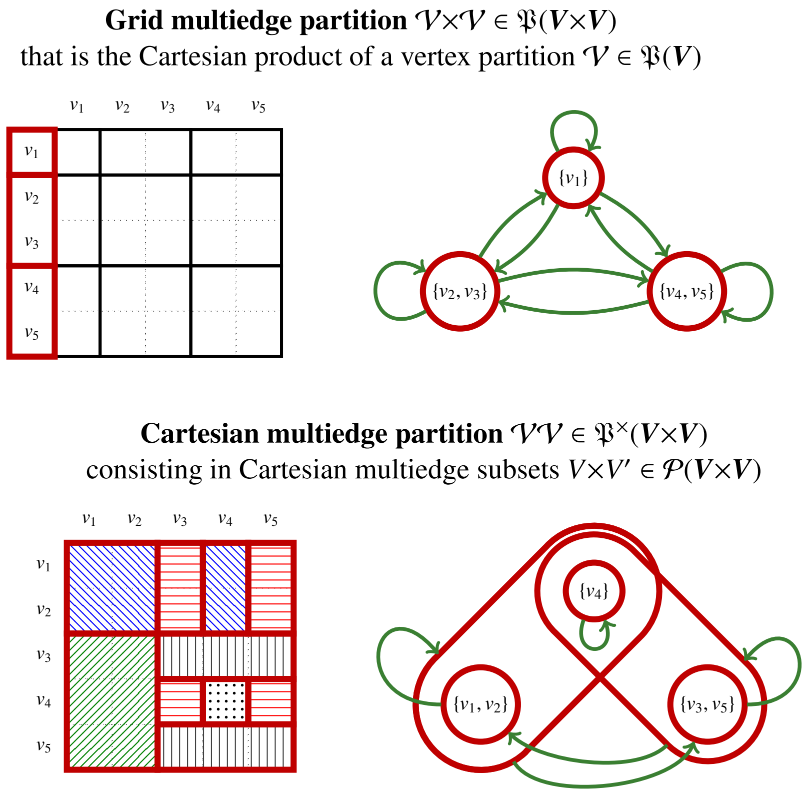

**Grid multiedge partition** $\mathcal{V}\times\mathcal{V} \in \mathfrak{P}(\boldsymbol{V}\times\boldsymbol{V})$
that is the Cartesian product of a vertex partition $\mathcal{V} \in \mathfrak{P}(\boldsymbol{V})$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$\{v_1\}$
$\{v_2, v_3\}$
$\{v_4, v_5\}$
**Cartesian multiedge partition** $\mathcal{V}\mathcal{V} \in \mathfrak{P}^{\times}(\boldsymbol{V}\times\boldsymbol{V})$
consisting in Cartesian multiedge subsets $V\times V' \in \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V})$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$v_1$ $v_2$ $v_3$ $v_4$ $v_5$
$\{v_4\}$
$\{v_1, v_2\}$
$\{v_3, v_5\}$

Figure 4: Two partitioning schemes that one might consider to define the solution space of the GCP. Multiedge subsets are represented with hatching when they are not "compact tiles".

In other terms, the induced two-dimensional partitioning of the multiedge set $\boldsymbol{V}\times\boldsymbol{V}$ is the Cartesian product of a one-dimensional partitioning of the vertex set $\boldsymbol{V}$. This first compression scheme is classically used in macroscopic graph models such as block models and community-based representations. One of the reason is that the result of compression can still be represented as a graph, which vertices are the subsets of the vertex partition and which multiedges are the subsets of the multiedge "grid" partition (see top-right graph of Figure 4)

However, this results in a quite constrained solution space for the optimisation problem: Only a small number of partitions of $\boldsymbol{V}\times\boldsymbol{V}$ are actually feasible. One might instead consider a less constrained solution space by allowing a larger number of multiedge partitions to be used for compression, by for example considering partitions of the multiedge set $\boldsymbol{V}\times\boldsymbol{V}$ that are made of Cartesian products of two vertex subsets $V\times V' \subseteq \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V})$.

**Definition 10** (Cartesian Multiedge Partitions)**.**
*Given a vertex set* $\boldsymbol{V}$*, the set of* Cartesian multiedge partitions $\mathfrak{P}^{\times}(\boldsymbol{V}\times\boldsymbol{V}) \subset \mathfrak{P}(\boldsymbol{V}\times\boldsymbol{V})$ *is the set of multiedge partitions that are made of Cartesian products of vertex subsets*[4]*:*

$$\begin{aligned}\mathfrak{P}^{\times}(\boldsymbol{V}\times\boldsymbol{V}) \quad &= \quad \{\{(V_1 \times V_1'), \ldots, (V_m \times V_m')\} \subseteq \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V}) \\ &\qquad : \ \cup_i (V_i \times V_i') = \boldsymbol{V}\times\boldsymbol{V} \ \wedge \ \forall i \neq j,\ (V_i \times V_i') \cap (V_j \times V_j') = \emptyset\}.\end{aligned}$$

Such Cartesian partitions of $\boldsymbol{V}\times\boldsymbol{V}$ hence contains "rectangular" multiedge subsets[5] (see bottom-left matrix of Figure 4). Hence, although the result of compression can no longer be represented as a graph, it can be represented as

[4] There is an abuse of notation in this definition when writing that "$V\times V' \in \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V})$". We should instead write that "$V\times V' \in \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V})$ such that $(V, V') \in \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V})$", but we prefer the former for notation conciseness.

[5] Note that each of these multiedge subsets is indeed a rectangle in the adjacency matrix, modulo a reordering of its rows and columns. It might happen however that no reordering can make all multiedge subsets look rectangular *at the same time*.

a *directed hypergraph* – or as a *power graph* [8, 9] – that is a graph where edges can join couples of vertex subsets instead of couples of vertices (see bottom-right graph of Figure 4).

*Related Work.* Most classical approaches for graph compression are based on the discovery of *vertex partitions*. This is for example the case of most work on graph summarisation [5, 6], block modelling [13], community detection [22], and modular decomposition [3]. As previously said, one of the interests of such vertex partitioning is that the result of compression is still a graph (a set of compressed vertices connected by a set of compressed edges) that can hence be represented, analysed, and visualised with traditional tools [4, 2]. However, the number and diversity of feasible solutions – that is the set of vertex partitions – is quite limited when compated to the full space of matrix partitions.

As proposed above, some approaches hence focus on *edge partitions*, instead of vertex partitions, in order to provide more expressive compression schemes, but only on *particular* edge partitions, in order to preserve some fundamental structures of graph data. The most generic framework is formalised by *power graph analysis* [8, 9] and *Mondrian processes* [32], where edge subsets are only required to be the Cartesian product of vertex subsets. Such edge subsets can hence be represented as compressed edges between couples of compressed vertices. But since different edge subsets can lead to overlapping vertex subsets, the resulting compressed graph is no longer a graph, but an *hypergraph* (or a *power graph*). While still interpretable within the broad realm of graph theory, power graphs are more expressive [2] than classical approaches – such as community partitions – since a given vertex might belong to different similarity groups: It might be similar to a given group of vertices with respect to a given part of the graph, and similar to another group of vertices with respect to another part (see examples in [9]).

### *3.4. Adding Constraints to the Set of Feasible Vertex Subsets*

A wide variety of approaches span from generic edge partitions to more constrained schemes that “impose restrictions on the nature of overlap” for example by “fixing the topology of connectivity between overlapping nodes” [9] such as in clique percolation, spin models, mixed-membership block models, latent attribute models, spectral clustering, and link communities. This aims at expressing particular topological properties within the compression scheme in order to produce meaningful graphs with respect to some particular analysis framework. “Frequent patterns possibly reflect some semantic structures of the domain and therefore are useful candidates for replacement” [1].

In the partitioning scheme hereabove presented, the Cartesian product of *any* two vertex subsets $(V, V') \in \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V})$ gives a feasible multiedge subset $V\times V' \in \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V})$. In other terms, the solution space is only limited by this Cartesian principle and by the partitioning constraints (covering and non-overlapping subsets). However, one can also control the shape of feasible partitions by directly specifying the feasible vertex subsets that can be combined to build the “rectangular” multiedge subsets of the Cartesian partition. Such additional constraints might prove to be useful to incorporate domain knowledge about particular vertex structures within the compression scheme: By driving the process, feasibility constraints might indeed facilitate the interpretation of the compression’s result according to the expert domain.

**Definition 11** (Feasible Multiedge Partitions)**.**
*Given a set of feasible vertex subsets $\hat{\mathcal{P}}(\boldsymbol{V}) \subseteq \mathcal{P}(\boldsymbol{V})$, the set of* feasible Cartesian multiedge partitions $\hat{\mathfrak{P}}^\times(\boldsymbol{V}\times\boldsymbol{V}) \subseteq \mathfrak{P}^\times(\boldsymbol{V}\times\boldsymbol{V})$ *is the set of Cartesian multiedge partitions which multiedge subsets are only made of feasible vertex subsets:*

$$\begin{aligned}\hat{\mathfrak{P}}^\times(\boldsymbol{V}\times\boldsymbol{V}) \;=\; & \{\{(V_1\times V'_1), \ldots, (V_m\times V'_m)\} \in \hat{\mathcal{P}}(\boldsymbol{V})\times\hat{\mathcal{P}}(\boldsymbol{V}) \\ & : \cup_i (V_i\times V'_i) = \boldsymbol{V}\times\boldsymbol{V} \;\wedge\; \forall i \neq j,\ (V_i\times V'_i)\cap(V_j\times V'_j) = \emptyset\}.\end{aligned}$$

In the most general case, the set of feasible vertex subsets $\hat{\mathcal{P}}(\boldsymbol{V})$ can be composed of all vertex subset in $\mathcal{P}(\boldsymbol{V})$. However, one can use more constrained – and hence more meaningful vertex structures – to drive the compression scheme and its possible applications. A survey about the types of constraints that have been used in many different domains can be found in [33]. We present below some of these structures and their basic combinatorics.

*The Complete Case.* When no additional constraint applies to vertex subsets, then:

$$\hat{\mathcal{P}}(\boldsymbol{V}) \;=\; \mathcal{P}(\boldsymbol{V}), \quad \text{and so} \quad \hat{\mathfrak{P}}(\boldsymbol{V}) = \mathfrak{P}(\boldsymbol{V}).$$

Such scheme can be used when no additional useful structure is known about the vertex set, for example to model *coalition structures* in multi-agent systems when assuming that every possible group of agents is an adequate candidate to constitute a coalition [34, 35]. It has been shown that, in this case, the number of feasible partitions grows considerably faster than the number of feasible subsets [36]:

$$|\mathcal{P}(\boldsymbol{V})| = 2^n \quad \text{and} \quad |\mathfrak{P}(\boldsymbol{V})| = \omega(n^{n/2}),$$

where $n = |\boldsymbol{V}|$ is the number of vertices in the graph.

*Hierarchies of Vertex Subsets.* A *hierarchy* $\hat{\mathcal{P}}(\boldsymbol{V}) = \mathcal{H}(\boldsymbol{V})$ is such that any two feasible vertex subsets are either disjoint, or one is included in the other:

$$\forall (V_1, V_2) \in \mathcal{H}(\boldsymbol{V})^2, \quad V_1 \cap V_2 = \emptyset \ \vee \ V_1 \subseteq V_2 \ \vee \ V_2 \subseteq V_1.$$

We mark $\hat{\mathfrak{P}}(\boldsymbol{V}) = \mathfrak{H}(\boldsymbol{V})$ the resulting set of feasible vertex partitions. Such vertex hierarchies can be used to model graphs that are known to have a multilevel nested structure that one wants to preserve during compression. This might include, among others, a *community structure* that have been preliminarily identified through a hierarchical community detection algorithm [37], a sequence of *geographical nested partitions* of the world's territorial units [24]; a *hierarchical communication network* in distributed computers [19].

It has been shown that the number of feasible subsets in a hierarchy is asymptotically bounded from above by the number of objects it contains [33]: $|\mathcal{H}(\boldsymbol{V})| = O(n)$. The resulting number of feasible partitions however depends on the number of levels and branches in the hierarchy. For a complete binary tree, it is asymptotically bounded by an exponential function: $|\mathfrak{H}(\boldsymbol{V})| = \Theta(\alpha^n)$, with $\alpha \approx 1.226$ [38]. Similar results have been found for complete ternary trees (with $\alpha \approx 1.084$ [39]), complete quaternary trees, and so on. Henceforth, for any bounded number of children per node in the hierarchy, the number of feasible partitions exponentially grows with the number of objects.

*Sets of Vertex Intervals.* Any total order $<$ on the vertex set $\boldsymbol{V}$ induces a *set of vertex intervals* $\hat{\mathcal{P}}(\boldsymbol{V}) = \mathcal{I}(\boldsymbol{V})$ defined as follows:

$$\mathcal{I}(\boldsymbol{V}) \ = \ \{[v_1, v_2] \ : \ (v_1, v_2) \in \boldsymbol{V}^2\} \quad \text{where} \quad [v_1, v_2] \ = \ \{v \in \boldsymbol{V} : v_1 \leq v \leq v_2\}.$$

We mark $\hat{\mathfrak{P}}(\boldsymbol{V}) = \mathfrak{I}(\boldsymbol{V})$ the resulting set of feasible vertex partitions. It can be represented as a "pyramid of intervals" [33] and such partitions are sometimes called *consecutive partitions* [40]. Such sets of intervals naturally apply to vertices having a temporal feature (*e.g.*, events, dates, or time periods are naturally ordered by the "arrow of time") and have hence been exploited for the aggregation of time series [41, 42, 24]. They might also model unidimensional spacial features, such as the geographical ordering of cities on a coast, or on a transport route [43].

The number of intervals of an ordered set of size $n$ is $|\mathcal{I}(\boldsymbol{V})| = \frac{n(n+1)}{2}$. The resulting number of feasible partitions is $|\mathfrak{I}(\boldsymbol{V})| = 2^{n-1}$ [33].

### 3.5. From Multigraphs to Multistreams

The last step of our generalisation work regards the integration of a temporal dimension within our framework in order to finally deal with the compression of temporal graphs [10]. As argued in the introduction of this article, we build on the *link stream* representation of such graphs [11, 12]. However, because we also want to deal with cases for which multiple edges are allowed between two vertices at a given time instance, we actually generalise from streams to multistreams, as we did in Subsection 3.1 to generalise from graphs to multigraphs.

**Definition 12** (Directed Multistream)**.**
*A* directed multistream $MS = (\boldsymbol{V}, \boldsymbol{T}, e)$ *is characterised by:*

- *A set of* vertices $\boldsymbol{V}$*;*
- *A set of* time instances $\boldsymbol{T} \subseteq \mathbb{R}$*;*
- *A multiset of* directed edges $(\boldsymbol{V} \times \boldsymbol{V} \times \boldsymbol{T}, e)$
  *where* $e : \boldsymbol{V} \times \boldsymbol{V} \times \boldsymbol{T} \to \mathbb{N}$ *is the* edge function*, that is the multiplicity function counting the number of edges* $e(v, v', t) \in \mathbb{N}$ *going from a given source vertex* $v \in \boldsymbol{V}$ *to a given target vertex* $v' \in \boldsymbol{V}$ *at a given time instance* $t \in \boldsymbol{T}$*.*

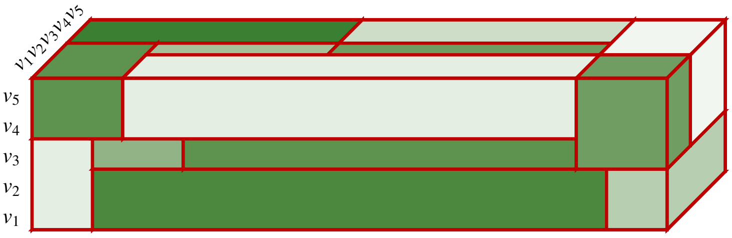


Figure 5: Multistream compression, that is the Cartesian partitioning of $\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$, consists in the natural tridimensional generalisation of multigraph compression, that is the Cartesian partitioning of $\boldsymbol{V}\times\boldsymbol{V}$.

In this context, the edge function counts the number of interactions happening between vertices at a given time: $e(v, v', t)$ is the number of edges going from source vertex $v$ to target vertex $v'$ at time $t$.

As illustrated in Figure 5, this formalism constitutes an elegant solution to generalise our compression scheme since it simply adds a third dimension $\boldsymbol{T} \subseteq \mathbb{R}$ to the multigraph's definition. The GCP is then generalised to this three-dimensional formalism by (i) defining a set of feasible time subsets that preserves the ordering of time instances, that is a set of intervals $\hat{\mathcal{P}}(\boldsymbol{T}) = \mathcal{I}(\boldsymbol{T})$ (see previous subsection), (ii) considering three-dimensional Cartesian multiedge subsets $V\times V'\times T \in \hat{\mathcal{P}}(\boldsymbol{V})\times\hat{\mathcal{P}}(\boldsymbol{V})\times\mathcal{I}(\boldsymbol{T})$, and (iii) computing the information loss on this generalised space in a similar fashion than for the static version of the compression problem. Here is the resulting generalisation in details.

see Definition 5

- Time instances: $t \in \boldsymbol{T} \subseteq \mathbb{R}$;
- Time subsets: $T \in \mathcal{P}(\boldsymbol{T})$;
- Multiedges: $(v, v', t) \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$;
- Edge function: $e : \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T} \to \mathbb{N}$;
- Compressed edge function:

$$e : \mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{V})\times\mathcal{P}(\boldsymbol{T}) \to \mathbb{N} \quad \text{with} \quad e(V, V', T) = \sum_{(v,v')\in V\times V'} \int_{t\in T} e(v, v', t)\, dt;$$

see Def. 11

- Feasible time subsets, that is time intervals: $T = [t_1, t_2] \in \hat{\mathcal{P}}(\boldsymbol{T}) = \mathcal{I}(\boldsymbol{T}) \subset \mathcal{P}(\boldsymbol{T})$;
- Feasible multiedge subsets: $V\times V'\times T \in \hat{\mathcal{P}}(\boldsymbol{V})\times\hat{\mathcal{P}}(\boldsymbol{V})\times\mathcal{I}(\boldsymbol{T}) \subset \mathcal{P}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$;

see Def. 10

- Feasible Cartesian multiedge partitions:

$$\mathcal{V}\mathcal{V}\mathcal{T} \in \hat{\mathfrak{P}}^{\times}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}) = \{\{(V_1\times V'_1\times T_1), \dots, (V_m\times V'_m\times T_m)\} \in \hat{\mathcal{P}}(\boldsymbol{V})\times\hat{\mathcal{P}}(\boldsymbol{V})\times\mathcal{I}(\boldsymbol{T}) \\ : \cup_i(V_i\times V'_i\times T_i) = \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T} \wedge (V_i\times V'_i\times T_i) \cap (V_j\times V'_j\times T_j) = \emptyset\};$$

see Definition 6

- Observed variable:

$$(X, X', X'') \in \boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T} \quad \text{with} \quad f_{(X,X',X'')}(v, v', t) = \frac{e(v, v', t)}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})};$$

see Definition 7

- Partition function:

$$\mathcal{V}\mathcal{V}\mathcal{T}(v, v', t) \text{ is the only multiedge subset } V{\times}V'{\times}T \text{ in } \mathcal{V}\mathcal{V}\mathcal{T} \text{ that contains } (v, v', t);$$

- Compressed variable:

$$\mathcal{V}\mathcal{V}\mathcal{T}(X, X', X'') \in \mathcal{V}\mathcal{V}\mathcal{T} \quad \text{with} \quad f_{\mathcal{V}\mathcal{V}\mathcal{T}(X,X',X'')}(V, V', T) = \frac{e(V, V', T)}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})};$$

see Definition 8

- External variable, in the case of a degree-preserving compression:

$$(Y, Y', Y'') \in \mathcal{V}\mathcal{V}\mathcal{T} \quad \text{with} \quad f_{(Y,Y',Y'')}(v, v', t) = \frac{e(v, \boldsymbol{V}, \boldsymbol{T})}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})} \frac{e(\boldsymbol{V}, v', \boldsymbol{T})}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})} \frac{e(\boldsymbol{V}, \boldsymbol{V}, t)}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})};$$

- Decompressed variable:

$$\mathcal{V}\mathcal{V}\mathcal{T}_{(Y,Y',Y'')}(X, X', X'') \in \boldsymbol{V}{\times}\boldsymbol{V}{\times}\boldsymbol{T}$$

$$\text{with} \quad f_{\mathcal{V}\mathcal{V}\mathcal{T}_{(Y,Y',Y'')}(X,X',X'')}(v, v', t) = \frac{e(V, V', T)}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})} \frac{e(v, \boldsymbol{V}, \boldsymbol{T})\, e(\boldsymbol{V}, v', \boldsymbol{T})\, e(\boldsymbol{V}, \boldsymbol{V}, t)}{e(V, \boldsymbol{V}, \boldsymbol{T})\, e(\boldsymbol{V}, V', \boldsymbol{T})\, e(\boldsymbol{V}, \boldsymbol{V}, T)}$$

$$\text{where } V{\times}V'{\times}T = \mathcal{V}\mathcal{V}\mathcal{T}(v, v', t);$$

see Definition 9

- Information loss, decomposed at the level of multiedge subsets:

$$\text{loss}(V, V', T) = \sum_{(v,v')\in V\times V'} \int_{t\in T} \frac{e(v, v', t)}{e(\boldsymbol{V}, \boldsymbol{V}, \boldsymbol{T})} \log_2 \left( \frac{e(v, v', t)}{e(V, V', T)} \Big/ \frac{e(v, \boldsymbol{V}, \boldsymbol{T})\, e(\boldsymbol{V}, v', \boldsymbol{T})\, e(\boldsymbol{V}, \boldsymbol{V}, t)}{e(V, \boldsymbol{V}, \boldsymbol{T})\, e(\boldsymbol{V}, V', \boldsymbol{T})\, e(\boldsymbol{V}, \boldsymbol{V}, T)} \right) dt.$$

The use of *integrals* instead of discrete sums to define the edge function, as well as the use of probability *density function* instead of discrete probability distribution to define the random variables, is due to the fact that the added temporal dimension $\boldsymbol{T}$ is continuous, contrary to the vertex set $\boldsymbol{V}$. A simpler setting for temporal graphs would consist in assuming a discrete representation of time, that would hence only require discrete operators. In this regard, the optimisation algorithm that is introduced in next section to solve the generalised GCP is only provided for the discrete case, for simplicity reasons.

*Related Work.* As discussed in the introduction of this article, and as illustrated in this subsection, the recent work on the *link stream* formalism [11, 12] proposes to deal with time as a simple addition to the graph's structural dimension. Considering temporal graphs as genuine tridimensional data, the arbitrary separation of structure and time is avoided, thus making the generalisation quite natural. Note that similar generalisation objectives have been addressed in previous work on graph compression, as for example the application of bidimensional *block models* to multidimensional matrices [13] or the application of *biclustering* to triplets of variables [14], which has then been exploited for the statistical analysis of temporal graphs [15]. The particular interest of such approaches also consists in the fact that they provide a unified compression scheme in which structural and temporal information is simultaneously taken into account.

## 4. Result: The Lossy Multistream Compression Problem

This section integrates all generalisations that have been proposed in previous section to the GCP in order to properly formalise the Lossy Multistream Compression Problem (MSCP). It then proposes an algorithmic solution to this problem by reducing it to the *Set Partitioning Problem*, a well-known combinatorial optimisation problem allowing to exploit state-of-the-art approaches.

For brevity purposes, pages 21 to 26 of this paper are not printed in the deliverable. To get the full document, please go to:

https://arxiv.org/abs/1807.06874

| For $\lvert\boldsymbol{V}\rvert = n$ vertices,<br>$\hat{\mathcal{P}}(\boldsymbol{T}) = \mathcal{I}(\boldsymbol{T})$,<br>and $\hat{\mathcal{P}}(\boldsymbol{V}) = \dots$ | **Number of nodes**<br>Number of *vertex* subsets<br>or *multiedge* subsets | | **Number of links**<br>Number of *vertex* coverings<br>or *multiedge* coverings | |
|---|---|---|---|---|
| | $\lvert\hat{\mathcal{P}}(\boldsymbol{V})\rvert$ | $\lvert\hat{\mathcal{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})\rvert$ | $\lvert\hat{\mathfrak{C}}(\boldsymbol{V})\rvert$ | $\lvert\hat{\mathfrak{C}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})\rvert$ |
| $\mathcal{P}(\boldsymbol{V})$ (complete set) | $\Theta(2^n)$ | $\Theta(4^n)$ | $\Theta(3^n)$ | $\Theta(6^n)$ |
| $\mathcal{I}(\boldsymbol{V})$ (set of intervals) | $\Theta(n^2)$ | $\Theta(n^6)$ | $\Theta(n^3)$ | $\Theta(n^7)$ |
| $\mathcal{H}(\boldsymbol{V})$ (hierarchy) | $\Theta(n)$ | $\Theta(n^4)$ | $\Theta(n)$ | $\Theta(n^5)$ |

Table 1: Size of the poset structure in terms of nodes (columns 1 and 2) and in terms of in-coming links (columns 3 and 4) for a unidimensional poset structure (columns 1 and 3) and for a tridimensional poset structure (columns 2 and 4) when different types of feasibility constraints apply to the set of vertex subsets (rows). The set of feasible time subsets is here assumed to be the set of time intervals: $\hat{\mathcal{P}}(\boldsymbol{T}) = \mathcal{I}(\boldsymbol{T}) \Rightarrow |\hat{\mathcal{P}}(\boldsymbol{T})| = \Theta(n)$ and $|\hat{\mathfrak{C}}(\boldsymbol{T})| = \Theta(n^3)$.

*Complexity of the resulting optimisation algorithm.* The space complexity of the resulting optimisation algorithm is given in Table 1 by the number of nodes $|\hat{\mathcal{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})|$ and by the number of links $|\hat{\mathfrak{C}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})|$ that are encoded in the tridimensional poset structure. It is exponential in the worst case, that is when all vertex subsets are feasible, polynomial of order 7 in the case of a set of vertex intervals, and polynomial of order 5 in the case of a vertex hierarchy.

The unidimensional poset structures, encoding the set of feasible vertex subsets and the set of time intervals, are considered as inputs of the optimisation problem and their building cost is hence not taken into account in the algorithm's complexity, although it is quite cheap and straightforward in the case of sets of intervals and hierarchies. Building the corresponding tridimensional poset structure requires as many operations as there are nodes and in-coming links. Filling it with the values of the cost function requires as many operations as there are in-coming links, thanks to the recursive decomposition of costs. Hence, the time complexity to build the overall data structure is equivalent to its space complexity.

Finally, when applied to the maximal element, that is to the multiedge set $\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T}$, Algorithm 2 is then recursively applied once to all feasible multiedge subsets $V\times V'\times T \in \hat{\mathcal{P}}(\boldsymbol{V}\times\boldsymbol{V}\times\boldsymbol{T})$. The bottleneck is then the summation of costs that are retrieved by the recursive calls, for each multiedge subset of each covered partition. Hence, here again, there are as many such operations as they are in-coming links in the tridimensional poset structure, so the overall time complexity of the approach is equivalent to the space complexity of the data structure we presented.

## 5. Conclusion

This article presents a formal framework for the compression of temporal graphs. By summarising homogeneous parts of the graph and replacing them with more general structural patterns, compression allows to reduce its description length while preserving its information content. This framework first builds on a simple and limited combinatorial problem, that we call the *Lossless Graph Compression Problem*, which exploits the (most classical) structural equivalence relation between vertices for the exact compression of simple graphs. Among the proposed generalisations to address the more complex *Lossy Multistream Compression Problem*, dealing with the approximated compression of temporal multigraphs, three main contributions are worth mentioning:

- The definition of an information-theoretic measure, relying on a proper formalisation of a multigraph stochastic model, to quantify and to control the information that is lost during compression, while also taking into account additional information that might be reinjected during the decompression step;
- The enhancement of the solution space of the initial problem (i) by defining a less constrained partitioning of the graph working at the multiedge level instead of the vertex level, and (ii) by allowing to express and to preserve additional vertex structures during compression;
- The generalisation from static graphs to temporal graphs by exploiting the *link stream* representation, which consists in the extension of the set of multiedges by a third dimension representing the temporal evolution of these edges, thus allowing the natural extension of all notions that have been previously introduced.

Building on a previous algorithmic framework to solve special versions of the *Set Partitioning Problem*, an exact algorithm is finally introduced for the Lossy MSCP. While it is exponential in the worst case, it is showed to be polynomial when particular vertex structures are assumed. Yet, in order to achieve the compression of large-scale temporal graphs, future research would need to work on heuristics for the approximate solving of the Lossy MSCP. This would require the definition of adequate operators on multiedge partitions to browse the solution space, taking into account its particular algebraic structure to efficiently evaluate slight modifications of the considered partitions, and to thus proceed to a greedy search for a local optima. Another improvement, regarding the current implementation of the optimisation algorithm, would consists in the acknowledgement that most link streams that are considered in empirical research are quite sparse, meaning that the support of the edge function is quite small. Hence, the data structures proposed in this article would benefit from a sparse representation of the data to decrease computation resources in real-case applications of this framework.

### Acknowledgement

The author sincerely thanks Yves Demazeau, Damien Dosimont, Matthieu Latapy, Léonard Panichi, Hindol Rakshit, Fabrice Rossi, Mridul Seth, Catherine Matias, Lucas Mello Schnorr, Lionel Tabourier, Fabien Tarissan, Tiphaine Viard, and Jean-Marc Vincent, for their valuable feedback. He also gratefully thanks all the members of Prof. Grasset's team for their warm welcome within their premises. This work has been partially funded by the European Commission H2020 FETPROACT 2016-2017 program under grant 732942 (ODYCCEUS) and by the French National Agency of Research (ANR) under grant ANR-15-CE38-0001 (AlgoDiv).

### References

### References

[1] F. Zhou, S. Mahler, H. Toivonen, Review of Network Abstraction Techniques, in: Workshop on Explorative Analytics of Information Networks at ECML PKDD, 2009, pp. 50–63.

[2] H. Toivonen, F. Zhou, A. Hartikainen, A. Hinkka, Compression of Weighted Graphs, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11), ACM, New York, NY, USA, 2011, pp. 965–973.

[3] P. Serafino, Speeding Up Graph Clustering via Modular Decomposition Based Compression, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC'13), ACM, New York, NY, USA, 2013, pp. 156–163.

[4] S. Navlakha, R. Rastogi, N. Shrivastava, Graph Summarization with Bounded Error, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08), ACM, New York, NY, USA, 2008, pp. 419–432.

[5] N. Zhang, Y. Tian, J. M. Patel, Discovery-Driven Graph Summarization, in: Proceedings of the 26th International Conference on Data Engineering (ICDE'2010), 2010, pp. 880–891.

[6] K. LeFevre, E. Terzi, GraSS: Graph Structure Summarization, in: Proceedings of the SIAM International Conference on Data Mining (SDM'2010), 2010, pp. 454–465.

[7] B. Pinaud, G. Melancon, J. Dubois, PORGY: A Visual Graph Rewriting Environment for Complex Systems, Computer Graphics Forum 31 (3) (2012) 1265–1274.

[8] T. Dwyer, N. H. Riche, K. Marriott, C. Mears, Edge Compression Techniques for Visualization of Dense Directed Graphs, IEEE Transactions on Visualization and Computer Graphics 19 (12) (2013) 2596–2605.

[9] S. E. Ahnert, Generalised power graph compression reveals dominant relationship patterns in complex networks, Scientific Reports 4 (4385).

[10] P. Holme, J. Saramäki (Eds.), Temporal Networks, Understanding Complex Systems, Springer-Verlag Berlin Heidelberg, 2013.

[11] T. Viard, M. Latapy, C. Magnien, Computing maximal cliques in link streams, Theoretical Computer Science 609 (2016) 245252.

[12] M. Latapy, T. Viard, C. Magnien, Stream Graphs and Link Streams for the Modeling of Interactions over Time, arXiv:1710.04073.

[13] S. P. Borgatti, M. G. Everett, Regular blockmodels of multiway, multimode matrices, Social Networks 14 (1) (1992) 91–120.

[14] N. Narmadha, R. Rathipriya, Triclustering: An Evolution of Clustering, in: Proceedings of the Online International Conference on Green Engineering and Technologies (IC-GET'16), 2016, pp. 1–4.

[15] R. Guigourès, M. Boullé, F. Rossi, A Triclustering Approach for Time Evolving Graphs, in: Co-clustering and Applications International Conference on Data Mining Workshop, IEEE, Brussels, Belgium, 2012, pp. 115–122.

[16] F. Lorrain, H. C. White, Structural equivalence of individuals in social networks, The Journal of Mathematical Sociology 1 (1) (1971) 49–80.

[17] E. Balas, M. W. Padberg, Set Partitioning: A Survey, SIAM Review 18 (4) (1976) 710–760.

[18] R. Lamarche-Perrin, Y. Demazeau, J.-M. Vincent, A Generic Algorithmic Framework to Solve Special Versions of the Set Partitioning Problem, in: A. Andreou, G. A. Papadopoulos (Eds.), Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI'14), IEEE Computer Society, 2014, pp. 891–897.

[19] D. Dosimont, R. Lamarche-Perrin, L. M. Schnorr, G. Huard, J.-M. Vincent, A Spatiotemporal Data Aggregation Technique for Performance Analysis of Large-scale Execution Traces, in: M. S. Pérez, G. Antoniu, K. Keahey (Eds.), Proceedings of the 2014 IEEE International Conference on Cluster Computing (CLUSTER'14), IEEE Computer Society, 2014, pp. 149–157.

[20] V. Batagelj, A. Ferligoj, P. Doreian, Direct and indirect methods for structural equivalence, Social Networks 14 (1) (1992) 63–90, special Issue on Blockmodels.

[21] R. A. Hanneman, M. Riddle, Introduction to social network methods, University of California, Riverside, CA, 2005.

[22] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75–174.

[23] I. S. Dhillon, S. Mallela, D. S. Modha, Information-theoretic Co-clustering, in: Proceedings of the Ninth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'03), ACM, New York, NY, USA, 2003, pp. 89–98.

[24] R. Lamarche-Perrin, Y. Demazeau, J.-M. Vincent, Building Optimal Macroscopic Representations of Complex Multi-agent Systems. Application to the Spatial and Temporal Analysis of International Relations through News Aggregation, in: N. T. Nguyen, R. Kowalczyk, J. M. Corchado, J. Bajo (Eds.), Transactions on Computational Collective Intelligence, Vol. XV of LNCS 8670, Springer-Verlag Berlin, Heidelberg, 2014, pp. 1–27.

[25] R. v. d. Hofstad, Configuration Model, Vol. 1, Cambridge University Press, 2016, Ch. III.7, p. 216255.

[26] S. Kullback, R. A. Leibler, On Information and Sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

[27] T. M. Cover, J. A. Thomas, Elements of Information Theory, 2nd Edition, John Wiley & Sons, Inc., Hoboken, NJ, 2006.

[28] H. He, A. K. Singh, Closure-Tree: An Index Structure for Graph Queries, in: 22nd International Conference on Data Engineering (ICDE'06), 2006, pp. 38–38.

[29] T. Feder, R. Motwani, Clique Partitions, Graph Compression and Speeding-Up Algorithms, Journal of Computer and System Sciences 51 (2) (1995) 261–272.

[30] C. Hernández, G. Navarro, Compressed Representation of Web and Social Networks via Dense Subgraphs, in: L. Calderón-Benavides, C. González-Caro, E. Chávez, N. Ziviani (Eds.), String Processing and Information Retrieval, Vol. 7608 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 264–276.

[31] M. Dhabu, P. S. Deshpande, S. Vishwakarma, Partition based graph compression, International Journal of Advanced Computer Science and Applications 4 (9) (2013) 7–12.

[32] D. M. Roy, T. Y. Whye, The Mondrian Process, in: Advances in Neural Information Processing Systems, Vol. 21, Curran Associates, Inc., 2009, pp. 1377–1384.

[33] R. Lamarche-Perrin, Y. Demazeau, J.-M. Vincent, A Generic Algorithmic Framework to Solve Special Versions of the Set Partiioning Problem, Tech. Rep. MIS-Preprint 105/2014, Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany (2014).

[34] T. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohmé, Coalition structure generation with worst case guarantees, Artificial Intelligence 111 (1-2) (1999) 209–238.

[35] T. Rahwan, N. R. Jennings, Coalition Structure Generation: Dynamic Programming Meets Anytime Optimisation, in: Proceedings of the Twenty-third Conference on Artificial Intelligence, AAAI, 2008, pp. 156–161.

[36] T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, Artificial Intelligence 135 (2002) 1–54.

[37] P. Pons, M. Latapy, Post-processing hierarchical community structures: Quality improvements and multi-scale view, Theoretical Computer Science 412 (8-10) (2011) 892–900. doi:http://dx.doi.org/10.1016/j.tcs.2010.11.041.

[38] B. Cloitre, Sequence A003095, in: The On-Line Encyclopedia of Integer Sequences, `http://oeis.org/A003095`, 2002.

[39] G. McGarvey, Sequence A135361, in: The On-Line Encyclopedia of Integer Sequences, `http://oeis.org/A135361`, 2007.

[40] S. Anily, A. Federgruen, Structured Partitioning Problems, Operations Research 39 (1) (1991) 130–149.

[41] B. Jackson, J.D. Scargle, D. Barnes, S. Arabhi, A. Alt, *et al.*, An algorithm for optimal partitioning of data on an interval, IEEE Signal Processing Letters 12 (2) (2005) 105–108.

[42] G. Pagano, D. Dosimont, G. Huard, V. Marangozova-Martin, J.-M. Vincent, Trace Management and Analysis for Embedded Systems, in: Proceedings of the 7th International Symposium on Embedded Multicore SoCs (MCSoC'13), IEEE Computer Society Press, 2013, pp. 119–122.

[43] M. H. Rothkopf, A. Pekeč, R. M. Harstad, Computationally Manageable Combinational Auctions, Management Science 44 (8) (1998) 1131–1147.

[44] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Inc., 1993, Ch. Lagrangian Relaxation and Network Optimization, pp. 598–648.

[45] A. K. Chakravarty, J. B. Orlin, U. G. Rothblum, A partitioning problem with additive objective with an application to optimal inventory groupings for joint replenishment, Operations Research 30 (5) (1982) 1018–1022.

[46] R. V. V. Vidal, Optimal Partition of an Interval – The Discrete Version, in: R. V. V. Vidal (Ed.), Applied Simulated Annealing, Vol. 396 of Lecture Notes in Economics and Mathematical Systems, Springer Berlin Heidelberg, 1993, pp. 291–312.

[47] B. Davey, H. Priestley, Introduction to Lattices and Order, 2nd Edition, Cambridge University Press, (2002).

# Multidimensional Outlier Detection in Temporal Interaction Networks: An Application to Political Communication on Twitter

Audrey Wilmet*, Robin Lamarche-Perrin†
*Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
†Institut des Systèmes Complexes de Paris Île-de-France, ISC-PIF, UPS 3611, Paris, France
Email: firstname.lastname@lip6.fr

***Abstract*—In social network Twitter, users can interact with each other and spread information via retweets. These millions of interactions may result in media events whose influence goes beyond Twitter framework. In this paper, we thoroughly explore interactions to provide a better understanding of the emergence of certain trends. First, we consider an interaction on Twitter to be a triplet $(s,a,t)$ meaning that user $s$, called the spreader, has retweeted a tweet of user $a$, called the author, at time $t$. We model this set of interactions as a data cube with three dimensions: spreaders, authors and time. Then, we provide a method which builds different contexts, where a context is a set of features characterizing the circumstances of an event. Finally, these contexts allow us to find relevant unexpected behaviors, according to several dimensions and various perspectives: a user during a given hour which is abnormal compared to its usual behavior, a relationship between two users which is abnormal compared to all other relationships, *etc.* We apply our method to a set of retweets related to the 2017 French presidential election and show that one can build interesting insights regarding political organization on Twitter.**

## I. Introduction

The use of social networks has exploded over the past fifteen years. The micro-blogging service Twitter is currently the most popular and fastest-growing one of them. Within this social network, users can post information via tweets as well as spread information by retweeting tweets of other users. This leads to a dissemination of information from a variety of perspectives, thus affecting users ideas and opinions.

As discussed in the works of Murthy et al. [19] and Weller et al. [32], for some of the most active users, Twitter even constitute the primary medium by which they get informed. These users only represent a negligible fraction of the population. Nevertheless, hot topics emerging on Twitter's data stream are relayed by traditional media and therefore reach a much broader audience. If such trends often naturally arise from discussions or are consequences of the reaction of all users to real-world events, they may also be originated by the intensive activity of a small group and mislead other users on the significance of certain topics.

The volume of user-generated data is considerable: over 500 millions of tweets are posted every day on Twitter. Moreover, this data results from interactions of millions of users over time and therefore includes numerous complex structures. In this context, it is difficult for users to have a concrete vision of trends taking place and, even more, to apprehend the way in which all interactions are organised and can lead to media events.

In this paper, we seek to make this task achievable. More precisely, we aim at finding outliers in interaction data formed from a set of retweets. For instance, an event in a data stream is an outlier: it can be view as a statistical deviation of the total number of retweets at a given point in time. More generally, outliers, depending on which dimensions define them, highlight instants, users, users during given periods, or interactions for which the retweeting process behave unusually. Therefore, they constitute important information which is worth noticing from the perspective of the user. In order to find these unexpected behaviors, we design a multidimensional and multilevel analysis method.

First of all, we consider an interaction on Twitter to be a triplet $(s,a,t)$ meaning that user $s$, called the spreader, has retweeted a tweet of user $a$, called the author, at time $t$. We model the set of interactions as a data cube with three dimensions: spreaders, authors and time. This representation enables us to access local information, that is the number of retweets between two users during a specific hour, as well as more global and aggregated information, as for instance, the total number of retweets during a given hour. Afterwards, we combine and compare these different quantities of interactions between them in order to find outliers according to different contexts. Using the two previous quantities, we could, for instance, find an unexpected relationship between a spreader $s$ and an author $a$ during an hour $h$, if the number of retweet from $s$ to $a$ during $h$ is significantly large *given* the total number of retweets observed during this hour. This analysis gives us insight into the possible reasons why some events emerge more than others and, in particular, whether they are global phenomena or, on the contrary, whether they originate from specific actors only.

Our method applies to all types of temporal interaction networks. One can add attributes to interactions by adding dimensions to the problem. In this paper, we add a semantic dimension referring to tweet contents by considering the 4-tuples $(s,a,k,t)$ meaning that $s$ retweeted a tweet written by $a$ and containing hashtag $k$ at time $t$. This allows us to explore interactions from other perspectives and gain crucial information on events taking place.

The paper is organized as follows. We review the related work on outlier detection within Twitter in Section II. We introduce the modelling of interactions as a data cube in Section III, then we describe our method to build relevant contexts in Section IV. After describing our datasets in Section V, we present a case study in Section VI. In particular, we investigate the causes of emergence of events found in the temporal dimension by exploring authors, spreaders, then hashtags dimensions. In Section VII, we discuss two future works that can be achieved using our method, in particular, a characterization of the second screen usage and a user-topic link prediction. Finally, we conclude the paper in Section VIII.

## II. Related Work

The problem of outlier detection on Twitter has attracted a significant amount of interest among scientists and has been approached in various ways depending on how outliers are defined and on the techniques used.

Some researchers consider outliers as real-world events happening at a given place and at a given moment. For example, Sakaki et al. [25] and Bruns et al. [2] trace specific *keywords* attributed to a real-world event and find such outliers by monitoring temporal changes in word usage within tweets. There are also methods based on tweet clustering. In these approaches, authors infer, from timestamps, geo-localizations and tweet contents, a similarity between each pair of tweet and find real-world events into clusters of similar tweets. These techniques include the one of Dong et al. [10], which computes similarities with a wavelet-based method between time series of keywords; the one of Li et al. [18] which aims at finding crime and disaster related events in a real time fashion; and the one of Walther et al. [31] which focuses on small scale events located in space.

Other researchers, instead, seek entities like bots, spammers, hateful users or influential *users*. Thus, they consider outliers as users with abnormal behaviors according to different criteria. Varol et al. [30] detect bots by means of a supervised machine learning technique. They extract features related to user activities along time, user friendships as well as tweet contents and use these features to identify bots by means of a labelled dataset. Stieglitz et al. [28] identify influential users by investigating the correlation between the vocabulary they use in tweets and the number of time they are retweeted. Ribeiro et al. [24] detect hateful users. They start by classifying users with a lexicon-based method and then show that hateful users differ from normal ones in terms of their activity patterns and network structure.

Finally, other works aim at finding privileged *relationships* between users. Among those, the work of Wong et al. [34] apply it to political leaning by combining an analysis of the number of retweets between two users with a sentiment analysis on the retweeted tweets.

All these works, although providing meaningful results, use different methods for different kind of outliers. Moreover, they only consider one perspective in the way they define them. With our approach, we want to treat these different types of outliers – *keywords, users, relationships* – in a unified way as well as to consider different contexts in which outliers are considered abnormal. Hence, not only we consider different entities as abnormal users; abnormal relationships; abnormal behaviors of users during specific hours, *etc.*, but also different contexts in which outliers are defined. Thus, an abnormal user may be abnormal during a given hour compared to the way it usually behaves during other hours, but also compared to the behavior of all other users during the same hour. In this way, our framework aims to give a more complete and systematic picture of how users act, interact, and are organized along time in a way similar to what Grasland et al. [15] do in the case of media coverage in newspapers.

In practice, instead of characterizing and detecting outliers using tweets'content, as a lot of current approaches do, included those set out above, we focus on the volume and structure of interactions. Indeed, text-mining techniques face challenges as the ambiguity of the language and the fact that resultant models are language-dependent and topic-dependent. Moreover, the structure of communication alone is already quite informative. Other authors point into this direction. For instance Chavoshi et al. [5] use a similar technique to the one of Varol et al. [30], but only exploit user activities through their number of tweets and retweets. In the same idea, Chierichetti et al. [6] look at the tweet/retweet volume and detect points in time when important events happen. Instead of volume-based features, another alternative to text-mining techniques is to use graph-based features. Song et al. [27], for instance, identify spammers in real time with a measure of distance and connectivity between users in the directed friendship graph (followers and followees). Bild et al. [1] designed a similar method but based on the retweet graph instead. Also based on the retweet graph, the method of Ten et al. [29] detects trends by noticing changes in the size and in the density of the largest connected component. Another example is the approach of Coletto et al. [7] which combines an analysis of the friendship graph and of the retweet graph to identify controversies in threads of discussion.

In this paper, we design a method able to handle multiple types of outliers by observing the retweets' volume in numerous different contexts. We believe that this multidimensional and multilevel analysis is essential to detect subtle unexpected behaviors as well as fully understand the way in which millions of interactions may result in media events.

## III. Formalism

We denote the set of interactions by a set $E$ of triplets such that $(s,a,t) \in E$ indicates that user $s$, called the spreader, has retweeted a tweet written by user $a$, called the author, at time $t$. We represent this set of interactions by a data cube [16]. In this section, we formally define this tool as well as the possible operations we can perform to manipulate data.

### A. Data Cube Definition

A data cube is a general term used to refer to a multidimensional array of values [16]. Given $N$ dimensions characterized by $N$ sets $X_1, \dots, X_N$, we can built $\sum_{i=0}^{N} \binom{N}{N-i}$ data cubes, each representing a different degree of aggregation of data. The quantity $\binom{N}{N-i}$ corresponds to the number of data cubes of dimension $N-i$ in which $i$ dimensions are aggregated. Within this set of data cubes, we call the base cuboid $\mathscr{C}_{base}$ the $N$-dimensional data cube which has the lowest degree of aggregation. More generally, a $n$-dimensional data cube is denoted $\mathscr{C}_n(X, f)$ where $X = X_1 \times \dots \times X_n$ is the Cartesian product of the $n$ sets $X_1, \dots, X_n$, and $f$ is a feature which maps each $n$-tuple to a value in a value space $W$:

$$\begin{aligned} f: \quad & X \longrightarrow W \\ (x_1, \dots, x_n) & \longmapsto f(x_1, \dots, x_n) \, . \end{aligned}$$

In the following, $n$-tuples are also called cells of the cube and denoted $x$ such that $x = (x_1, \dots, x_n) \in X$.

*Dimensions* are the sets of entities with respect to which we want to study data. As a first step, we can consider three dimensions: spreaders, denoted $S$, authors, denoted $A$, and time, denoted $T$. In addition, we can organise elements of a dimension into sub-dimensions. For instance, the temporal dimension can be organised depending on temporal granularity. In our case, we divide it into the two sub-dimensions days, denoted $D$, and hours of the day, denoted $H$, such that $t = (d, h)$ denotes the hour $h$ of day $d$, with $(d, h) \in D \times H$. While the set of days $D$ depends on the dataset, $H$ is the set of hours of the day such that $H = \{0, \cdots, 23\}$.

*The feature* is a numerical measure which provides the quantities according to which we want to analyse relationships between dimensions. Here, we consider the quantity of interaction, denoted $v$. It gives the number of retweets for any combination of the four dimensions. In the base cuboid $\mathscr{C}_{base} = \mathscr{C}_4(S \times A \times D \times H, v)$, $v$ gives the number of times $s$ retweeted $a$ during hour $h$ of day $d$ (see Figure 1):

$$v: \; S \times A \times D \times H \longrightarrow \mathbb{N} \, .$$

Data cubes of smaller dimensions are obtained by aggregating the base cuboid along one or several dimensions. We discuss this operation along with others in the next subsection.

### B. Data Cube Operations

We can explore the data through three operations called aggregation, expansion and filtering.

*Aggregation* is the operation which consists in seeing information at a more global level. Given a data cube $\mathscr{C}_n(X, f)$, the aggregation operation along dimension $X_i$ leads to a data cube of dimension $n-1$, $\mathscr{C}_{n-1}(X', f)$ where $X' = X_1 \times \dots \times X_{i-1} \times X_{i+1} \times \dots \times X_n$. Formally, a dimension $X_i$ is aggregated by adding up values of the feature for all elements $x_i \in X_i$. We indicate by a $\cdot$ the dimension which is aggregated with respect

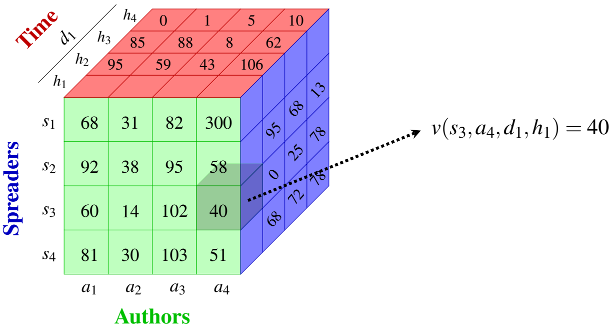


Fig. 1: **Base Cuboid** $\mathscr{C}_4(S \times A \times D \times H, v)$ **–** The base cuboid is not aggregated along any of its dimensions. It contains local information with respect to the quantity of interaction $v$. For instance, the gray cell indicates that $s_3$ retweeted $a_4$ 40 times on day $d_1$ at hour $h_1$.

to $f$. Hence, $\mathscr{C}_{n-1}(X', f)$ is constituted of $n-1$-dimensional cells denoted $x' = (x_1, \dots, x_{i-1}, \cdot, x_{i+1}, \dots, x_n) \in X'$ where

$$f(x') = \sum_{x_i \in X_i} f(x) \quad .$$

For instance, one can aggregate along the dimension of hours of the day such that

$$v(s, a, d, \cdot) = \sum_{h \in H} v(s, a, d, h)$$

gives the total number of time $s$ retweeted $a$ during day $d$. Opposed to the base cuboid, the apex cuboid is the most summarized cuboid. It is aggregated along all dimensions and hence consists in only one cell containing the grand total $f(\cdot, \dots, \cdot) = \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} f(x)$. In our case, the apex cuboid contains the total number of retweets.

We can also aggregate interactions according to a set of subsets of dimension $X_i$. Let $P_i$ denote this partition such that the intersection of any two distinct sets in $P_i$ is empty and the union of the sets in $P_i$ is equal to $X_i$. Then, given a data cube $\mathscr{C}_n(X, f)$, the aggregation operation along $P_i$ leads to a data cube $\mathscr{C}_n(X', f)$ with $X' = X_1 \times \dots \times X_{i-1} \times P_i \times X_{i+1} \times \dots \times X_n$. This cube is constituted of $n$-dimensional cells denoted $x = (x_1, \dots, x_{i-1}, C_k, x_{i+1}, \dots, x_n) \in X'$, with $C_k \in P_i$, such that

$$f(x') = \sum_{x_i \in C_k} f(x) \quad .$$

For instance, one can aggregate according to a partition of hours $P_H = \{H_N, H_D\}$, where $H_N$ is the set of nocturnal hours and $H_D$ the set of daytime hours such that

$$v(s, a, d, H_N) = \sum_{h' \in H_N} v(s, a, d, h')$$

in $\mathscr{C}_4(S \times A \times D \times P_H)$, gives the total number of time $s$ retweeted $a$ during nocturnal hours on day $d$.

*Expansion* is the reverse operation which consists in seeing information at a more local level by introducing additional dimensions. Given a data cube $\mathscr{C}_n(X, f)$, the expansion operation on dimension $X_{n+1}$ leads to a data cube of dimension $n+1$, $\mathscr{C}_{n+1}(X', f)$ where $X' = X \times X_{n+1}$.

*Filtering* is the operation which consists in focusing on one specific subset of data. Given a data cube $\mathscr{C}_n(X, f)$, the filtering operation leads to a sub-cube $\mathscr{C}_n(X', f)$ by selecting subsets of elements within one or more dimensions such that $X' = X'_1 \times ... \times X'_n$ with $X'_1 \subseteq X_1, ..., X'_n \subseteq X_n$.

It is also possible to combine operations together. For instance, we can filter the data cube aggregated on the partition of hours, $\mathscr{C}_4(S \times A \times D \times P_H, v)$, in order to focus on spreaders that abnormally retweet authors overnight on a given day. Note that the resulting data cube $\mathscr{C}_4(S \times A \times D \times \{H_N\}, v)$ is different from the data cube $\mathscr{C}_4(S \times A \times D \times H_N, v)$: in the first case, a cell $(s, a, d, H_N)$ gives the total number of time $s$ retweeted $a$ during nocturnal hours on day $d$; while in the second case, a cell $(s, a, d, h)$ give the number of times $s$ retweeted $a$ during hour $(d, h)$ where $h \in H_N$ is a nocturnal hour.

Figure 2 shows a set of all data cubes that can be obtained considering the three dimensions: spreaders, authors and time. It also illustrates how to navigate from one to another thanks to the previously described operations.

## IV. Method

In this paper, our goal is to find abnormal data cube cells, *i.e.*, entities $x \in X$ for which the observation $f(x)$ is abnormal. As an observation' abnormality is relative to the elements to which it is compared [17], a given cell may be abnormal or not depending on the *context*. The context, denoted $\mathscr{C}$, is the set of elements which are taken into account in order to assess the abnormality of an entity $x \in X$. In this section, we design a set of steps in order to shape various contexts and show, through several examples, that it leads to a deeper exploration of interactions compared to an elementary outlier detection.

### A. Construction of a Context

An abnormal entity $x \in X$ is an entity which behavior deviates from its expected one. Hence, one way to find outliers in a set of entities $x \in X$ is to consider the following elements:
– a set of observed values $\mathscr{O} = \{f(x), x \in X\}$;
– a set of expected values $\mathscr{E} = \{f_{exp}(x), x \in X\}$;
– a set of deviation values $\mathscr{D} = \{d(f(x), f_{exp}(x)), x \in X\}$, which quantify the differences between observed and expected values.
Together, these elements constitute the context $\mathscr{C}$. Then, given $\mathscr{C}$, an outlier $x \in X$ is a point whose absolute deviation value, $|d(f(x), f_{exp}(x))|$, is significantly larger than most others deviation values.

We build more or less elaborate contexts by playing with the considered observed, expected and deviation values.

### B. Observed values

According to the type of unexpected behaviors we are looking for, the first step consists in choosing a cube among the set of cubes obtained from the base cuboid using one or several operations. This cube, denoted $\mathscr{C}_{obs}$, constitutes the set of entities and observed values.

For instance, we can look for abnormal authors at given hours. To to so, we focus on the cube aggregated on spreaders such that $\mathscr{C}_{obs} = \mathscr{C}_3(A \times D \times H, v)$. We may also want to find abnormal authors during nocturnal hours only. In this case, we consider the aggregated and filtered data cube $\mathscr{C}_{obs} = \mathscr{C}_3(A \times D \times H_N, v)$.

In the first case, we consider all entities of the same type, $(a, d, h) \in A \times D \times H$: we are in a *global context*. On the contrary, when we only consider a subset of all entities, as in the second example with $(a, d, h) \in A \times D \times H_N$, we are in a *local context*.

### C. Expected values

Once the set of observed values is fixed, we build a model of expected behavior based on a combination of other data cubes $\mathscr{C}_m(X', f)$, called *comparison data cubes*. For the context to be relevant, these must derive from the aggregation of $\mathscr{C}_{obs} = \mathscr{C}_n(X, f)$ on one or more dimensions. Hence, $n > m$ and $X = X' \times Y$ where $Y$ is the Cartesian product of the aggregated dimensions. In the following, we build three different types of expected contexts: the basic, aggregative and multi-aggregative contexts.

*1) Basic Contexts:* When seeking abnormal cells within a data cube $\mathscr{C}_n(X, f)$, the most elementary context we can consider is the one in which the expected value is a constant, identical for each cell. We call it the *basic context*. The model of expected behavior is that interactions are uniformly distributed over cells. In this case, the comparison data cube is the apex cuboid $\mathscr{C}_0(\cdot, f)$ and the expected value is the average number of interactions per cell:

$$f_{exp}(x) = \frac{f(\cdot)}{|X|} \,.$$

*For instance, in data cube $\mathscr{C}_3(A \times D \times H, v)$, an abnormal cell $c^* = (a^*, d^*, h^*)$ indicates that during hour $h^*$ of day $d^*$, author $a^*$ has been retweeted an abnormal number of times compared to the average number of times any author is retweeted during any hour, $v_{exp}(a, d, h) = \frac{v(\cdot,\cdot,\cdot)}{|A \times D \times H|}$.*

*2) Aggregative Contexts:* To find more subtle and local outliers, expected values must be more specific to each cell. The process is the same as in the basic context except that the considered comparison cube $\mathscr{C}_m(X', f)$ is not aggregated over all dimensions of $X$, *i.e.* $X = X' \times Y$ with $Y \neq X$:

$$f_{exp}(x) = \frac{f(x')}{|Y|} \,,$$

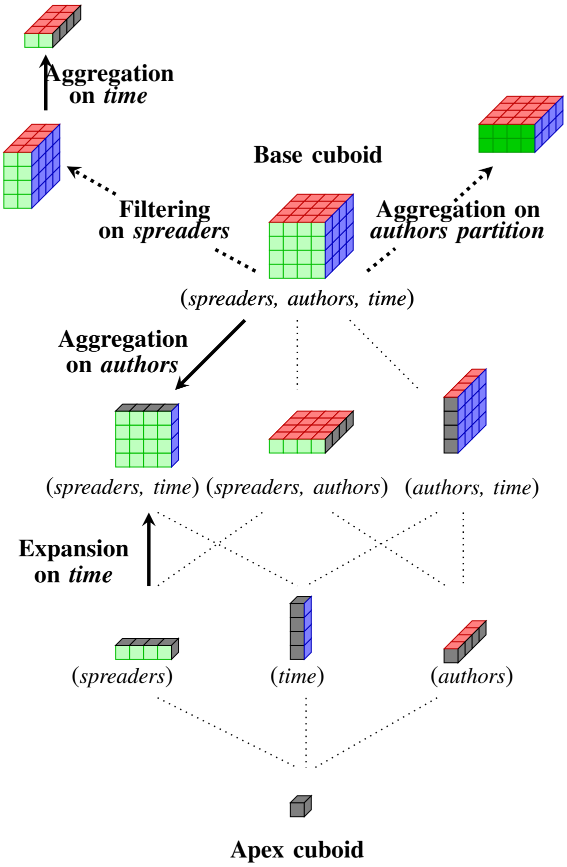


Fig. 2: **Set of data cubes obtained by considering the three dimensions: spreaders, authors and time.**

Each data cube models interactions under a particular perspective:
– we can move from one cuboid to another either by aggregation or expansion.
– we can aggregate on a partition. In the example (top-right cube), we aggregate the base cuboid over authors communities.
– we can focus on a particular subset by filtering a given data cube. In the example (top-left cube), we focus on two spreaders within the base cuboid.
– we can combine operations and aggregate a filtered data cube.

From top to bottom, we have access to more and more aggregated information, for instance:
– the $4D$ cell $x = (s_1, a_2, d_1, h_4)$ associated to the value $v(s_1, a_2, d_1, h_4) = 9$ means that $s_1$ has retweeted $a_2$ 9 times on day $d_1$ during hour $h_4$;
– the $3D$ cell $x = (a_2, d_1, h_4)$ associated to the value $v(\cdot, a_2, d_1, h_4) = 1,288$ means that $a_2$ has been retweeted $1,288$ times on day $d_1$ during hour $h_4$ (by all spreaders);
– the $1D$ cell $x = a_2$ associated to the value $v(\cdot, a_2, \cdot, \cdot) = 29,362$ means that $a_2$ has been retweeted $29,362$ times (in the whole dataset);
– the $0D$ cell $x = (\cdot, \cdot, \cdot, \cdot)$ associated to the value $v(\cdot, \cdot, \cdot, \cdot) = 1,142,004$ means that the total number of retweets is equal to $1,142,004$.

such that $x = (x', y) \in X' \times Y$. Defined as such, the expected value is the value that one should observe if all interactions on $X'$ were homogeneously distributed on dimensions $Y$. We call these contexts, *aggregative contexts*.

*For instance, in data cube $\mathscr{C}_3(A \times D \times H, v)$, relatively to data cube $\mathscr{C}_2(D \times H, v)$ and expected values*

$$v_{exp}(a, d, h) = \frac{v(\cdot, \cdot, d, h)}{|A|},$$

*such that $Y = A$ and $X' = D \times H$, an abnormal cell $c^* = (a^*, d^*, h^*)$ indicates a significant deviation between the number of retweets received by $a^*$ during hour $(d^*, h^*)$ and the one that should have been observed if all authors had received the same number of retweets during hour $(d^*, h^*)$.*

*3) Multi-aggregative Contexts:* Aggregative contexts assume that interactions are homogeneously distributed among dimensions $Y$. It is possible to create contexts which differentiate the repartition of interactions according to each cell activity. We call them *multi-aggregative contexts*. Unlike the other two, they require multiple comparison data cubes. There are no generic formulas: the number and types of comparison cubes as well as expected values depend on the application.

*For instance, if we take back the previous example, we can consider, instead, the following expected values:*

$$v_{exp}(a, d, h) = v(\cdot, \cdot, d, h) \times \frac{v(\cdot, a, \cdot, \cdot)}{v(\cdot, \cdot, \cdot, \cdot)}.$$

*This way, it is expected that the number of retweets during $(d, h)$ is distributed among authors proportionally to their mean activity. We can also add information on authors activity during specific hours, and consider the cubes $\mathscr{C}_2(D \times H, v)$, $\mathscr{C}_2(A \times H, v)$ and $\mathscr{C}_1(H, v)$, such that*

$$v_{exp}(a, d, h) = v(\cdot, \cdot, d, h) \times \frac{v(\cdot, a, \cdot, h)}{v(\cdot, \cdot, \cdot, h)}.$$

*In this context, an abnormal cell $c^* = (a^*, d^*, h^*)$ indicates a significant deviation between the number of retweets received by $a^*$ during hour $h^*$ of day $d^*$ and the one that should have been observed if $a^*$ had been retweeted the way it is used to during hour $h^*$ on other days.*

Each of these contexts can either be global or local depending on the chosen set of observed values within $\mathscr{C}_{obs}$.

### *D. Deviation values*

Finally, for each cell $x$ within $\mathscr{C}_{obs}$, we measure the deviation between the observed value $f(x)$ and its expected value $f_{exp}(x)$. In this paper, we use two different deviation functions: the ratio and the Poisson deviation.

The *ratio* between an observed value and an expected value is defined such that

$$d_r(f(x), f_{exp}(x)) = \frac{f(x)}{f_{exp}(x)}.$$

Note that this deviation function does not distinguish between $f(x) = 2$ and $f_{exp}(x) = 1$, on the one hand, and $f(x) = 2{,}000$ and $f_{exp}(x) = 1{,}000$, on the other hand.

To take into account the significance to which a value deviates, we define another deviation function: the *Poisson deviation*. Indeed, in the cases in which the feature consists in counting the number of interactions during a given period, as $v(x)$, it can be modelled by a Poisson counting process of intensity $f_{exp}$ [15], such that

$$\Pr(v(x) = k) = \frac{f_{exp}(x)^k e^{-f_{exp}(x)}}{k!}.$$

In this case, the Poisson deviation $d_p$ can be defined as follows. If $f(x) \le f_{exp}(x)$, we calculate the probability of observing a value $f(x)$ or less, knowing that we should have observed $f_{exp}(x)$ on average. This probability is the cumulative distribution function of a Poisson distribution with parameter $f_{exp}(x)$. Accordingly, we denote it $F_{f_{exp}}(f(x))$. Then, by symmetry, we define $d_p$ such that:

$$d_p(f(x), f_{exp}(x)) = \begin{cases} \log(F_{f_{exp}}(f(x)) \text{ if } f(x) \le f_{exp}(x), \\ -\log(\bar{F}_{f_{exp}}(f(x)) \text{ if } f(x) > f_{exp}(x), \end{cases}$$

where the logarithm is calculated for convenience in order to have a better range of values.

In both cases, most of observed values are expected to be similar to their corresponding expected values. Consequently, the distribution of $\mathscr{D}$ is expected to follow a normal distribution in which most values fluctuates around a mean: $\bar{d_r} = 1$ for the ratio and $\bar{d_p} = 0$ for the Poisson deviation. Outlying cells, instead, correspond to deviation values significantly distant from the mean[1].

### *E. Examples*

Figure 3 illustrates several situations in which we find different abnormal authors during given hours by considering different contexts and a ratio deviation function: – Triplet $(a_1, d_1, 19h)$ is abnormal in the global basic context: it has been retweeted 1,500 times (15% of a 10,000) which is higher than all other triplets.

– Triplet $(a_2, d_2, 19h)$ is abnormal in the global aggregative context: its proportion of retweet is 50% which is higher than all other triplets.

– Triplet $(a_1, d_n, 19h)$ is abnormal in the global multi-aggregative context: the deviation in the activity of $a_1$ with respect to its usual activity at $19h$ is higher than all other triplets.

– Triplet $(a_3, d_2, 19h)$ is abnormal in the local aggregative context: its proportion of retweet is higher than other triplets $(a, d, h)$ in which $a$ is not an influential author.

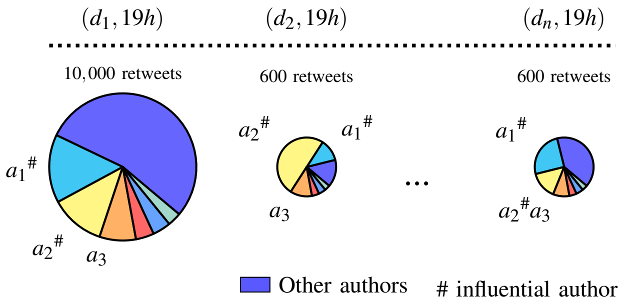


Fig. 3: **Different contexts lead to different outliers –** The numbers of retweets per hour distributed among authors are represented as pie charts.

As this example shows and as we will show in practice in the next sections, our approach, combining data cubes to build different contexts, leads to numerous kinds of outliers which allows us to thoroughly analyse temporal interactions under different perspectives.

## V. Datasets

In this paper, we choose to study the organization of interactions on Twitter by analysing different sets of politics-related retweets. Indeed, since Twitter is an integral part of means of communication used by political leaders to disseminate information to the public, finding abnormal entities corresponding to different kinds of unexpected behaviors in this situation is of great interest. To do so, we use two different datasets.

**Dataset** $D_1$ is a set of retweets related to political communication during the 2017 French presidential elections. We use a subset of the dataset collected by Gaumont et al. [12] as part of the *Politoscope* project. It contains politics-related retweets during the month of August 2016. Formally, our dataset consists in the set of retweets $E$, such that $(s, a, d,\ h) \in E$ means that $s$ retweeted $a$ at hour $h$ of day $d$, where either the corresponding tweet contains politics-related keywords, or $a$ belongs to a set of 3,700 French political actors listed by the Politoscope project. It contains 1,142,004 retweets and involves 211,155 different users. In this dataset, the set of days is $D = \{1, \cdots, 31\}$.

**Dataset** $D_2$ is the same as dataset $D_1$ except that it contains an additional dimension. It consists in the set of re-tweets $E$, such that $(s, a, k, d, h) \in E$ means that $s$ retweeted a tweet written by $a$ and containing the hashtag $k$ at hour $h$ of day $d$. It contains $|K| = 30{,}057$ different hashtags.

[1] We use the classical assumption that a value is anomalous if its distance to the mean exceeds three times the standard deviation [4], [16].

In the following, usernames are only mentioned when they correspond to official Twitter accounts of politicians, or public organizations, such as city halls, newspapers, or shows. Otherwise, they are designated by generic terms *user-n*, where $n$ is an integer to differentiate anonymous users.

## VI. Experiments

As a first illustration of our method, we present a case study which, based on events found in the temporal dimension, proposes possible causes of their emergence by exploring other dimensions. First, we apply our method on dataset $\mathscr{D}_1$ and focus on the three dimensions: spreaders, authors and time. Then, we add the hashtag dimension with dataset $\mathscr{D}_2$ in order to gain more insight on events.

### A. Events

We define an event $e = ((d_1^*, h_1^*), \cdots, (d_n^*, h_n^*)) \in \mathscr{E}$ to be a set of consecutive abnormal hours. For convenience, we denote it $e = (d^*, h_1^* \text{ - } h_n^*)$ when all hours span over the same day $d^*$.

Figure 4 shows the evolution of the number of retweets per hour[2]. We can distinguish three distinct behaviors:
– nocturnal hours, characterized by a number of retweets fluctuating around 350,
– daytime from the 1st of August to the 24th, characterized by a higher number of retweets fluctuating around 1,700,
– daytime from the 24th of August to the 31st, characterized by a global increase in the number of retweets which fluctuates around 2,900.

*1) Basic Context:* First of all, we look for events in the basic context. The sets of entities and observed values are provided by data cube $\mathscr{C}_2(D \times H, v)$. Expected values are defined such that

$$v_{exp}^{b}(d,h) = \frac{v(\cdot,\cdot,\cdot,\cdot)}{|D \times H|}\,.$$

Figure 5 (Left) shows the distribution of deviation values by considering a ratio-based deviation. We find seven abnormal hours leading to three events such that

$$\mathscr{E} = \{(24, 20h \text{ - } 22h), (25, 19h), (28, 14h \text{ - } 15h)\}\,.$$

We see that these hours correspond to the three peaks of activity on Figure 4. Hence, this context does not highlight local anomalies but only global ones, deviating from all observations. Therefore, it is biased by circadian and weekly rhythms and does not have access to abnormal nocturnal hours nor hours located during the first part of the month.

*2) Aggregative Context:* To take into account the overall increase in the number of retweets during the month, we need to use a aggregative context in which expected values incorporate the overall activity of the day provided by data cube $\mathscr{C}_1(D, v)$:

$$v_{exp}^{a}(d,h) = \frac{v(\cdot,\cdot,d,\cdot)}{|H|}\,.$$

As such, deviation values are independent of daily variations in the data. This is what we observe in Figure 5 (Center). We find 10 abnormal hours. Among those, six hours are part of the first period of the month: the 3rd at $11h$, the 12th at $23h$, the 21st at $21h$, and the 22th from $17h$ to $20h$. Nevertheless, extreme values are still biased by circadian rhythms which prevent us from detecting abnormal nocturnal hours.

*3) Multi-aggregative Context:* To address this issue, we use a multi-aggregative context in which we add aggregated information relating to the typical activity per hour, provided by data cubes $\mathscr{C}_1(H, v)$ and $\mathscr{C}_0(\cdot, v)$:

$$v_{exp}^{m-a}(d,h) = v(\cdot,\cdot,d,\cdot) \times \frac{v(\cdot,\cdot,\cdot,h)}{v(\cdot,\cdot,\cdot,\cdot)}\,.$$

Moreover, we take the Poisson distance as a deviation measure to account for the significance of deviations. We find 40 abnormal hours (see Figure 5 (Right)). Among those, several are adjacent, which leads to 17 distinct events (see Table I).

Hour $(11^{th}, 0h)$ is abnormal. It means that, on average, at $0h$, we expect to observe $v(\cdot,\cdot,\cdot,0h)/v(\cdot,\cdot,\cdot,\cdot) = 3.16\%$ of the total number of retweets of the day. Hence, on hour $(11^{th}, 0h)$, we expect to observe $v(\cdot,\cdot,11^{th},\cdot) \times 3.16\% = 909$ retweets. However, we observe $1,418$ retweets in $\mathscr{C}_2(D \times H, v)$. This deviation from the expected value is much more important than those observed for most hours $(d,h) \in D \times H$. As a consequence, $(11^{th}, 0h)$ is an abnormal hour in this particular multi-aggregative context.

In Table I, we see several hours of generally low activity as nocturnal hours. This last result shows that using more sophisticated contexts leads to more subtle outliers.

### B. Abnormal authors during events

Now, we focus on determining whether an abnormal event is due to specific authors which have been retweeted predominantly, or, on the contrary, results from a more global phenomenon in which we observe an overall increase of the activity.

To do so, we use a local and multi-aggregative context. Observed values are provided by the filtered and aggregated data cube $\mathscr{C}_3(A \times \{e\}, v)$, where $e \in \mathscr{E}$ is an abnormal event. A cell $(a, e)$ within this cube gives the total number of times author $a$ has been retweeted during event $e$. This way, we focus on how interactions are organized among authors within each event.

We proceed in a similar way to obtain expected values. Instead of considering the set of authors during event $e$, we consider the set of authors during each of the hourly periods corresponding to $e$ on all days. We denoted this set of hours $H_e = \{h^* \in H \,|\, (d^*, h^*) \in e\}$. We focus on data cube $\mathscr{C}_3(A \times D \times P_H, v)$, aggregated on the partition of $H$, $P_H = \{H_e\}$. Operations performed to switch from the original

[2] Note that due to a server failure from Tuesday the 9th to Thursday the 11th, no activity is observed during this period.

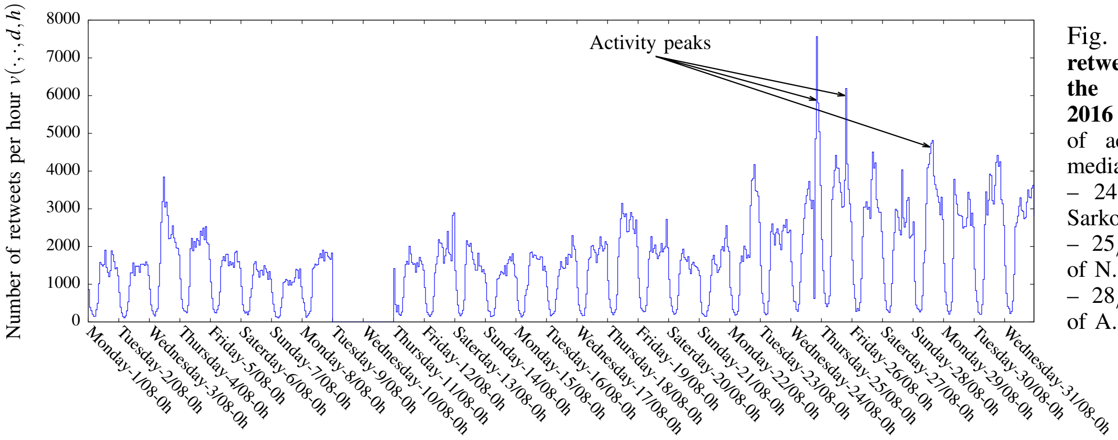


Fig. 4: **Number of retweets per hour along the month of August 2016 –** The three peaks of activity correspond to media events:
– 24/08: interview of N. Sarkozy on television news,
– 25/08: political meeting of N. Sarkozy,
– 28/08: political meeting of A. Juppé.

| Events | Significant Abnormal Authors | Media Events |
|---|---|---|
| $(3^{th}, 10h$ - $13h)$ | several | Police intervention in a church |
| $(11^{th}, 0h)$ | marseille | Fire in the city of marseille |
| $(11^{th}, 3h)$ | FrancoisFillon | Unknown |
| $((12^{th}, 22h), \cdots, (13^{th}, 1h))$ | fhollande | Olympic victory of France |
| $(13^{th}, 9h)$ | none | Unknown |
| $(19^{th}, 22h)$ | none | Olympic victory of France |
| $(21^{th}, 21h)$ | none | Olympic victory of France |
| $(22^{th}, 16h$ - $22h)$ | several | Announcement of N. Sarkozy's campaign |
| $(23^{th}, 7h$ - $8h)$ | none | Unknown |
| $(24^{th}, 20h$ - $22h)$ | several | Interview of N. Sarkozy on television news |
| $(25^{th}, 19h)$ | NicolasSarkozy | Political Meeting of N. Sarkozy |
| $(26^{th}, 15h$ - $18h)$ | several | Council of state on burkini wearing |
| $(27^{th}, 15h)$ | alainjuppe | Political Meeting of A. Juppé |
| $(28^{th}, 0h)$ | several | Interview of N. Kosciusko-Morizet on a talk-show |
| $(28^{th}, 13h$ - $15h)$ | JLMelenchon | Political Meeting of J-L. Mélenchon |
| $(29^{th}, 7h$ - $9h)$ | NicolasSarkozy | Interview of N. Sarkozy on a radio program |
| $(30^{th}, 17h$ - $18h)$ | none | Resignation of E. Macron from government |

TABLE I: **List of detected abnormal events and authors together with their associated media events.**

cube $\mathscr{C}_3(A \times D \times H, v)$ to data cube $\mathscr{C}_3(A \times D \times \{H_e\}, v)$ is depicted in Figure 6.

Finally, expected values are defined using the comparison data cubes $\mathscr{C}_2(A \times \{H_e\}, v)$ and $\mathscr{C}_1(\{H_e\}, v)$, obtained by aggregation of $\mathscr{C}_3(A \times D \times \{H_e\}, v)$, and data cube $\mathscr{C}_2(\{e\}, v)$, obtained by aggregation and filtering of $\mathscr{C}_3(A \times D \times \{H_e\}, v)$:

$$v_{exp}(a,e) = v(\cdot,\cdot,e) \times \frac{v(\cdot,a,\cdot,H_e)}{v(\cdot,\cdot,\cdot,H_e)},$$

where $v(\cdot,\cdot,e) = \sum_{(d^*,h^*) \in e} v(\cdot,\cdot,d^*,h^*)$, is the number of retweets observed during $e$; $v(\cdot,a,\cdot,H_e)$ is the total number of retweets author $a$ received during hours of $H_e$; and $v(\cdot,\cdot,\cdot,H_e)$ is the total number of retweets observed during $H_e$.

According to this context, a couple $(a^*, e) \in A \times \{e\}$ is abnormal when there is a significant deviation between the number of retweets received by $a$ during $e$, and the number of retweets $a$ is expected to receive on average during the corresponding period on other days. In the following, we discuss the three different situations which arise through specific examples.

### *1) One main author*

Figure 7 (Left) displays the distribution of deviation values for event $e = (29^{th}, 7h$ - $9h)$. Most observations $d \in \mathscr{D}$

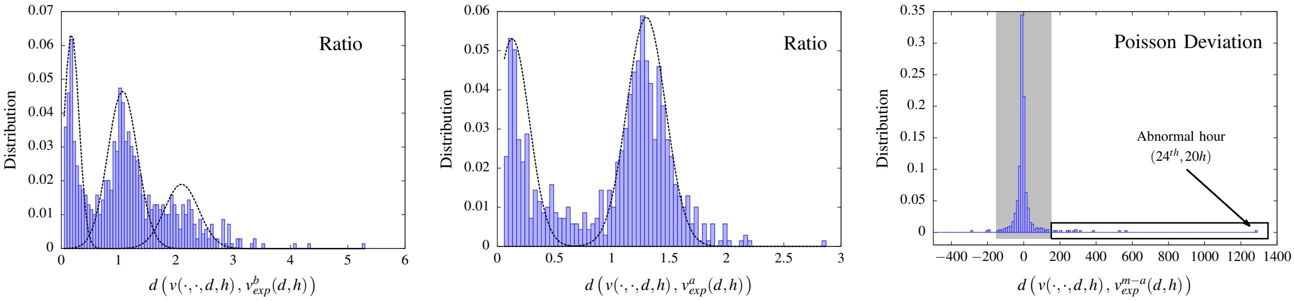


Fig. 5: **Deviation values of hours in basic, aggregative, and multi-aggregative contexts –** (Left) Basic – The three bell curves correspond to three distinct behaviors: nocturnal hours ($\bar{d_r^1} = 0.2$), daytime from the 1st of August to the 24th ($\bar{d_r^2} = 1.1$) and daytime from the 24th of August to the 31th ($\bar{d_r^3} = 2.1$). (Center) Aggregative – We only observe two behaviors, the one corresponding to nocturnal hours which fluctuates around $\bar{d_r^1} = 0.13$ and the one corresponding to daytime which fluctuates around $\bar{d_r^2} = 1.3$. (Right) Multi-aggregative – Most deviation values are centred on $\bar{d_p} = 0$ (gray zone), meaning that they are likely to be generated by a Poisson counting process with intensity $v_{exp}^{m-a}(d,h)$. Values far away from the mean represent hours which behave significantly differently compared to the way they should.

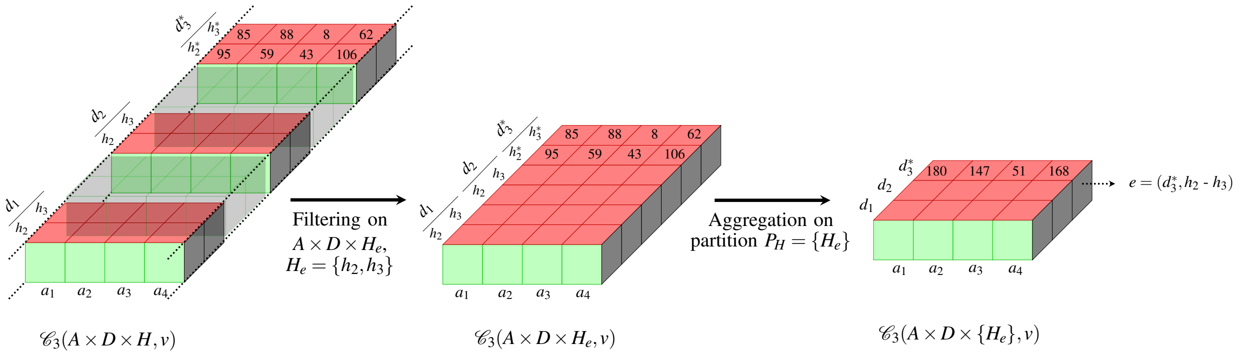


Fig. 6: **Local and multi-aggregative context to focus on authors during events –** To investigate the possible causes of the emergence of event $e = (d_3^*, h_2^* \text{ - } h_3^*)$, we characterize the authors' usual behaviors during the corresponding time periods on other days.

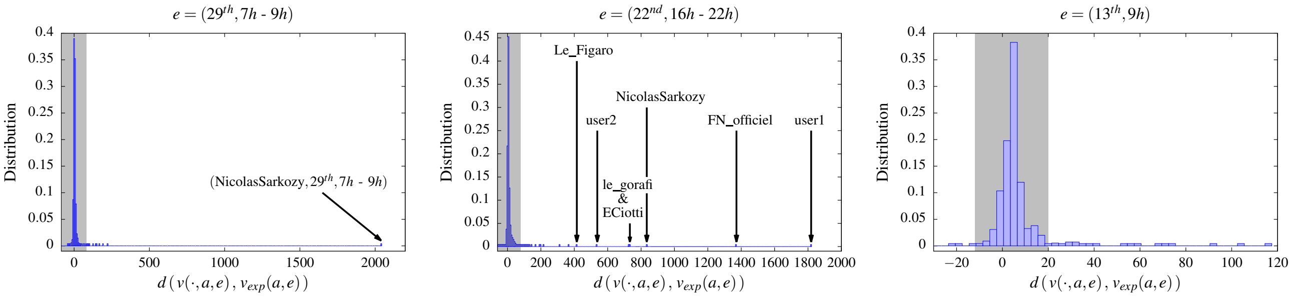


Fig. 7: **Abnormal authors during events -** (Left) Event ($29^{th}$, $7h$ - $9h$): we see that *NicolasSarkozy* is probably responsible for the observed event since its activity deviates significantly from its usual one. (Center) Event ($22^{nd}$, $16h$ - $22h$): the cause of this event is multiple, we see that several authors, mostly individuals and politicians from the right-wing, are more retweeted than they usually are. (Right) Event ($13^{th}$, $9h$): the distribution is more homogeneous, which makes this event a more global phenomenon.

follow a Gaussian distribution centred on $\bar{d_p} = 0$. We find 14 abnormal values. Among those, the one corresponding to $(\text{NicolasSarkozy}, 29^{th}, 7h \text{ - } 9h)$ significantly deviates from others. Indeed, in the considered context, we expect NicolasSarkozy to account for

$$\frac{v(\cdot, \text{NicolasSarkozy}, \cdot, \{7h, 8h, 9h\})}{v(\cdot, \cdot, \cdot, \{7h, 8h, 9h\})} = 2.2\%$$

of all retweets observed from $7h$ to $9h$. Thus, on the $29^{th}$ of August from $7h$ to $9h$, we expect him to be retweeted $v(\cdot, \cdot, (29^{th}, 7h \text{ - } 9h)) \times 2.2\% = 194$ times. Yet, he was retweeted $1,644$ times, which explain its large deviation value. Table I lists events with a similar distribution. In most cases, we observe that the corresponding media event is centred on the main author. For instance, they often indicate a political meeting of this author.

*2) Several main authors*

Figure 7 (Center) displays the distribution of the set of deviation values for event $e = (22^{nd}, 16h \text{ - } 22h)$. Once more, most observations $d \in \mathscr{D}$ follow a Gaussian distribution centred on $\bar{d_p} = 0$. We detect 42 outliers. Several values significantly deviates from the mean, indicating, this time, several main authors.

These events are not due to a single popular author, but to several authors, considerably retweeted. In contrast to the previous example, this suggest that they originate from the reaction of a few authors to some external fact in which they have an interest. This is what we observe in Table I: media events related to events with similar distributions are often indicative of situations according to which the main authors are not related, but on which they react. For example, the event on August the $3^{th}$, on the intervention of the police in a church, and the one on August the $26^{th}$, on burkini wearing, are media events intensely taken up by political members of right and extreme-right wings.

*3) No main authors*

Figure 7 (Right) displays the distribution of deviation values for event $e = (13^{th}, 9h)$. In opposition to previous examples, we see that values are more homogeneously distributed and spread over a smaller range.

The absence of significant outliers shows that these events are more global phenomena than the previous ones: they emerge because numerous authors are being retweeted instead of a few, intensely. This suggest that they originate from the reaction of a multitude of authors to a general current affair. This is the case, for instance, of the two Olympic victories of France on the $19^{th}$ and $21^{th}$ of August (see Table I).

Studying interactions by looking at authors enables us to have a deeper understanding of events. In particular, it enables us to identify authors which are unexpectedly and primarily retweeted. This gives us hints on the event's origin: it might results of a focus on a single author, or multiple authors, or none in particular.

### C. Abnormal spreaders during events

Among the three previous cases, we are now interested in events generated by a single author (case 1). In particular, we seek to determine if their emergence is due to a large number of spreaders, or on the contrary, if they emerge only because of a small number of spreaders which retweet them abnormally.

To do so, we proceed as in the previous section and locally study interactions in the filtered data cube $\mathscr{C}_3(S \times \{a^*\} \times \{e\}, v)$, where $a^*$ is the predominant abnormal author corresponding to event $e$. A cell $(s, a^*, e)$ within this cube gives the total number of times $s$ retweeted $a^*$ during $e$. This way, we focus on how each of the spreaders retweeted $a^*$ during the event.

Expected values are defined from data cube $\mathscr{C}_4(S \times \{a^*\} \times D \times \{H_e\}, v)$, using the comparison data cubes $\mathscr{C}_3(S \times \{a^*\} \times \{H_e\}, v)$ and $\mathscr{C}_2(\{a^*\} \times \{H_e\}, v)$, obtained by aggregation, and $\mathscr{C}_3(S \times \{a^*\} \times \{e\}, v)$ obtained by aggregation and filtering:

$$v_{exp}(s, a^*, e) = v(\cdot, a^*, e) \times \frac{v(s, a^*, \cdot, H_e)}{v(\cdot, a^*, \cdot, H_e)},$$

where $v(\cdot, a^*, e)$ is the total number of retweets $a^*$ received during $e$; $v(s, a^*, \cdot, H_e)$ is the total number of time spreader $s$ retweeted author $a$ on hours of $H_e$; and $v(\cdot, a^*, \cdot, H_e)$ is the total number of retweets author $a$ received during $H_e$.

According to this context, a triplet $(s, a^*, e) \in S \times \{(a^*, e)\}$ is abnormal because there is a deviation between the number of time $s$ retweeted $a$ during $e$, and the number of time $s$ is expected to retweet $a$ during this same period on other days. Similarly, three situations arise.

*1) Global phenomena*

For events $(\text{fhollande}, (12^{th}, 22h), \cdots, (13^{th}, 1h))$ and $(\text{marseille}, 11^{th}, 0h)$, we observe distributions in which the range of deviation values is very small (see Figure 8). In the first case, we observe 22 different deviation values. Moreover, 90% of all triplets $(s, a^*, e)$ have their deviation equal to 1.7, 2.2, 2.8, or 3.1. For marseille, we observe the same patterns: there are only 7 different deviation values, among which 90% of all triplets are distributed between values 1.41, 1.23, and 1.16 (see Figure 8). Some of the behaviors corresponding to these values are described in Table II.

These distributions show a limited number of spreaders behaviors. None of them have significantly different activities than others. Thus, the emergence of fholland and marseille is due to a global phenomenon in which a large number of spreaders retweeted them.

*2) Group of online activists*

Figure 9 shows the distributions of deviation values for ev-ents $(\text{NicolasSarkozy}, 25^{th}, 19h)$, $(\text{alainjuppe}, 27^{th}, 15h)$, $(\text{JLMelenchon}, 28^{th}, 13h \text{ - } 15h)$ and $(\text{NicolasSarkozy}, 29^{th}, 7h \text{ - } 9h)$. Most observations $d_p \in \mathscr{D}$ follow a Gaussian

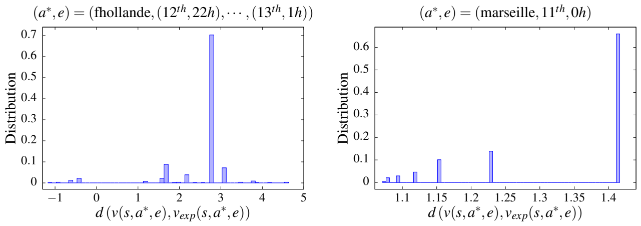


Fig. 8: **Distribution of deviation values in the case where all spreaders behave normally –** Bars beyond (*resp.* below) 0 correspond to spreaders which retweet $a^*$ during $e$ more (*resp.* less) than usual. For instance, the most extreme positive value for fhollande corresponds to a spreader which never retweeted fhollande from $22h$ to $1h$, except six times during the event. The most extreme negative value corresponds to a spreader which retweeted him once during the event, even though he retweeted him 7 times in total during this period.

| Event | $((12^{th},22h),\cdots,(13^{th},1h))$ | | | | $(11^{th},0h)$ | | |
|---|---|---|---|---|---|---|---|
| Abnormal author | fhollande | | | | marseille | | |
| Deviation value | 1.7 | 2.2 | 2.8 | 3.1 | 1.41 | 1.23 | 1.16 |
| % of spreaders | 9 | 4 | 70 | 7 | 66 | 14 | 10 |
| Number of retweets during $e$ | 1 | 2 | 1 | 2 | 1 | 2 | 3 |
| Total Number of retweets from $h_i$ to $h_j$ | 2 | 3 | 0 | 0 | 0 | 0 | 0 |

TABLE II: **Most probable behaviors in the case where all spreaders behave normally –** In both cases, the most probable deviation value corresponds to spreaders which retweets $a^*$ only once during the period. For marseille, we observe that the larger the number of retweets during $e$, the smaller the deviation value. This is due to Poisson deviation which takes into account the importance of the deviation between the observed value and its expected one.

distribution centred on a mean $\bar{d}_p$. Contrary to distributions in Sections VI-A and VI-B, $\bar{d}_p$ varies from 1.6 to 2.3. This shift indicates that globally, spreaders have an activity which is higher than usual, which partly explains the emergence of main author $a^*$.

We detect negative and positive outliers. Negative outliers indicate spreaders who retweet $a^*$ less that they are supposed to. As such, they do not influence the emergence of $a^*$. On the contrary, positive outliers, who are spreaders more active than usual, play a key role regarding the importance of $a^*$ during $e$. This is what we observe in Table III. For all events, we notice a small group of spreaders which extensively retweets $a^*$ and which accounts for a significant proportion of the total number of retweets. Within this group, several spreaders retweet $a^*$ more than 50 times during the event. Even if they represent a very small portion of all spreaders, they are a major cause of the emergence of $a^*$ during $e$.

### 3) One online activist

Event $(\text{FrancoisFillon}, 11^{th}, 3h)$ is an extreme case of the previous situation. The group of abnormal spreaders solely consists in one user which retweets FrancoisFillon 73 times at $3h$. Hence, the emergence of FrancoisFillon the $11^{th}$ at $3h$ is only due to this unique spreader which accounts for 100% of all its retweets.

Here again, local analysis of spreaders leads us to notice that some events are more global phenomena than others. In particular, some authors emergence is partly due to a small group of spreaders that substantially retweet them, which could mislead other users on the significance of these authors. Thereby, this analysis highlights crucial information that should be taken into account to evaluate the relevance of an event.

## D. Abnormal hashtags

It is possible to gain supplementary information on previous events by adding a content-based dimension using hashtags. In this section, we apply our method on dataset $\mathscr{D}_2$ and focus on the four dimensions: spreaders, authors, hashtags and time. First, we search for hours in which some hashtags are abnormally retweeted, then establish a correlation with previously detected events.

We are interested in abnormal triplets $(k^*, d^*, h^*)$ in data cube $\mathscr{C}_3(K \times D \times H, v)$. Given the ephemeral nature of hashtags, we use expected values slightly different than the previous ones. This time, we take into account the expected activity during hour $h$ and we adjust it with the number of hashtags $k$ retweeted on day $d$:

$$v_{exp}(k,d,h) = v(\cdot,\cdot,k,d,\cdot) \times \frac{v(\cdot,\cdot,\cdot,\cdot,h)}{v(\cdot,\cdot,\cdot,\cdot,\cdot)}.$$

This way, we do not assume that the number of hashtags observed at fixed hours is constant. According to this context, a triplet $(k^*, d^*, h^*)$ is abnormal when there is a significant deviation between the number of retweets containing hashtag $k^*$ during $(d^*, h^*)$, and the number of hashtags $k^*$ that would be retweeted on day $d$ if they were distributed among hours proportionally to their activity.

We find 225 abnormal triplets $(k^*, d^*, h^*)$, including 114 different hashtags (by ignoring differences in cases and accents). Among the 225 abnormal triplets, 43% correspond to a previously found abnormal event (in Subsections VI-A, VI-B and VI-C). Tables IV, V, and VI display abnormal hashtags according to their corresponding event, for events with respectively one, several and no main author(s). We can make several observations.

First, we notice that an event is often attached to a political slogan together with a radio or television show. In this case, there are three possible situations: either the show receives a political guest, or the show speaks about a topicality associated with one or more politician(s), or on the opposite, the show and the political slogan are uncorrelated – for instance, in the case in which several current events happen within the same period.

We notice that events in Tables V and VI are always associated with a general term, independent from a political slogan or a show. As suggested by the analysis of anomalous

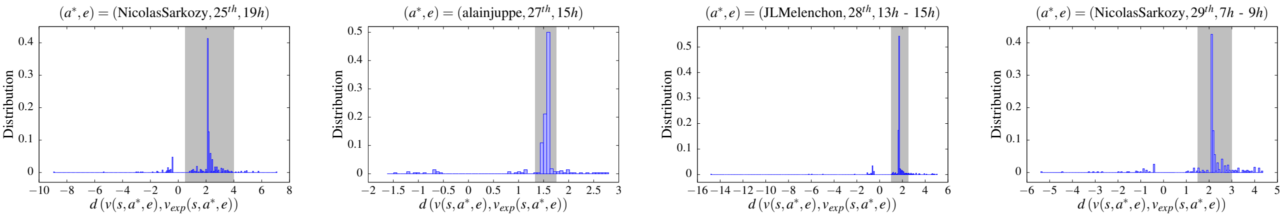


Fig. 9: **A group of spreaders behave abnormally -** In each distribution, similar behaviors are observed. Most spreaders retweet $a^*$ once or twice during $e$ while they usually never retweet $a^*$ at this time of day. These unusual but not significantly deviating behaviors are represented by the Gaussian curve with an average $\bar{d}_p$ between 1.6 and 2.3. Those who are used to retweet $a^*$ at this time of day have deviation values either close to 0, if they retweeted as they are used to, negative, if they retweeted less, or positive, if they retweeted more. In this last case, the group of spreaders which behave abnormally is largely responsible for the emergence of $a^*$.

| Event | $(25^{th}, 19h)$ | $(27^{th}, 15h)$ | $(28^{th}, 13h$ - $15h)$ | $(29^{th}, 7h$ - $9h)$ |
|---|---|---|---|---|
| Abnormal Author | NicolasSarkozy | alainjuppe | JLMelenchon | NicolasSarkozy |
| % of abnormal spreaders | 2.7 | 6.7 | 4.5 | 6 |
| % of retweets | 14 | 40 | 37 | 25 |

TABLE III: **Group of influential spreaders -** We observe that a small proportion of spreaders constitutes in fact a significant part of all retweets received by the main author during the event. For instance, for $(\text{alainjuppe}, 27^{th}, 15h)$, we detect 19 abnormal spreaders (6.7% of all spreaders). Together, they retweeted alainjuppe 513 times at $15h$, which consists in 40% of all its retweets during this hour.

| Event | $((12^{th}, 22h), ..., (13^{th}, 1h))$ | $(25^{th}, 19h)$ | $(27^{th}, 15h)$ | $(28^{th}, 13h$ - $15h)$ | $(29^{th}, 7h$ - $9h)$ |
|---|---|---|---|---|---|
| Abnormal hashtags | judo<br>rio2016<br>fra<br>espritbleu<br>*(blue spirit)* | **Campaign Slogan:**<br>toutpourlafrance<br>(*all for France*)<br><br>**Location:**<br>chateaurenard | **Campaign Slogan:**<br>3moispourgagner<br>(*3 month to win*) | **Campaign Slogan:**<br>benoithamon2017<br>lagauchepourgagner<br>*(left for win)*<br>insoumis28aout<br>(*rebellious of august* $28^{th}$)<br><br>**TV/Radio program:**<br>LeGrandJury | **Campaign Slogan:**<br>toutpourlafrance<br>*(all for France)*<br><br>**TV/Radio program:**<br>rtlmatin<br>télématin *(morning show)*<br>bourdindirect<br>invitépol *(political guest)* |

TABLE IV: **Abnormal hashtags of events with one main author.**

| Event | $(3^{rd}, 10h$ - $13h)$ | $(22^{th}, 16h$ - $22h)$ | $(24^{th}, 20h$ - $22h)$ | $(26^{th}, 15h$ - $18h)$ | $(28^{th}, 0h)$ |
|---|---|---|---|---|---|
| Abnormal hashtags | sainterita<br>*(name of a church)* | sarkozy<br><br>**Campaign Slogan:**<br>toutpourlafrance<br>*(all for France)*<br><br>**TV/Radio program:**<br>clubdelapresse, e1soir | sarko<br><br>**Campaign Slogan:**<br>toutpourlafrance<br>*(all for France)*<br><br>**TV program:**<br>ns20h | burkini<br>conseildetat<br>*(council of state)*<br><br>**TV/Radio program:**<br>BFMTV | salafisme *(salafism)*<br><br>**TV program:**<br>ONPC |

TABLE V: **Abnormal hashtags of events with several main authors.**

authors, this shows that the corresponding event results from the reaction to an external fact. For instance, hashtags "*Rio2016*" are related to the global reaction of users to Olympic victories of France. Hashtag "SainteRita", on the other hand, is related to the reaction of users to a police intervention in a church. Furthermore, events $(22^{nd}, 16h$ - $22h)$ and $(24^{th}, 20h$ - $22h)$, attached to hashtags "*Sarkozy*" and "*Sarko*", suggest that there is a discussion about Nicolas Sarkozy apart from official tweets and hashtags released by his team. In particular, on the $22^{nd}$, people react to the announcement of Nicolas Sarkozy's candidacy to presidency: this event corresponds with the first use of hashtag "*ToutpourLaFrance*" which is his campaign slogan.

We observe another interesting fact: on the $28^{th}$ from $13h$ to $15h$, we detect the campaign slogan of JLMelenchon, "*insoumis28aout*", which is expected since JLMelenchon is the predominant author of this event. However we also detect campaign slogans of benoithamon, another politician – "benoithamon2017" and "LaGauchePourGagner" – which is unexpected since it does not appear as a predominant author in the previous study.

Finally, we notice that events $(11^{th}, 0h)$, $(11^{th}, 3h)$, $(13^{th}, 9h)$ and $(23^{th}, 7h\text{-}8h)$ are not related to any detected hashtags. This is due to the fact the analysis performed in this subsection is global. With local analysis of abnormal hashtags, centred on events, as done before with authors in Subsection VI-B, we succeed in identifying the corresponding discussed topics. For instance, during event $(13^{th}, 9h)$, we identify abnormal

| Event | $(13^{th},9h)$ | $(19^{th},22h)$ | $(21^{st},21h)$ | $(23^{th},7h\text{-}8h)$ | $(30^{th},17h\text{-}18h)$ |
|---|---|---|---|---|---|
| Abnormal hashtags | ⁄ | rio2016 | rio2016<br>boxe *(boxing)* | ⁄ | macron |

TABLE VI: **Abnormal hashtags of events with no main authors.**

hashtags *etatdurgence* (*state of emergency*), *cazeneuve* and *islamigration*, referring to a measure taken that same day by the minister of the Interior, Bernard Caze-neuve.

In this section, we applied our method to datasets $\mathscr{D}_1$ and $\mathscr{D}_2$. We detected abnormal events, independent of the activity of the day or time considered. Then, we performed local analysis on each of these events, using numerous different contexts, more or less filtered or aggregated. This allowed us to understand their emergence. For instance, we learned that on the $11^{th}$ at $3h$, one unique spreader intensely retweets FrancoisFillon; that from the $12^{th}$ at $22h$ to the $13^{th}$ at $1h$, numerous spreaders retweet fhollande once, regarding an Olympic victory of France in judo; or, that on the $27^{th}$ at $15h$, a small group of spreaders is largely responsible for the emergence of alainjuppe during its political meeting. Our method provides the possibility of studying further aspects of interactions by choosing new relevant contexts. In the following section, we discuss two other possible applications.

## VII. Other Applications

Observations made in the previous section open up several research perspectives. On the one hand, given the ubiquity of news related hashtags within each events – as TV and Radio programs –, it would be interesting to characterize more precisely the reaction of users to television shows through Twitter. On the other hand, we could focus on topic dynamic over time and, in particular, on prediction of user-topic links.

### A. Characterization of second screen usage

The characterization of second screen usage is a very recent field of study. The term *second screen* refers to a web-connected screen, like a smartphone or a laptop, that people use to comment about TV programs on social media while watching television. As part of this study, it is interesting to analyse the differences between what is said in the TV program and the ensuing discussions on Twitter. This has been applied in many situations, in particular, to follow sport events [9] and political debates [13], [11], [14]. In the following, we provide elements to characterize the second screen usage with our method. This is a novel approach since previous studies often consist either in manual comparison between tweets content and a record of the discussion that took place in the TV show, or in a focus on the television audience or the number of tweets over time. We focus on Nicolas Sarkozy's appearance on television news for the launch of his campaign, on the $24^{th}$ of August from $20h$ to $22h$.

First, we focus on abnormal authors using the same expected values as in Section VI-B, separately on each hour. Figure 10 displays the distribution of the set of deviation values for $e_1 = (24^{th}, 20h)$, $e_2 = (24^{th}, 21h)$ and $e_3 = (24^{th}, 22h)$. At $20h$, there are two predominant authors: *NicolasSarkozy* and *TTpourlaFrance*, his team's account. At $21h$, another situation occurs. The set of values is more homogeneous: there are more outliers, but less significant. Among these, we see many journalists as well as right-wing politicians supporting Nicolas Sarkozy. We notice that some people, neither related to a newspaper nor to a political team, begin to appear among abnormal authors. Finally, at $22h$, the range of values is even smaller, meaning that the observed event is not the result of a focus on a limited number of authors, but a global phenomenon where everyone retweets everyone. Among outliers, we only see journalists and anonymous users. Hence, the more time passes, the more distributions are homogeneous, showing that the event becomes a global phenomenon as information spread.

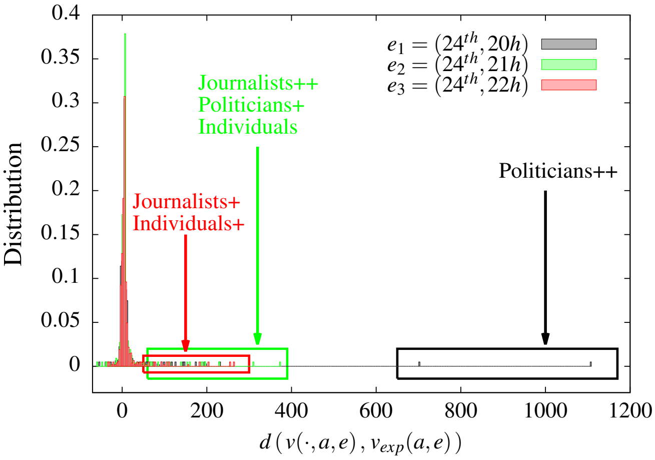


Fig. 10: **Evolution of abnormal authors distributions on the $24^{th}$ of August from $20h$ to $22h$.**

The previous analysis shows that the political interview on television is taken up by users on social media. In the same way as with abnormal authors, we now focus on abnormal hashtags to analyse how the discussion evolves over time. We observe similar distributions. At $20h$, the two hashtags *ns20h* and *toutpourlafrance* point out strongly. At $21h$ and $22h$, distributions are more homogeneous. The two previous hashtags released by Nicolas Sarkozy's team are still abnormal at $21h$, but become normal again at $22h$. Other hashtags are abnormal only from $21h$ to $22h$ or from $22h$ to $23h$. Among these, we find terms used by Nicolas Sarkozy during his interview, such as *chomage* (*unemployment*). Finally, we observe an evolution of hashtags referring to the same topic: at $21h$, *hollande*, then at $22h$, *hollandedemission* (*hollande resignation*); or *schengen* at $21h$, then *stopschengen* at $22h$; or *burkini* from $20h$ to $22h$, then *bikini* from $22h$

onwards.

This preliminary analysis could be continued. For instance, when studying abnormal hashtags, we could use local contexts, restrained to journalists, or Nicolas Sarkozy's political team, or independent users, in order to analyse which hashtags each of these communities propagate. Also, we could focus on the evolution of hashtags belonging to a same topic and see if they are retweeted by the same community of spreaders.

### B. Predicting User-Topic Links

The latter question attract a lot of interest among researchers: many are interested in topic dynamics and in particular, predicting user-topic links. The first difficulty lies in finding the set of terms forming a topic, *i.e.* a consistent semantic content. Some researchers characterize it from a set of hashtags whose temporal evolutions are similar [21], or from clusters of hashtags which are highly associated within tweets [3]. Others use text processing techniques to infer a topic from the entire text within tweets, rather than only using hashtags [35]. To predict user-topic links, most researchers use machine learning techniques for sentiment analysis [22], [26], [8], [23]. We also find methods based on lexicon [20]. In the following, we propose a new approach which consist in finding topics among abnormally retweeted hashtags.

We only have the structure of retweets $(s,a,k,d,h)$. In order to identify topics from this data, we take advantage of the fact that users are engaged in a cause, especially in the case of political communication. That is, an author will often post tweets related to this cause, and spreaders committed to the cause will retweet them intensely. Thus, we define a topic as being a set of hashtags retweeted intensely by the same spreaders and for which a common group of authors is intensely retweeted.

Formally, let $K_N \subseteq K$ be a set of $N$ hashtags. We proceed as follows. First, for each hashtag $k_i \in K_N$, we locally search what are the abnormal spreaders associated to $k_i$ according to the following expected values

$$v_{exp}(s,k_i,d,h) = v(\cdot,\cdot,k_i,d,h) \times \frac{v(s,\cdot,\cdot,\cdot,h)}{v(\cdot,\cdot,\cdot,\cdot,h)} \quad .$$

We obtain an abnormal spreader group denoted $S^*_{k_i}$ such that $s \in S^*_{k_i}$ is a spreader that retweets hashtag $k_i$ abnormally during a specific hour, given its usual activity at this time of the day. After performing this step on all hashtags, we define the group of spreaders related to $K_N$ as the set of abnormal spreaders common to all hashtags in the set: $S^*_{K_N} = \bigcap_{i=1}^{i=N} S^*_{k_i}$. We proceed symmetrically to find the set of abnormal authors related to $K_N$, denoted $A^*_{K_N}$. Given the set of abnormal spreaders and authors related to $K_N$, we say that $K_N$ is a topic if both $S^*_{K_N}$ and $A^*_{K_N}$ are non-empty (see Figure 11 for illustration). Note that we are only interested in abnormal authors and spreaders since they are the ones which unquestionably want to propagate the topic.

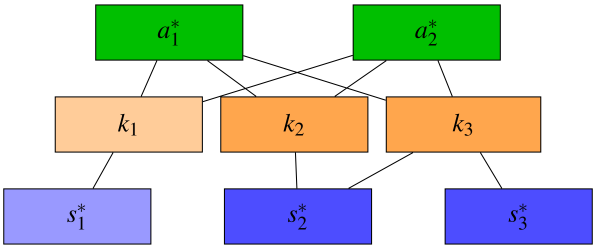


Fig. 11: **Formation of topics from hashtags –** For $K_3 = \{k_1,k_2,k_2\}$, $S^*_{k_1} = \{s^*_1\}$, $S^*_{k_2} = \{s^*_2\}$, and $S^*_{k_3} = \{s^*_2,s^*_3\}$. Then, $S^*_{K_3} = \emptyset$ and $K_3$ does not constitute a topic. On the other hand, $K_2 = \{k_2,k_2\}$ is a topic since $A^*_{K_2} = \{a^*_1,a^*_2,a^*_3\}$ and $S^*_{K_2} = \{s^*_2\}$.

With $N = 3$ and by considering the set of triplets obtained from the 114 abnormal hashtags identified in the previous section, we find 876 topics. For instance, we identify topic $K_3$ = {*chateaurenard, ns20h,toutpourlafrance*}, which has 4 abnormal authors belonging to the same political party, $A^*_{K_3}$ = {GilAverous, LArribage, NicolasSarkozy, TTpourlaFrance}, and 48 abnormal spreaders; topic $K'_3$ = {*3moispourgagner, legrandrdv (radio program), uemedef2016 (summer school of the employers' federation of France)*} associated to one abnormal author, alainjuppe, and to a group of 18 abnormal spreaders; and topic $K''_3$ = {*boxe (boxing), judo, rio2016*} associated to 7 abnormal authors from different origins, and only 3 abnormal spreaders[3]. Figure 12 shows the temporal evolution of each hashtag in each topic. We see that hashtags belonging to the same topic do not necessarily have the same dynamics.

After this step and from this set of topics, we can infer user's communities according to which topic they are used to retweet or being retweeted. Now, we address the problem of predicting user-topic links. More precisely, we want to predict the number of interactions between spreader $s$, in community $c_s$, and topic $K_N$ during hour $(d,h)$. Link prediction is inextricably related to abnormal link detection. Indeed, if the detection of abnormal quadruplets $(s,K_N,d,h)$ is based on measuring the deviation between an observed value $v(s,K_N,d,h)$ and its expected value $v_{exp}(s,K_N,d,h)$, link prediction focuses on describing normal behavior and therefore, is based on expected values only. For instance, we could predict the number of interactions between $s$ and $K_N$ during $(d,h)$ as

$$v_{exp}(s,K_N,d,h) =$$

$$\underset{(1)}{\frac{v(c_s,\cdot,K_N,\cdot,\cdot)}{v(\cdot,\cdot,K_N,\cdot,\cdot)}} \times \underset{(2)}{\frac{v(s,\cdot,\cdot,\cdot,h)}{v(c_s,\cdot,\cdot,\cdot,h)}} \times \underset{(3)}{\frac{v(\cdot,\cdot,K_N,d,h)}{|D|}}$$

[3] Note that in this case, we only find 3 abnormal spreaders since events related to sport are usually homogeneous events which do not exhibit groups of active spreaders.

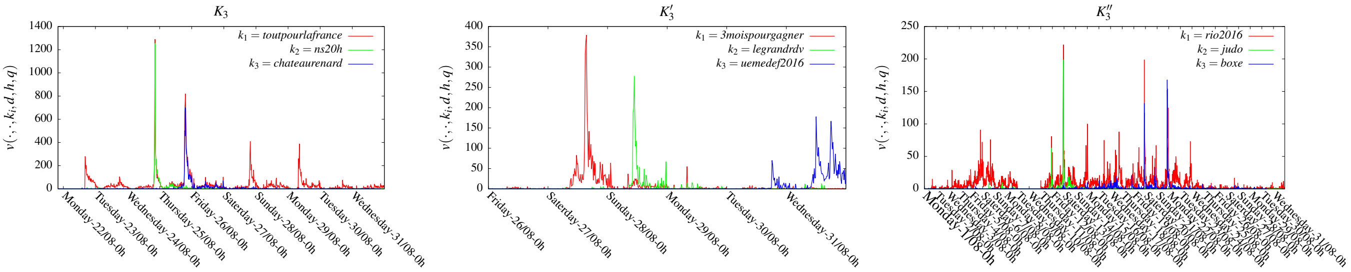


Fig. 12: **Evolution of the number of retweets containing hashtag $k_i$ for three different topics –** Notice that, in order to have a better accuracy, we plotted the number of retweets containing hashtag $k_i$ per quarter $q$. We see that hashtags dynamics within a same topic can be uncorrelated as in $K'_3$, or correlated as *ns20h* and *chateaurenard* with *toutpoutlafrance* in $K_3$.

which takes into account (1) the activity of $s$'s community towards topic $K_N$, (2) the activity of $s$ within its community during the hour of the day $h$, and (3) the expected number of retweets of $K_N$ during hour $h$ of day $d$.

This prediction can be improved by taking into account the behavior of authors that $c_s$ is used to retweet, towards topic $K_N$. Also, if $K_N$ is a new topic, we could imagine to replace the activity of topic $K_N$ by the mean activity of a set of related topics.

Thus, our method may be useful in many empirical studies and applications. In turn, these applications provide feedback and questions necessary to create more and more complex and relevant contexts and thus, take advantage of the scope of possibilities offered by our method.

## VIII. Conclusion

In this paper, we provided a method to meticulously explore millions of interactions and find unexpected behaviors under a multitude of situations. We applied it in the context of politics, where the stakes to unravel relevant information in the flow of data are particularly high. We showed that our method successfully highlights events and provide explanations for their emergence. In particular, we found abnormally retweeted authors, groups of very active spreaders, and hot topics during the corresponding abnormal periods. Hence, our method highlights crucial information that should be taken into account to evaluate an event reliability on Twitter.

One interesting perspective that could be considered would be to aggregate the base cuboid over authors, spreaders or hashtag (or topics) partitions. This would allow us to study each community separately – especially the ones corresponding to political parties; the relationship they have with each other; as well as the one they have with the different hastags (*resp.* topics). This in turn would enable us to gain insights about communication strategies deployed by each political parties.

Moreover, our method applies to temporal networks modelling entities interacting over time in general. Hence, as discussed in Section VII, numerous applications can benefit from it, as for instance, the characterization of second screen usage on social media (*e.g.* Facebook or Twitter) and link prediction (*e.g.* in IP traffic or e-mail exchanges).

## Acknowledgements

This work is funded in part by the European Commission H2020 FETPROACT 2016-2017 program under grant 732942 (ODYCCEUS) and by the ANR (French National Agency of Research) under grants ANR-15- E38-0001 (AlgoDiv).

## References

[1] D. R. Bild, Y. Liu, R. P. Dick, Z. M. Mao, and D. S. Wallach. Aggregate characterization of user behavior in Twitter and analysis of the retweet graph. *ACM Transactions on Internet Technology (TOIT)*, 15(1):4, 2015.

[2] A. Bruns, J. E Burgess, K. Crawford, and F. Shaw. # qldfloods and@ qpsmedia: Crisis communication on twitter in the 2011 south east queensland floods. *ARC Centre of Excellence for Creative Industries and Innovation*, 01 2012.

[3] F. M. Cardoso, S. Meloni, A. Santanche, and Y. Moreno. Topical homophily in online social systems. *arXiv preprint arXiv:1707.06525*, 2017.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[5] N. Chavoshi, H. Hamooni, and A. Mueen. Temporal patterns in bot activities. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1601–1606. International World Wide Web Conferences Steering Committee, 2017.

[6] F. Chierichetti, J. M. Kleinberg, R. Kumar, M. Mahdian, and S. Pandey. Event Detection via Communication Pattern Analysis. In *ICWSM*, 2014.

[7] M. Coletto, K. Garimella, A. Gionis, and C. Lucchese. Automatic controversy detection in social media: A content-independent motif-based approach. *Online Social Networks and Media*, 3:22–31, 2017.

[8] E. Colleoni, A. Rozza, and A. Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332, 2014.

[9] D. Corney, C. Martin, and A. Göker. Spot the ball: Detecting sports events on twitter. In *European Conference on Information Retrieval*, pages 449–454. Springer, 2014.

[10] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard. Multiscale event detection in social media. *Data Mining and Knowledge Discovery*, 29(5):1374–1405, 2015.

[11] D. Freelon and D. Karpf. Of big birds and bayonets: Hybrid twitter interactivity in the 2012 presidential debates. *Information, Communication & Society*, 18(4):390–406, 2015.

[12] N. Gaumont, M. Panahi, and D. Chavalarias. Reconstruction of the socio-semantic dynamics of political activist Twitter networks-Method and application to the 2017 French presidential election. *PLoS ONE*, 13(9), 2018.

[13] F. Giglietto and D. Selva. Second screen and participation: A content analysis on a full season dataset of tweets. *Journal of Communication*, 64(2):260–277, 2014.

[14] H. Gil de Zúñiga, V. Garcia-Perdomo, and S. C. McGregor. What is second screening? exploring motivations of second screen use and its effect on online political participation. *Journal of Communication*, 65(5):793–815, 2015.

[15] C. Grasland, R. Lamarche-Perrin, B. Loveluck, and H. Pecout. International agenda-setting, the media and geography: A multi-dimensional analysis of news flows. *L'Espace géographique*, 45(1):25–43, 2016.

[16] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[17] D. M. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[18] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang. Tedas: A twitter-based event detection and analysis system. In *Data engineering (icde), 2012 ieee 28th international conference on*, pages 1273–1276. IEEE, 2012.

[19] D. Murthy. *Twitter: Social Communication in the Twitter Age*. Digital Media and Society. Wiley, 2013.

[20] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[21] L. Pépin, J. Blanchard, F. Guillet, P. Kuntz, and P. Suignard. Visual analysis of topics in twitter based on co-evolution of terms. In *Data Science, Learning by Latent Structures, and Knowledge Discovery*, pages 169–178. Springer, 2015.

[22] F. Ren and Y. Wu. Predicting user-topic opinions in twitter with social and topical context. *IEEE Transactions on Affective Computing*, 4(4):412–424, 2013.

[23] A. Reyes-Menendez, J. Saura, and C. Alvarez-Alonso. Understanding# worldenvironmentday user opinions in twitter: A topic-based sentiment analysis approach. *International journal of environmental research and public health*, 15(11):2537, 2018.

[24] M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. Almeida, and W. Meira Jr. Characterizing and Detecting Hateful Users on Twitter. *arXiv preprint arXiv:1803.08977*, 2018.

[25] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[26] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng. Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 24–29, 2013.

[27] J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *International workshop on recent advances in intrusion detection*, pages 301–317. Springer, 2011.

[28] S. Stieglitz and L. Dang-Xuan. Political communication and influence through microblogging–An empirical analysis of sentiment in Twitter messages and retweet behavior. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 3500–3509. IEEE, 2012.

[29] M. Ten Thij, T. Ouboter, D. Worm, N. Litvak, H. van den Berg, and S. Bhulai. Modelling of trends in twitter using retweet graph dynamics. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 132–147. Springer, 2014.

[30] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini. Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*, 2017.

[31] M. Walther and M. Kaisser. Geo-spatial event detection in the twitter stream. In *European conference on information retrieval*, pages 356–367. Springer, 2013.

[32] K. Weller, A. Bruns, J. Burgess, M. Mahrt, and C. Puschmann. *Twitter and society*, volume 89. Peter Lang, 2014.

[33] A. Wilmet and R. Lamarche-Perrin. Multidimensional outlier detection in interaction data: Application to political communication on twitter. In *International Workshop on Complex Networks*, pages 147–155. Springer, 2019.

[34] F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang. Quantifying political leaning from tweets, retweets, and retweeters. *IEEE transactions on knowledge and data engineering*, 28(8):2158–2172, 2016.

[35] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2216–2229, 2016.

# Link weights recovery in heterogeneous information networks

Hong-Lan Botterman[1] and Robin Lamarche-Perrin[2]

[1] *Sorbonne Université, CNRS, LIP6, F-75005 Paris, France*
[2] *Institut des systèmes complexes de Paris Île-de-France, ISC-PIF, UPS 3611, Paris, France*

Socio-technical systems usually consists of many intertwined networks, each connecting different types of objects (or actors) through a variety of means. As these networks are co-dependent, one can take advantage of this entangled structure to study interaction patterns in a particular network from the information provided by other related networks. A method is hence proposed and tested to recover the weights of missing or unobserved links in heterogeneous information networks (HIN) - abstract representations of systems composed of multiple types of entities and their relations. Given a pair of nodes in a HIN, this work aims at recovering the exact weight of the incident link to these two nodes, knowing some other links present in the HIN. To do so, probability distributions resulting from path-constrained random walks i.e., random walks where the walker is forced to follow only a specific sequence of node types and edge types, capable to capture specific semantics and commonly called a meta-path, are combined in a linearly fashion in order to approximate the desired result. This method is general enough to compute the link weight between any types of nodes. Experiments on Twitter and bibliographic data show the applicability of the method.

## I. INTRODUCTION

Networked data are ubiquitous in real-world applications. Examples of such data are humans in social activities, proteins in biochemical interactions, pages of Wikipedia or movies-users from Amazon just to name a few. These are abstracted by a network where nodes represent the entities (e.g. individuals or pages) of the examined system whilst (directed) links stand for existing physical or virtual ties between them. Weights can also be put on links to state, for instance, their importance. In some cases, the nodes and/or the links are of different nature. For example, in social activities, the links can reflect online or offline communication or more obviously, in the movie-user case, nodes represent two different objects. Taking these differences explicitly into account in the modeling can only enrich the understanding of the inspected system. Thus, heterogeneous information networks (HIN), abstract representations of systems composed of multiple types of entities and their relations, are good candidates to model such data together with their relations, since they can effectively fuse a huge quantity of information and contain rich semantics in nodes and links.

In the last decade, the heterogeneous information network analysis has attracted a growing interest and many novel data mining tasks have been designed in such networks, such as similarity search, clustering, classification and link prediction [19]. The latter can sometimes refer to the term recovery, in the sense that links already exist but are missing or imperfectly observed in the data. This could be due to sampling or depending on the system under scrutiny, due to node/agent's voluntary decision not to give access to all her data (e.g. online social apps). Whatever the reason, capturing the presence of a link is sometimes not enough sufficient. For instance, in a social network, knowing two individuals are linked does not say anything about the frequency of their communication or the strength of their friendship. Hence recovering the actual link weight can bring useful information as for instance, in recommendation systems where the weight can be taken for the “rating" a user would give to an item. The goal of this work is to recover, for a given pair of nodes in a weighted HIN, the actual incident link weight to these two nodes, knowing some other links present in the HIN.

Link prediction can be related to node similarity problem. Indeed, the similarity score between two nodes, result of a particular function of these two nodes, can be seen as the strength of their connection. Here, this function is related to a particular random walk on the graph and so, to the probabilities of reaching one node through different paths, starting from another.

In HIN, most of similarity scores [8, 20] are based on the concept of meta-path. In simple terms, this corresponds to a concatenation of node types linked by corresponding link types and the type of a node/link is basically a label in the abstract representation. Meta-paths can be used as a constraint to a classic random walk: the walker is allowed to take only paths satisfying a particular meta-path. These path-constrained random walks have the sensitivity to explicitly take into account different semantics present in HIN. For instance, in a bibliographic network, we can distinguish four types of entities: Authors (A), Papers (P), Venues (V) and Topics (T). Starting from a particular paper, if a walker follows the meta-path PVP, he is likely to end to any another paper published in the same venue than the first. Now, if he follows the meta-path PTP, the ending paper will be about the same topic. Even if the starting and ending papers are the same, the semantics behind may be radically different.

Back to our goal, we can see it as a (linear) regression problem where the aim is to recover the link weight i.e., a continuous value. This means that the target link weight between a pair of nodes is approximated by a linear combination of probabilities, results of path-constrained random walks performed on the HIN. These probabilities thus translate the fact of being at a particular node starting from another one and are the regressors for the linear regression. Thenceforth, in order to make recovery tasks, data is commonly split into two sets: training and test. The proposed method aims at finding a relevant set of meta-paths together with their coefficient such that the difference between the exact link weight and its approximation is minimized for the training set. Obtained coefficient are then tested on the test set.

The rest of this paper is organized as follows. In Sec. II, some basic concepts about HIN are presented and the problem statement is exposed. Sec. III explains our method and we apply it on empirical data in Sec. IV. First, in Sec. IVA, the method is tested to recover the link weights between entities of different types into Twitter data. Then, in Sec. IVB, it is applied on bibliographic data with the same type of target nodes. We review some related work in Sec. V and we finally conclude and discuss some perspectives in Sec. VI.

## II. PREMIMINARY CONCEPTS

In this section, we present some basic concepts of weighted HIN useful for the following and define the "weight recovery" problem. Fig. 1 illustrates this section.

**Definition 1 (Weighted directed multigraph)** *A weighted directed mutligraph is a 5-tuple $G := (V, E, w, \mu_s, \mu_t)$ with $V$ the node set, $E$ the link set, $w : E \to \mathbb{R}^+$ the function that assigns to each link a real non negative weight, $\mu_s : E \to V$ the function that assigns to each link a source node, $\mu_t : E \to V$ the function that assigns to each link a target node.*

This concept allows us to introduce the definition of HIN which basically is a weighted directed multigraph with multiple node and link types.

**Definition 2 (Heterogeneous Information Network)** *A HIN $H := (G, \mathscr{V}, \mathscr{E}, \phi, \psi)$ is a weighted directed multigraph $G$ along with $\mathscr{V}$ the node type set, $\mathscr{E}$ the link type set, $\phi : V \to \mathscr{V}$ the function that assigns a node type to each node and $\psi : E \to \mathscr{E}$ the function that assigns a link type to each link such that if two links belong to the same link type, the two links share the same starting and target node type i.e., $\forall e_1, e_2 \in E$, $\big(\psi(e_1) = \psi(e_2)\big) \Rightarrow \big(\phi(\mu_s(e_1)) = \phi(\mu_s(e_2)) \wedge \phi(\mu_t(e_1)) = \phi(\mu_t(e_2))\big)$.*

Fig.1a illustrates such a network composed of five node types and twenty link types. However, disentangling the different entities present in the HIN is not necessarily a trivial task. Indeed, it sometimes takes a broader view of the system in question to describe it. For that purpose, the concept of network schema i.e. the meta level description of the HIN, is proposed. In simple terms, this corresponds to the graph defined over the node and link types of the associated HIN. It is represented in Fig.1b.

**Definition 3 (HIN Schema)** *Let $H$ be a HIN. The schema $T_H$ for $H$ is a directed graph defined on the node types $\mathscr{V}$ and the link types $\mathscr{E}$ i.e., $T_H := (\mathscr{V}, \mathscr{E}, \nu_s, \nu_t)$ with $\nu_s : \mathscr{E} \to \mathscr{V} : E^* \mapsto \nu_s(E^*) := \phi\big(\mu_s(e)\big)$ the function that assigns to each link a source node and $\nu_t : \mathscr{E} \to \mathscr{V} : E^* \mapsto \nu_t(E^*) := \phi\big(\mu_t(e)\big)$ the function that assigns to each link a target node, where $e \in E$ such that $\psi(e) = E^*$*

Note that we can effectively take any such element $e$ since $\{e \in E \,|\, \psi(e) = E^*\}$ is the equivalence class of any of its elements, with the equivalence relation "has the same type of". By definition of HIN, it is sufficient to take one member of the equivalence class to know the node types that the link type $E^*$ connects.

Two entities in a HIN can be linked via different paths and these paths have different semantics. These paths can be defined as meta-paths as follows [19].

**Definition 4 (Meta-path)** *A meta-path $\mathscr{P}$ of length $n \in \mathbb{N}$ is a sequence of node types $V_0, \cdots, V_n \in \mathscr{V}$ linked by link types $E_1, \cdots, E_n \in \mathscr{E}$ as follows: $\mathscr{P} = V_0 \xrightarrow{E_1} V_1 \cdots V_{n-1} \xrightarrow{E_n} V_n$ which can also be denoted as $\mathscr{P} = E_1 E_2 \cdots E_n$.*

Given a meta-path $\mathscr{P} = V_0 \xrightarrow{E_1} V_1 \cdots V_{n-1} \xrightarrow{E_n} V_n$ and a path $P = v_0 \xrightarrow{e_1} v_1 \cdots v_{n-1} \xrightarrow{e_n} v_n$, if $\forall i \in \{0, \ldots, n\}$, $\phi(v_i) = V_i$, $\forall i \in \{1, \ldots, n\}$, $\mu_s(e_i) = v_{i-1}$, $\mu_t(e_i) = v_i$ and $\psi(e_i) = E_i$, then path $P$ satisfies meta-path $\mathscr{P}$ and we note $P \in \mathscr{P}$. Hence, a meta-path is a set of paths.

In Fig.1b, an example of meta-path is ■ → ▲ → ★, in blue, in the network schema. Blue paths in the HIN in Fig.1a are said to satisfy this meta-path since each one of their segments respects the aforementioned conditions.

**Problem 1 (Weight recovery)** *Let be a HIN $H = (G, \mathscr{V}, \mathscr{E}, \phi, \psi)$, with $G = (V, E, w, \mu_s, \mu_t)$ a directed weighted multigraph, and a target link type $E_c$ between two node types. The "weight recovery problem" is to find a set of relevant meta-paths $\mathscr{E}_{\mathscr{P}}$ and a linear function $F$ of probabilities resulting from random walks constrained by these meta-paths that best quantifies, for each pair of nodes in $H$, the strength of their connection via $E_c$.*

## III. METHOD

We present our method for solving Problem 1 in three steps. Consider a HIN and let us denote by $E_c$ the target link type defined between $V_0$ and $V_n$. We consider a meta path $\mathscr{P} = V_0 \xrightarrow{E_1} V_1 \cdots V_{n-1} \xrightarrow{E_n} V_n$ different from $E_c$. There may be repetitions in this sequence of nodes and links. Let us introduce the notation $\mathscr{P} \equiv \mathscr{P}^{0,n}$ and let us denote by $\mathscr{P}^{a,b}$ the truncated meta path of $\mathscr{P}$ from node type $V_a$ to $V_b$.

### A. Path-Constrained Random Walk.

Let $X_i \in V_i$ be a random variable representing the position of a random walker in the set $V_i$. A random walk starting from $X_0$ constrained by the meta-path $\mathscr{P}$ corresponds to a discrete-time Markov chain i.e., a sequence of random variables $X_0, X_1, \ldots, X_n$ with the Markov property: $\forall i \in \{0, \ldots, n\}, \forall (v_0, \ldots, v_n) \in V_0 \times \ldots \times V_n$,

$$\mathbb{P}(X_i = v_i \,|\, X_{i-1} = v_{i-1}, \ldots, X_0 = v_0) = \mathbb{P}(X_i = v_i \,|\, X_{i-1} = v_{i-1}).$$

Here, since there may be more than one link type between two node types, we introduce de notation $\mathbb{P}((X_i = v_i \,|\, X_{i-1} = v_{i-1}) \,|\, \mathscr{P}^{i,i+1}) =: \mathbb{P}((v_i \,|\, v_{i-1}) \,|\, \mathscr{P}^{i,i+1}) = \mathbb{P}((v_i \,|\, v_{i-1}) \,|\, E_i)$ to emphasize the fact that the random walk is constrained by the meta-path $\mathscr{P}$. This means that for a walker to reach $v_i$ from

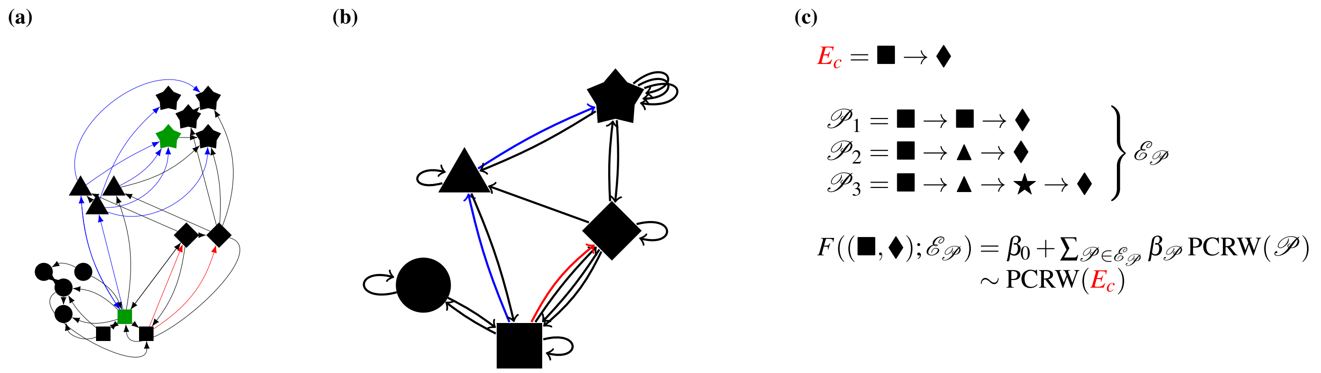

(a)
(b)
(c)
$E_c = \blacksquare \rightarrow \blacklozenge$
$\mathscr{P}_1 = \blacksquare \rightarrow \blacksquare \rightarrow \blacklozenge$
$\mathscr{P}_2 = \blacksquare \rightarrow \blacktriangle \rightarrow \blacklozenge$
$\mathscr{P}_3 = \blacksquare \rightarrow \blacktriangle \rightarrow \bigstar \rightarrow \blacklozenge$
$\mathscr{E}_{\mathscr{P}}$
$F((\blacksquare,\blacklozenge);\mathscr{E}_{\mathscr{P}}) = \beta_0 + \sum_{\mathscr{P}\in\mathscr{E}_{\mathscr{P}}} \beta_{\mathscr{P}}\,\mathrm{PCRW}(\mathscr{P})$
$\sim \mathrm{PCRW}(E_c)$

Figure 1: **(a)** Example of HIN composed of multiple node types, represented by diverse shapes, an multiple link types. Nodes are already grouped by shapes. **(b)** Its associated network schema composed by five nodes and twenty links. Each node corresponds to a set of nodes in the corresponding HIN. In the same way, each link is a set of links in the corresponding HIN. See for instance the paths and meta-path of length two in blue $\blacksquare \rightarrow \blacktriangle \rightarrow \bigstar$; the blue paths are said to satisfy the blue meta-path. **(c)** Illustration of the problem statement. For each pair of nodes in ($\blacksquare$,$\blacklozenge$), there is possibly a link connecting them. The link weight is approximated by a linear combination of the path-constrained random walk results i.e., probability distributions of being at a particular node. Roughly speaking, the probabilities resulting from the random walk constrained by the target meta path $E_c = \blacksquare \rightarrow \blacklozenge$, denoted by $\mathrm{PCRW}(E_c)$, are expressed as a linear combination $F$ of probabilities resulting from the random walks constrained by three different meta-paths $\mathscr{P}_1 = \blacksquare \rightarrow \blacksquare \rightarrow \blacklozenge$, $\mathscr{P}_2 = \blacksquare \rightarrow \blacktriangle \rightarrow \blacklozenge$, $\mathscr{P}_3 = \blacksquare \rightarrow \blacktriangle \rightarrow \bigstar \rightarrow \blacklozenge$, denoted by $\mathrm{PCRW}(\mathscr{P}_1)$, $\mathrm{PCRW}(\mathscr{P}_2)$ and $\mathrm{PCRW}(\mathscr{P}_3)$ respectively, whose real-valued coefficients are $\beta_{\mathscr{P}_1}, \beta_{\mathscr{P}_2}$ and $\beta_{\mathscr{P}_3}$ respectively plus a possible independent term $\beta_0$, that is to say $F((\blacksquare,\blacklozenge);\mathscr{E}_{\mathscr{P}}) = \beta_0 + \sum_{\mathscr{P}\in\mathscr{E}_{\mathscr{P}}} \beta_{\mathscr{P}}\,\mathrm{PCRW}(\mathscr{P})$. One can see that other meta-paths exist between nodes in ($\blacksquare$,$\blacklozenge$). The problem is to identify the "best" $\mathscr{E}_{\mathscr{P}}$ and a linear function $F$ with respect to $\mathrm{PCRW}(E_c)$.

$v_{i-1}$, he has to follow only links of type $E_i \equiv \mathscr{P}^{i,i+1}$. The probability $\mathbb{P}((v_i\,|\,v_{i-1})\,|\,E_i)$ thus defined is computed as

$$\mathbb{P}((v_i\,|\,v_{i-1})\,|\,E_i) = \frac{w_{E_i}(v_{i-1},v_i)}{\sum_k w_{E_i}(v_{i-1},v_k)}$$

where $w_{E_i}(v_j,v_k)$ is the link's weight of type $E_i$ between nodes $v_j$ and $v_k$.

Thenceforth, given $v_n \in V_n$ and $v_0 \in V_0$, the probability of reaching $v_n$ from $v_0$ following the meta path $\mathscr{P}$, denoted by $\mathbb{P}((v_n|v_0)\,|\,\mathscr{P})$, is simply defined by the random walk starting at $v_0$ and ending at $v_n$ following only paths satisfying $\mathscr{P}$. This conditional probability may be expressed recursively by means of the law of total probability

$$\begin{aligned}\mathbb{P}((v_n|v_0)\,|\,\mathscr{P}) &= \sum_{v_{n-1}\in V_{n-1}} \Bigg[\mathbb{P}\Big((v_n|v_{n-1})\,|\,E_n\Big) \\ &\qquad \times \mathbb{P}\Big((v_{n-1}|v_0)\,|\,\mathscr{P}^{0,n-1}\Big)\Bigg] \\ &= \sum_{v_{n-1}\in V_{n-1}} \Bigg[\frac{w_{E_n}(v_{n-1},v_n)}{\sum_k w_{E_n}(v_{n-1},v_k)} \\ &\qquad \times \mathbb{P}\Big((v_{n-1}|v_0)\,|\,\mathscr{P}^{0,n-1}\Big)\Bigg] \qquad (1)\end{aligned}$$

with $\mathbb{P}((v_1|v_0)|\mathscr{P}^{0,1}) = w_{E_1}(v_0,v_1)/\sum_k w_{E_1}(v_0,v_k)$ the basis of recurrence. In the following, we use the notation $\mathrm{PCRW}(\mathscr{P})$ to denote the column vector of such conditional probabilities $\mathbb{P}((v_n|v_0)\,|\,\mathscr{P})$, $\forall v_0$, $v_n$ i.e., $\mathrm{PCRW}(\mathscr{P}) = [\mathbb{P}((v_0|v_0)\,|\,\mathscr{P}), \mathbb{P}((v_1|v_0)\,|\,\mathscr{P}), ...., \mathbb{P}((v_n|v_n)\,|\,\mathscr{P})]^{\mathrm{T}}$.

For instance, in the HIN in Fig.1a, the probability for a walker to reach the green star $\bigstar$ from the green square $\blacksquare$ following the meta-path $\blacksquare \rightarrow \blacktriangle \rightarrow \bigstar$ equals 5/12.

Note that we forbid the walker to return to the initial node on the penultimate step of the walk i.e., if $V_{n-1} = V_0$, the sum in eq. (1) only holds for all $v_{n-1} \neq v_0$. It prevents us from using what we are looking for to find what we are looking for.

**Remark 1 (Hole nodes)** *It is possible that a node $v_i \in V_i$ is not connected to any node $v_j \in V_j$ by the link type $E_{ij}$ and thus, the transition probability is not defined. To overcome this problem, we provide each set $V_k$ with a hole node $h_k$ on which point all the disconnected nodes. Plus, all the holes are connected with each other and holes cannot point to another node (i.e., no hole node). Formally, $\forall V_k \in \mathscr{V}$, $V_k^h := V_k \cup \{h_k\}$. $\forall E_{ij} \in \mathscr{E}$, if $w_{E_{ij}}(v_i,v_j) = 0, \forall v_j \in V_j$ then $w_{E_{ij}}(v_i,h_j) = 1$, otherwise $w_{E_{ij}}(v_i,h_j) = 0$. Furthermore, $\forall E_{ij} \in \mathscr{E}$, $w_{E_{ij}}(h_i,h_j) = 1$ and $\sum_{v_j\in V_j} w_{E_{ij}}(h_i,v_j) = 0$. In this fashion, transition probabilities are always well defined.*

### B. Linear Regression Model.

Since $H$ is a HIN, multiple types of links can connect the nodes. Hence, there is no reason to restrict ourselves to a single meta path to compute the reachability of one node from another. As a result, the similarity between $v_n$ and $v_0$ is defined by several path-constrained random walk results combined through a linear regression model of the form

$$F((v_n|v_0)\,|\,\mathscr{E}_{\mathscr{P}}) := \beta_0 + \sum_{\mathscr{P}\in\mathscr{E}_{\mathscr{P}}} \beta_{\mathscr{P}}\,\mathbb{P}((v_n|v_0)\,|\,\mathscr{P})$$

where $\mathscr{E}_{\mathscr{P}}$ is the set of selected meta-paths and the vector $\boldsymbol{\beta} := [\beta_0, \beta_1, \cdots, \beta_{|\mathscr{E}_{\mathscr{P}}|}]^{\mathrm{T}}$ is real-valued coefficients. The coefficients stress the contribution of each meta-path in the final similarity score $F((v_n|v_0)\,|\,\mathscr{E}_{\mathscr{P}})$. Since the components of $\boldsymbol{\beta}$ are not confined in [0,1] and do not sum to 1, $F$ is a real-valued function whose image is neither confined in [0,1].

Now, we have a linear regression problem since we want to recover the exact link weights with respect to $E_c$. The dependant variable is thus PCRW($E_c$) whilst the predictors are PCRW($\mathscr{P}$), $\mathscr{P} \in \mathscr{E}_{\mathscr{P}}$. The choice of linear model is simply motivated by its interpretability in our particular case. Given example node pairs and their link weights, $\beta$ is estimated by the least squares method which is appreciated for its applicability and simplicity. In formulae with $\mathbb{1}$ the column vector whose entries are 1:

$$\begin{array}{ccccccc} \underset{\downarrow}{\text{PCRW}(E_c)} & & \underset{\downarrow}{\mathbb{1}} & \underset{\downarrow}{\text{PCRW}(\mathscr{P}_0)} & \ldots & \underset{\downarrow}{\text{PCRW}(\mathscr{P}_{|\mathscr{E}_{\mathscr{P}}|})} & \end{array}$$

$$\begin{bmatrix} \\ \text{PCRW}(E_c) \\ \\ \end{bmatrix} = \begin{bmatrix} & & \\ & \text{PCRW}(\mathscr{E}_{\mathscr{P}}) & \\ & & \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{\beta} \\ \\ \end{bmatrix} + \begin{bmatrix} \\ \boldsymbol{\varepsilon} \\ \\ \end{bmatrix}$$

$$= \left[ \; F(\mathscr{E}_{\mathscr{P}}) \; \right] + \left[ \boldsymbol{\varepsilon} \right]$$

and we choose $\hat{\beta}$ such that the residual sum of squares RSS $= \varepsilon^T \varepsilon = \|\varepsilon\|^2$ is minimized i.e., $\hat{\beta} = \left(\text{PCRW}(\mathscr{E}_{\mathscr{P}})^{\text{T}}\, \text{PCRW}(\mathscr{E}_{\mathscr{P}})\right)^{-1} \text{PCRW}(\mathscr{E}_{\mathscr{P}})^{\text{T}}\, \text{PCRW}(E_c)$.

### C. Forward Selection Procedure.

In order to determine the set $\mathscr{E}_{\mathscr{P}}$, we use the forward selection with $p$-value and $r^2$ criteria. This is a greedy approach but very simple and intuitive. The $p$-values are used to test the significance of each predictor. Given the hypothesis $H_0 : \beta = 0$ against the hypothesis $H_1 : \beta \neq 0$, the $p$-value $p$ is the probability, under $H_0$, of getting a statistics as extreme as the observed value on the sample. We reject the hypothesis $H_0$, at the level $\alpha$, if $p \leq \alpha$ in favor of $H_1$. Otherwise, we reject $H_1$ in favor of $H_0$. Conversely, the $r^2$ score is used to test the quality of the entire model. It is the proportion of the variance in the dependent variable that is predictable from the predictors. Note that the $r^2$ = 1-RSS/TSS where TSS is the total sum of squares i.e., is the sum of the squares of the difference of the dependent variable and its mean. Hence, maximizing the $r^2$ is equivalent to minimizing the RSS.

So, given $k$ predictors or explanatory variables which are the probability distributions PCRW($\mathscr{P}_k$), the forward selection procedure works as follows

- Start with a null model i.e. no predictor but only an intercept. Typically, this is the average of the dependent variable;
- Try $k$ linear regression models (i.e., models with only one predictor) and chose the one which gives the best model with respect to the criterion. In our case, the one that minimizes RSS or alternatively, the one that maximizes the coefficient of determination $r^2$;
- Search among the remaining variables the one that, added to the model, gives the best result i.e., the higher $r^2$ such that all the variables in the model are significant i.e., their $p$-value is below the chosen threshold. Iterate this step until no further improvement.

### D. Validation

Since we would like to use the regression model as a prediction model (i.e., not only a descriptive one), we use Monte Carlo cross-validation a.k.a. repeated random sub-sampling validation [23]. Given a data set of $N$ points, the method simply splits them into a training subset $s_t$ and a test subset $s_v$. The model is then trained on $s_t$ and tested on $s_v$. This procedure is repeated multiple times and the results are then averaged over the splits. Note that the results of Monte Carlo cross-validation tends towards those of leave-$p$-out cross-validation [3] as the number of random splits tends to infinity. The drawbacks of this method are the possibility that some observations may never be selected for training or on the contrary, may be used at each split. Plus, the results depend on the different random splits i.e., it displays Monte Carlo variation. However, it has advantage (over $k$-fold cross validation [3]) as the proportion of the split is independent of the folds (iterations). It means Monte Carlo allows to explore somewhat more possible partitions, though one is unlikely to get all of them since there exist $\binom{N}{s_t}$ unique training subsets.

**Remark 2 (Division of a node type)** *Given a HIN $H$ with $\mathscr{V} = \{V_1, ..., V_k, ..., V_m\}$ the set of node types with $V_k = \{V_{k,1}, ..., V_{k,q}\}$, one can want to understand the "role" of each $V_{k,r}$. Let two node types $V_i$ and $V_j$ (not necessarily distinct) be the target node types and $\mathscr{E}_{\mathscr{P}}$ the set of meta-paths. Plus, let $V_i$ and $V_j$ be linked by a specific meta-path including the node type $V_k$, namely, $\mathscr{P} = V_i \cdots \xrightarrow{e_k} V_k \cdots \xrightarrow{e_j} V_j$ with $\mathscr{P} \in \mathscr{E}_{\mathscr{P}}$. We can thus construct $q$ subsets $S_{i,r} = \{v_i \in V \,|\, \phi(v_i) \in V_i \wedge \exists P = v_i \cdots \xrightarrow{e_k} v_{k,r} \cdots \xrightarrow{e_j} v_j\}$ and $q$ subsets $S_{j,r} = \{v_j \in V \,|\, \phi(v_j) \in V_j \wedge \exists P = v_i \cdots \xrightarrow{e_k} v_{k,r} \cdots \xrightarrow{e_j} v_j\}$ $(r = 1,..,q)$ such that with $v_{k,r} \in V_{k,r} \subseteq V_k$ ($v_j \in V_j$ and $v_i \in V_i$ resp.) and $P \in \mathscr{P}$. We can thus build $q$ linear regression models: one for each HIN $H_r$ formed from the node set $\{v \in V \,|\, \phi(v) \in \mathscr{V} \setminus \{V_k, V_i, V_j\}\} \cup \{S_{i,r}, S_{j,r}\}$ with meta-paths $\mathscr{E}_{\mathscr{P}} \setminus \mathscr{P}$. Analysing the vector $\hat{\beta}$ of each final model can bring some insight about the "role" of each $V_r$.*

## IV. EXPERIMENTS

We test the proposed methods on two real-worls data sets. The first one, related to FIFA WorldCup 20104 Twitter data, allows us to perform tests between target nodes with different types. The task consists in recovering the user-hashtag frequency. The second data set, related to bibliographic data, focuses on target nodes of the same types and tackles the problem of co-authorship.

### A. FIFA WorldCup 2014 Twitter data.

We present the data set on which we test the proposed method as well as the construction of the resulting graphs. Then, we report our results concerning different tests namely, the importance of meta path length, a description task and finally a recovery task.

#### *1. Data Set Description and Setup.*

The data we use is a set of tweets collected from Twitter during the Football World Cup 2014. This period extents from June 12

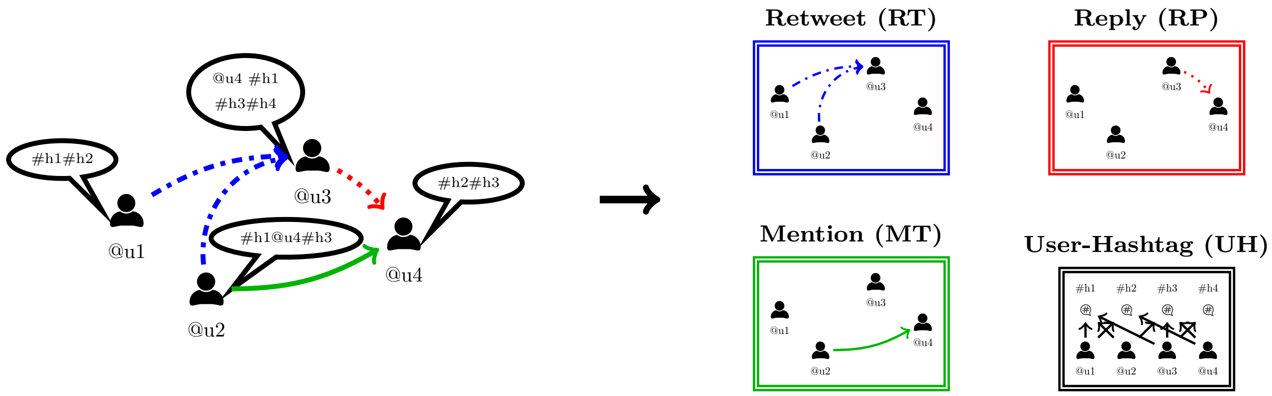


Figure 2: Illustration of the construction of graphs based on Twitter interactions where four users interact with each other through three types of interactions: retweet, reply and mention; and write some hashtags in their post. The underlying HIN is such that $\mathscr{V}$ ={users, hashtags} and $\mathscr{E}$ ={RT, RP, MT, UH}. The four graphs associated to the types of actions are displayed separately for convenience.

to July 13, 2014. Twitter allows multiple kinds of interactions between its users. Here, we consider retweet (RT), reply (RP) and mention (MT) actions plus the fact of posting hashtags (UH). The RT relationship means that a user broadcasts a tweet previously posted by another user. The RP action is simply a response tweet to another user in connection with her previous tweet. The last action considered here is the MT action. This happens when a user mentions explicitly another user in her post.

Based on these actions, we construct a HIN with two node types $\mathscr{V}$ ={users, hashtags} and four edge types $\mathscr{E}$ ={RT, RP, MT, UH} as illustrated in Fig. 2. Each node represents a user or a hashtag. We create a link from $u_1$ to $u_2$ if $u_1$ retweets, replies (to) or mentions $u_2$ and the weight of the link correspond to the number of times $u_1$ performs the specific action towards $u_2$ during the whole world cup. For the user-hashtag graph, a link exists between $u$ and $h$ if $h$ appears in $u$'s post and the weight of the link corresponds to the number of times $u$ post $h$ during the whole world cup. Note that we exclude hashtags present in the retweeted posts since in these cases, users do not write them themselves. Furthermore, considering them would provoke a trivial correlation between UH and RT-UH. All graphs are directed and weighted.

The data set contains 13,826 users and 14,392 hashtags. The RT graph is composed of 6,069 nodes and 19,495 links, the RP graph is composed of 8,560 nodes and 11,782 links and the MT graph is composed of 11,782 nodes and 60,506 links. Note that Pearson coefficient between the stochastic matrices rises to 0.1776, 0.6783 and 0.4286 for RT/RP, RT/MT et RP/MT respectively. Thus, the retweet and mention relationships are clearly correlated which may cause some problems for the proposed method, as we shall see, since it is well known that least squares method is sensitive to that. Since the data is related to the world cup, the most used hashtags of bipartite users-hashtags graph UH are those referring to the 32 countries involved in the final phase as well as those referring directly to the event (#WorldCup2014, #Brazil, #Brasil2014, #CM2014, ...). The semi finalists have the greatest in-strength (in-strength of the node $j$ is $s_j^{in} = \sum_i w_{ij}$).

### 2. *Results.*

We apply the proposed method to find if the hashtags posted by users (UH) can be explained by other relations (RT, RP, MT and their combinations). For instance, given a user $u$, explaining UH by RT-UH and MT-RP-UH means that the hashtags posted by $u$ are, to some extent, a combination of those posted by the users retweeted by $u$ and those posted by the users who received a response from users mentioned by $u$. In other words, we try to understand if, in the case of the football World Cup 2014, the probability that users post hashtags can be explained by the relations these users have with other users and the probability that these latter have to post these hashtags.

**Meta-Paths of Length 2.** We test linear regression models with all the possible combinations of meta-paths of length 2 (see Table 1). This test allows a first glimpse of the contribution of the simplest predictors. First, the more the predictors, the better the value of $r^2$. It thus could be tempting to consider them all. Nevertheless, it does not mean that all predic-

| Mod. | Meta-Path | Coef. | $p$-values | $r^2$ |
|---|---|---|---|---|
| A0 | Average : 1.8704e-05 | | | 0.2992 |
| A1 | RT-UH | 0.6273 | - | 0.3594 |
| B1 | RP-UH | 0.4291 | - | 0.2289 |
| C1 | MT-UH | 1.0289 | - | 0.4606 |
| A2 | RT-UH | 0.5795 | 0.0062 | 0.6116 |
| | RP-UH | 0.3957 | 0.0105 | |
| B2 | RT-UH | -0.3578 | 0.0612 | 0.5943 |
| | MT-UH | 1.4534 | 0.0087 | |
| C2 | RT-UH | 0.0051 | 0.0138 | 0.6111 |
| | MT-UH | 0.9391 | 0.0057 | |
| A3 | RP-UH | -0.1283 | 0.0791 | 0.6818 |
| | RP-UH | 0.0791 | 0.0113 | |
| | MT-UH | 1.1466 | 0.0111 | |

Table 1: Coefficients and $p$-values for linear regressions whose regressors correspond to meta-paths of length 2 in order to explain the user-hasthag distribution (UH). Model A0 corresponds to the null model: no predictor but one intercept that is the average of the explained variable.

tors are significant. Indeed, the analysis of the coefficients and $p$-values makes it possible to realize the correlation of some variables. In models B2 and A3, the RT-UH and MT-UH meta-paths are both present. However, the $p$-value associated to RT-UH is greater than 0.05 which states that we accept the null hypothesis for this predictor. This could be a consequence of the correlation between RT-UH and MT-UH.

In summary and as it can be seen in Table 1, the best model according to the $r^2$ and the $p$-values with threshold $\alpha = 0.05$ would be the model A2 whose predictors are RT-UH and RP-UH. The gain in the $r^2$ with respect to any other model with 1 regressor (and so simpler model) is worth it i.e., important $r^2$ improvement and not really more complexity added. This means that, for a given user, the hashtags she posts can be explained by the hashtags posted by the users she retweets with a contribution of 0.5795 and the users she replies to with a contribution of 0.3957. This model accounts for 61.16% of the variance.

**Importance of Meta Path Length.** This subsection looks at the length of the meta-paths for a given link type. More specifically, we compute, for each link type, the $r^2$ score when the only predictor is associated to a random walk of length $l = 1, \dots, 10$ repeating the same link type. For instance, for $l = 2$ and the retweet action, the predictor will be RT-RT-UH representing the hashtags posted by people who are retweeted by people who are themselves retweeted. Intuitively, the importance of a meta-path decreases with its length ($= l + 1$) since considering longer meta-paths means considering more extended neighborhoods, hence the information is more diffused. By way of illustration, the walker can attain a lot of nodes with some of them really far from the starting node.

This is corroborated in Fig.3a where we can see a tendency to decrease with respect to the meta-path length. Each link type brings a different quantity of information and the MT type is the more informative for our purpose.

Plus, this analysis exposes a characteristic of the reply dynamics: most of the time, the replies involved only two people [12]. This is reflected through the oscillations of the reply scores in Fig. 3a. The scores associated to odd length random walks are low since the walker is forbidden to return to the initial node on the penultimate step of the walk (see Fig. 4).

We also draw in black the $r^2$ scores when we do not differentiate the link types (ALL) i.e., all the link weights between to nodes are aggregated. This score is below the average score of the three specific link types. One can see that just taking the mention or retweet type is more informative than the aggregation which reinforces the relevance of differentiating the link types.

Fig. 3b shows $r^2$ scores when we combine variables of different lengths related to the same link type in the model. Actually, the $r^2$ associated to $n$ number of variables is related to the model whose predictors are all meta-paths of length smaller or equals to $n+1$ and whose the steps except the last are in the same type of links. For instance, for 3 variables, the predictors are RT-UH, RT-RT-UH and RT-RT-RT-UH (for the RT case). Again, the more the variables, the better the score. Also, the increase is not linear; the best improvement happens when we combine length-1 and length-2 variables which indicates the need to consider them together. We can also observe that scores given by the RT and MT types are really similar when considering more than two variables while there is a clear difference in the $r^2$ score for single variable. It means that their respective combinations have the same result in term of $r^2$ although the underlying semantics are different. Once again, the $r^2$ score for the aggregation is shown and is far below the other scores. This indicates that it is important to distinguish the types of links.

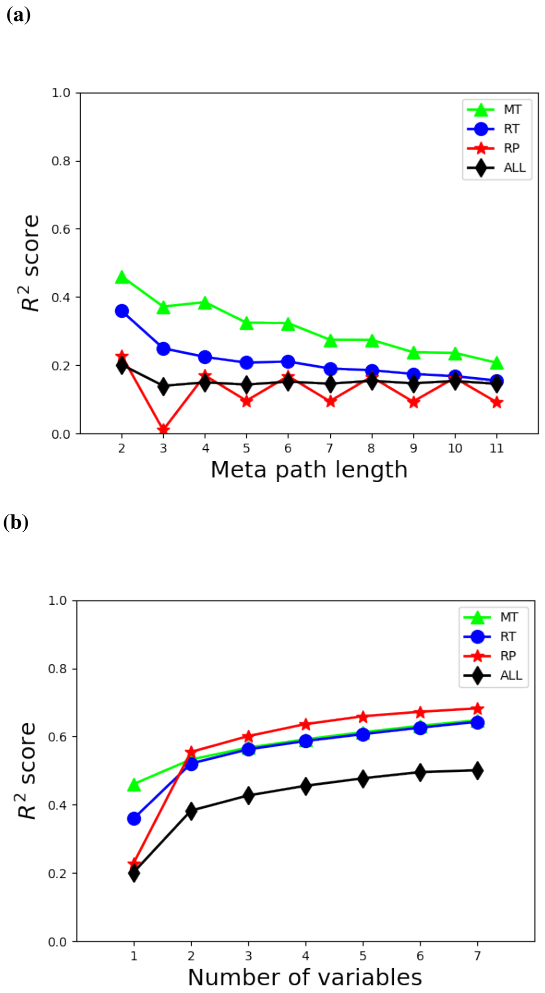


Figure 3: **(a)** Linear regression $r^2$ scores with one predictor associated to a meta-path whose length varies between 2 and 11. **(b)** Linear regression $r^2$ scores according to the number of meta-paths of the same link type (see main text for explanation).

Since it is often desirable to keep a model simple both in term of interpretability and computation time, there is a trade-off between the highest possible $r^2$ and the cost to attain it. The tests here performed tend to show that considering too long as well as too many meta-paths is not necessarily useful in our case. Indeed, the gain in the $r^2$ is not worth it considering the complexity it brings.

**Forward Linear Regression for Data Description.** We apply the proposed algorithm on the entire data set with a threshold $\alpha = 0.05$ for $p$-values. As a reminder, the procedure stops when there is no longer possible to improve the $r^2$ by adding significant regressors. Since the length of meta-paths is unbounded, the set of possible meta-paths is infinite. Here, the $k$ potential predictors are those of length less than or equal to 4. This is motivated by the test performed in the previous subsec-

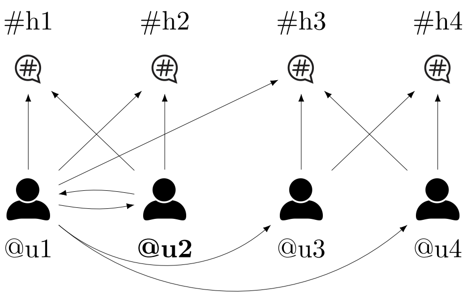


Figure 4: Typical example of reply case focused on user $u_2$. The hashtags posted by $u_2$ are $h_1$ and $h_2$. The probabilities resulting from the random walk UH starting from $u_2$ are then $[1/2, 1/2, 0, 0]^T$. For meta-path of length 2, a walker starting from $u_2$ following meta-path RP-UH has to go, with probability 1, to $u_1$ and then to $h_1$, $h_2$ and $h_3$. The resulting probabilities are $[1/3, 1/3, 1/3, 0]^T$. Now, for meta-path of length 3, the walker can not return to $u_2$ after being on $u_1$: he has to go to $u_3$ or $u_4$. But since these latter are not in connection with $u_2$ via the reply action, their hashtags are more different. This time, the probabilities are $[0, 0, 1/2, 1/2]^T$, which is far from those obtained with UH: $[1/2, 1/2, 0, 0]^T$. Consequently, the $r^2$ is really low (in this case, it is null). However, for meta-path of length 4, the walker can return to $u_2$ after being on $u_1$ so in the next step (the third step), the walker can only jump to $u_1$ who is a direct neighbor of $u_2$. The rationale is the same for longer meta-paths: for even lengths, the walker is not affected by the restriction on the penultimate step of the walk while for odd lengths, it has huge importance.

| Mod. | Meta-Path | Coef. | $p$-values | $r^2$ |
|---|---|---|---|---|
| 0 | Average: 1.8704e-05 | | | 0.2992 |
| 1 | MT-UH | 1.0289 | - | 0.4606 |
| 2 | MT-UH | 0.9391 | 0.0057 | 0.6112 |
| | RP-UH | 0.0052 | 0.0137 | |
| 3 | MT-UH | 0.8464 | 0.0062 | 0.6682 |
| | RP-UH | 0.0335 | 0.0124 | |
| | RT-RP-UH | 0.1077 | 0.0138 | |
| 4 | MT-UH | 0.8114 | 0.0063 | 0.6947 |
| | RP-UH | 0.0362 | 0.0109 | |
| | RT-RP-UH | 0.0766 | 0.0142 | |
| | RP-MT-UH | 0.0676 | 0.0143 | |
| 5 | MT-UH | 0.1974 | 0.0094 | 0.7129 |
| | RP-UH | 0.5556 | 0.0146 | |
| | RT-RP-UH | 0.0650 | 0.0125 | |
| | RP-MT-UH | 0.1591 | 0.0160 | |
| | MT-RT-UH | 0.0074 | 0.0124 | |

Table 2: Results of the forward stepwise linear regression.

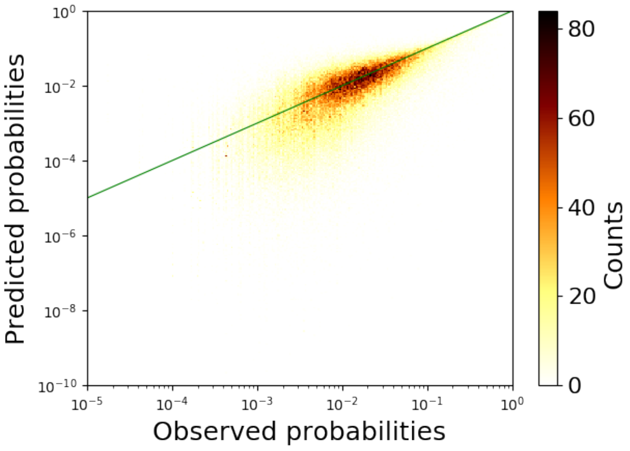


Figure 5: Density plot of observed versus estimated values for the model 5. Green line represents the perfect matching between observed and estimated data.

tion. In addition, the semantics of longer paths are less clear than shorter paths.

Results are reported in Table 2. The final model thus obtained contains five predictors related to meta-paths whose length are no longer than 3 and no intercept. This regression model accounts for 71.29% of the variance. To comfort the goodness of fit of the model, we plot in Fig. 5 the density plot in log-log scale of the predicted probabilities versus the observed ones in the data. The green line represents the ideal case where predicted probabilities match observed ones. Most of the data points fall to this line which reinforces the use of a linear model.

The best improvement with respect to $r^2$ comes with the addition of the second variable (see Mod. 2 of Table 2). The model with two predictors is actually a local extremum since the model with the best $r^2$ is the one with RT-UH and RP-UH predictors (see Table 1). Although the difference is tenuous, this allows to point two weaknesses of the method: there is no guarantee of finding the best model and the order of the variable selection is important. Note that the first two variables are part of the most direct relationships (meta-paths of length 2) which is intuitive: the direct neighborhood of a user shares common topics with her. The last meta path included in the model (Mod. 5) provokes an important change in the other coefficients. This suggests this meta-path is either correlated to other meta-paths already present in the model or the presence of outliers i.e., observation which is "distant" from other observations. It is well known that ordinary least squares method is sensitive to that.

**Forward Linear Regression for Data Recovery.** We validate the method by performing a task aiming to recover the weights of missing links. In other words, this part tries to answer to the question: is it possible to know, in a quantitative way, the way some people post some hashtags, knowing the way other people do ?

We perform Monte Carlo cross-validation with 80% of the users as the training set and obtain the vector $\beta$ for them. Then, we use it on the testing set i.e. the remaining 20% and compute the $r^2$ associated to each model. We proceed to ten splits i.e., we create ten training sets. The final models do not include the same variables as before. Not surprisingly, it depends on the 80% selected. The number of predictors is five or six. Nevertheless, whatever the training set, the meta-path MT-UH is always the first predictor to be selected. After, there is no more consensus on the second regressor but the RP-UH and RT-RP-UH always compete for the second place. Again, it is not surprising to obtain the RP-UH meta-path since, for a user, it is related to one of the closest neighbors with respect to our graph construction and very weakly correlated to the MT-UH meta-path already present in the model. Although the best $r^2$ scores of the final models reach, on average, 0.7 for the training sets, we only get, on average, a best score of 0.5 for the test sets (Fig. 6). The method seems to reach a limit. One also observes that even if a model better fits the training set, it does not mean that it will give the best recovery. Indeed, it is sometimes better to consider a model with fewer regressors, and so a lower $r^2$ for training set, to better recover.

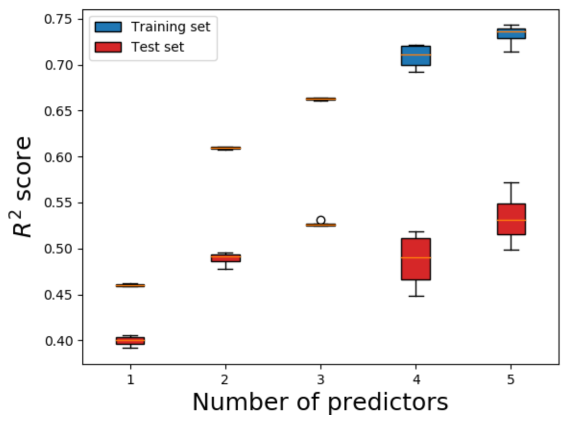


Figure 6: Boxplot of the $r^2$ scores of training sets and test sets. The training set scores increase with the number of predictors in the model while for the testing set, the scores seem to reach a threshold.

## B. Bibliographic data.

Bibliographic networks are also good examples of heterogeneous information networks since they contain multiple types of nodes and links. We here focus on scientific publications.

### 1. *Data Set Description and Setup.*

Fig. 7 illustrates an example of such networks where one can distinguish four types of nodes that is authors, papers, venues and topics; and four types of links (eight when we differentiate a type from its inverse) that is write, publish, cite and belong to.

The HIN we analyse in this article is constructed from DBLP publications [1]. The data set contains 95,855 authors with 1,537,407 co-author relationships and 186,175 papers with 1,356,893 citation relationships. The papers belong to nine distinct topics: Artificial Intelligence, Computer Graphic: multimedia, Computer Networks, Database: Data Mining: Information Retrieval, Human Computer Interaction: Ubiquitous Computing, Information Security, Interdisciplinary Studies, Software Engineering and Theoretical Computer Science. These topics are represented in the 92 venues present in the data set.

The presented method is used to find out if the co-author relationship A→P←A is correlated with other directly extractable relationships of the underlying graph. Table 3 shows the different meta-paths used in the models selected according to their semantics contrary to the previous experiment. Since there is only one directed type of links between two given types of nodes, we only mention the types of nodes to describe the meta-path.

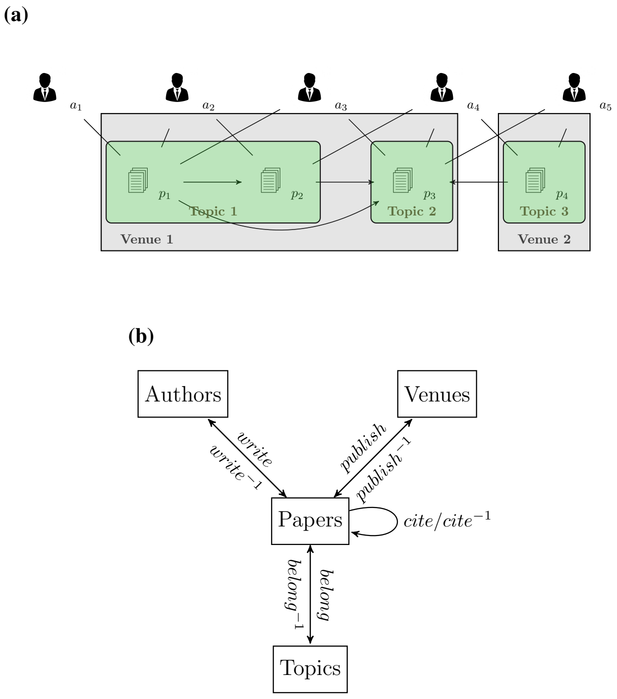


Figure 7: **(a)** Example of a bibliographic network. **(b)** Its associated network schema.

| Meta-Path | Meaning | Feature |
|---|---|---|
| **A→P←A** | **are co-authors** | |
| A→P←A→P←A | share co-authors[1] | $v_A$ |
| A→P→P←A | cite the other's paper | $v_{PP}$ |
| A→P←P←A | are cited by the other's paper | |
| A→P→P←P←A | co-cite the same paper | $v_{PPP}$ |
| A→P←P→P←A | are co-cited by the same paper | |
| A→P→V←P←A | have paper in the same conference | $v_V$ |
| A→P→T←P→A | have paper about the same topic | $v_T$ |

Table 3: Meta-paths describing some notions of proximity between authors. The Features gather some meta-paths that are similar if the direction of the arrows is neglected or alternatively, if one only considers the node types composing the meta-paths.

As mentioned, meta-paths are no longer determined by their length but selected by a more solid prior knowledge of the data. Here are given some motivations about the selected meta-paths.

- A→P→A←P→A means that two authors have written with a third common author. It represents a triangle when the AP-PA graph is projected onto A. This meta-path is the most “social”;
- A→P→P←A and A→P←P←A state for the interest of a person (say *a*) for the work of another (say *b*). It could be meaningful to think that if *a* is interested in *b*'s work and cites it, *a* is eager to communicate with *b* and even to

[1] distinct of the targeted authors

collaborate and to publish with her. The same holds if $a$ and $b$ exchange their role;

- A→P→P←P←A means that two authors cite the same paper and are thus inspired by the same ideas. This could be a good reason for a co-author relation;
- A→P←P→P←A is quite different since it states that a third person (say $c$) cites the work of $a$ et $b$ but it does not mean that $a$ and $b$ work on the same thing. So, we expect this meta-path to be less significant that the previous one, albeit the structure is fairly close;
- A→P→V←P←A and A→P→T←P←A mean that $a$'s paper and $b$'s paper are in the same venue or belong to the same topic respectively. Even if some venues can gather a lot of people, being accepted in the same venue might trigger collaborations. Plus, working on the same topic can also be a source of collaboration.

Starting from the data, we construct four matrices associated to four bipartite graphs. In particular, $AP$ where $AP_{ap}$ equals 1 when authors $a$ writes paper $p$, 0 otherwise. $PP$ where $PP_{pq}$ equals 1 when paper $p$ cites paper $q$, 0 otherwise. $PV$ where $PV_{pv}$ equals 1 when paper $p$ is published/presented in conference/venue $v$, 0 otherwise. $PT$ where $PT_{pt}$ equals 1 when paper $p$ belongs to topic $t$, 0 otherwise. These matrices are binary but it does not imply the co-author matrix ($AA$) is binary too. In order to compute the proposed variables/meta-paths, matrices are transformed into row-stochastic matrices i.e., normalized such that the sum of each line equals 1. In this setting, we can consider these matrices as transition matrices and perform random walks on it. For Fig. 7a, we have the following matrices:

$$AP = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad PV = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$PP = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad PT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the red entries (last columns and rows of each matrix) are related to the so-called hole nodes (see Sec. III A). Remark that paper $p_3$ points to the hole node in the $PP$ graph since it does not cite any paper.

Furthermore, note that for meta-paths of the form A→P→ "node type" ←P←A with "node type" in {P, V, T}, one forbids the walker to return to the same paper in his second and fourth step. It prevents us from using what we are looking for. For instance, in Fig. 7, a walker constrained by the A→P→A←P←A meta-path and having travelled through the path $a_1 \to p_1 \to a_3$ cannot return on $p_1$ at her next step but has to go to $p_3$.

### 2. *Results.*

As said, the aim of this experiment is to express the distribution of co-author relationship of all the authors in the data set by a combination of other distributions. The results are once again divided into explanatory and recovery tasks.

**Forward Liner Regression for Data Description.** Two tests are performed: first, we consider all the presented meta-paths as regressors (Table 4) and second, we aggregate some meta-paths a.k.a. features (see third column of Table 3) and utilize them into the algorithm (Table 5). We propose this aggregation because if the direction of the arrows is neglected, the meta-paths composing a feature are the same. In other words, the sequence of the node types is the same. The aim is to quantify the quality loss (if any) of the prediction when aggregating meta-paths into features.

*Meta-paths as regressors.* For the first test, we see that only three meta-paths are retained into the final model. This latter is able to explained 66,61% of the variance in the dependent variable from the independent variables. According to this model, the most significant meta-paths to explained the co-author relationship are related to the way authors share the same co-authors (some kind of transitivity[2]), cite and co-cite, plus the venues in which papers are published/presented. Meta-path related to "topic" is not included in the model.

| Meta-Path | Coefficient | $p$-value |
|---|---|---|
| A→P←A→P←A | 1.2507 | 0.0038 |
| A→P→P←A | 0.9237 | 0.0099 |
| A→P←P←A | - | - |
| A→P→P←P←A | 0.2813 | 0.0395 |
| A→P←P→P←A | - | - |
| A→P→V←P←A | 0.1539 | 0.0099 |
| A→P→T←P←A | - | - |
| $r^2$ | 0.6661 | |

Table 4: Results of the linear model for all selected meta-paths.

*Features as regressors.* When we aggregated some meta-paths into features, those related to citing the same paper and the venues are not included in the model (see Table 5). For the first one, it could be explained by the fact that only one meta-path (A→P→P←P←A) among two is imported in the first test[3]. No immediate reason is given for the absence of $v_V$ variable. Plus, this second model only accounts for 59.97% of the variance: each meta-path brings its own meaning and even if some of them seem close to each other, wanting to aggregate them is not beneficial for our purpose. Actually, we have already mentioned a "fundamental" difference between variables of $v_{PPP}$. As in the previous case, feature related to "topic" is

| Feature | Coefficient | $p$-value |
|---|---|---|
| $v_A$ | 1.2133 | 0.0028 |
| $v_{PP}$ | 1.8549 | 0.0034 |
| $v_{PPP}$ | - | - |
| $v_V$ | - | - |
| $v_T$ | - | - |
| $r^2$ | 0.5997 | |

Table 5: Results of the linear model for meta-paths aggregated into features.

[2] Transitivity of the authors-authors graph equals 0.6948.

[3] Of course, the same remark can be made for the $v_{PP}$ meta-paths and yet, $v_{PP}$ is part of the model.

not significant for the specific objective when other variables (see Table 3) are considered in the forward linear regression.

*Topics under scrutiny.* The small number of considered topics, compared with the number of papers, could partly explain why the topic meta-path is not taken into account. Indeed, only one topic is assigned to each paper so the meta-path P→T→P generate a dense "paper-paper matrix"[4] and when computing the matrix product A→P→T←P←A, any relevant information is somewhat lost.

Thus we think the meta-path A→P→T←P←A brings a too diffuse information. However, the idea of considering topics is not meaningless since an author interested in a topic is often interested for a while and therefore, has the time to collaborate with other people, who are themselves interested in the same subject. Authors writing about a same topic might partly be co-authors.

| Topic | #auth. | #pap. | #ven. | $r^2$ |
|---|---|---|---|---|
| A. I. | 41538 | 65927 | 23 | 0.5914 |
| Comp. Graph. Mult. | 25989 | 18877 | 13 | 0.6358 |
| Comp. Net. | 22374 | 30212 | 9 | 0.6321 |
| Database | 5865 | 9294 | 7 | 0.7349 |
| Hum. Comp. Inter. | 4660 | 10666 | 5 | 0.7723 |
| Info. Sec. | 5298 | 6943 | 6 | 0.7211 |
| Interdisc. Std. | 46111 | 2614 | 11 | 0.7838 |
| Software Eng. | 8147 | 20506 | 8 | 0.7222 |
| Th. Comp. Sci. | 10824 | 21136 | 11 | 0.5796 |

Table 6: Results of the different topics.

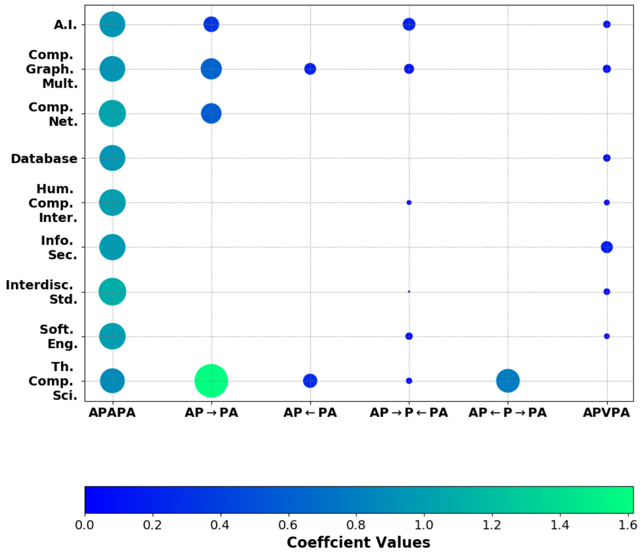


Figure 8: Coefficient values of the final models of the different domains. Size of the bullets is proportional to the coefficient values.

So, we split the data into nine subsets, each one related to one topic, and apply the method with the six meta-paths cited above i.e., all except A→P→T←P←A (see Remark 2). Results are reported in Table 6. On average, we have a better descriptive model than before: $\langle r^2 \rangle = 0.6970$ (and $\sigma$= 0.0710). This could mean that inside some topics, there are some patterns more homogeneous or frequent and we are more capable of explaining them. However, for Artificial Intelligence and Theoretical Computer Science, it is harder to find a model that fits the data well.

In Fig. 8 are reported the final models' coefficient values for the different topics. Meta-path A→P→A←P←A is selected by each topic: sharing the same co-author is the most useful to explain co-author relationship in a given topic. Meta-path A→P→V←P←A is important for 7 topics out of 9. Only Computer Networks and Theoretical Computer Science do not take it into account. Note that only Theoretical Computer Science includes A→P←P→P←A in its final model. This topic is also the only one for which A→P→A←P←A has not the greatest coefficient, it is surpassed by A→P→P←A and closely followed by A→P→P←P←A. The paper relations seem highly important for this domain.

**Forward Linear Regression for Data Recovery.** We are now interested in the recovery of link weights. Average results Monte-Carlo cross-validations are reported in Table 7. All *p*-values associated to the regressors are below the fixed threshold $\alpha = 0.05$.

| Topics | Meta-Paths | $\langle r^2_{test} \rangle$ |
|---|---|---|
| All topics | A→P←A→P←A<br>A→P→P←A<br>A→P←V→P←A | 0.5508 |
| A.I | A→P←A→P←A<br>A→P→P←P←A | 0.4994 |
| Comp. Graph. Mult. | A→P←A→P←A<br>A→P→P←A<br>A→P→P←P←A | 0.5133 |
| Comp. Net. | A→P←A→P←A<br>A→P→P←A | 0.5322 |
| Database | A→P←A→P←A<br>A→P←V→P←A | 0.7258 |
| Hum. Comp. Inter. | A→P←A→P←A<br>A→P←P→P←A<br>A→P→V←P←A | 0.7338 |
| Info. Sec. | A→P←A→P←A<br>A→P←V→P←A | 0.6509 |
| Interdisc. Std. | A→P←A→P←A<br>A→P←V→P←A | 0.7440 |
| Software Eng. | A→P←A→P←A<br>A→P→P←P←A<br>A→P←V→P←A | 0.6450 |
| Th. Comp. Sci. | A→P←A→P←A<br>A→P→P←A<br>A→P←P←A<br>A→P→P←P←A<br>A→P←P→P←A | 0.3557 |

Table 7: Results of the recovery task for the general case (all topics) and per topic.

[4] The same comment could be made for meta-path P→V→P since the number of venues is also limited - although to a lesser extent since a topic encompasses several venues. The number of non zero entries of the matrix APTPA (not really the same as PTP but the final result is encompassed in APTPA) equals 6,515,232 while for APVPA, this number raises to 3,940,634, which is still 1.6 times lower.

For Database, Human Computer Interaction and Interdisciplinary Studies, the recovery is somehow achievable in the sense that the score of the test set is almost as good as for the training set. For the other domains, the quality loss is more significant, even for Information Security and Software Engineering which have a good $r^2$ for the training set. Finally, note that for Theoretical Computer Sciences, the $r^2$ for recovery is really low and its true relevance can be somewhat even questioned (albeit the $p$-values are below 0.05). However, to be sure of its relevance, the results computed from our data set are compared with a null hypothesis model that preserve some properties of the network topology (e.g. degree distributions) but randomly reshuffles the links among the nodes. The aim is to show that degree distributions only are not enough to generate such a correlation in the data and that this correlation arises from the particular data or at least, from more involved topological properties. Indeed, results for such null models are not significant (no regressor with $p$-value smaller than 0.12) and the average score $\langle r^2 \rangle$ over 15 generations of null graphs are at most equal to 0.26.

## V. RELATED WORK

Compared to previous work, which usually focuses on undirected binary graphs, the approach we present addresses the recovery of *directed* and *weighted* links in HIN. To this end, our regression model directly estimates the weight of links without computing any intermediate ranking on these links, or applying any threshold to reduce the recovery task to binary graphs.

As previously explained, our work is based on node similarity measures and thus, is also related to link prediction. Similarity measures and link prediction have been extensively studied in the past few years. One often roughly differentiates two kinds of approaches: unsupervised versus supervised. For the first category, one often proposes different similarity measures based upon either node attributes or the topology of the underlying graph. One can further distinguish local from global indices. Local indices makes use of local neighborhood information e.g. Adamic-Adar index, Common Neighbor or Preferential Attachment Index, Ressource Allocation just to name a few. By contrast, global indices are based on global properties such as paths. These encompass Shortest Path, Katz or measures using random walks e.g. Random Walk with Restart, PageRank, Hitting Time, Commute Time and so on. Based on these aforementioned features, a plethora of supervised methods have been conceived to predict links. Amongst them, one distinguishes feature-based classification [2, 17] from probabilistic model [10, 22] and matrix factorization [16]. However, all these measures are mostly used in homogeneous networks and for a review of these methods, see [11, 13].

Recently, several measures have tackled the problem of node similarity in HIN which takes into account not only the structure similarity of two entities but also the metapaths connecting them. Amongst these measures, PathCount (PC[20]) and Path Constrained Random Walk (PCRW[8]) are the two most basic and gave birth to several extensions [4, 5, 7, 25].

Methods related to PC are based on the count of paths between a pair of nodes, given a meta path. PathSim [21] measures the similarity between two objects of same type along a symmetric meta path which is restrictive since many valuable paths are asymmetric and the relatedness between entities of different types is not useless. Two measures based on it [6, 24] incorporate more information such as the node degree and the transitivity. However, all these methods have the drawback of favoring highly connected objects since they deal with raw data.

Methods related to PCRW are based on random walks and so the probability of reaching a node from another one, given a meta path. Considering a random walk implies a normalisation and, depending on the data, offers better results. An adaptation, HeteSim [18], measures the meeting probability between two walkers starting from opposite extremities of a path, given a meta path. However, this method requires the decomposition of atomic relations for odd-length meta-paths. This decomposition allows the walkers to meet at the middle of the meta-path and at the same node type but it is very costly for large graphs. To address this issue, AvgSim [15] computes the similarity between two nodes using random walks conditioned by a meta path and its inverse. But it is mostly appreciated in undirected networks since in these cases, it is just as sensible to walk a path in one direction as in the other.

In these cited works, when the similarity scores are used for link prediction/detection, the scores are ranked and then, the presence of links is inferred based on this ranking. Also some work try to combine meta-paths but the target values to recover are binary; the networks are unweighted. At variance with these works, we set ourselves in the general framework of directed and weighted HINs. We do not use any ranking or threshold but take directly the similarity measures obtained by means of an adequate combination of PCRWs as link weights. This allows not only to perform description tasks but also, to some extent, recovery tasks.

## VI. CONCLUSION

We have considered a linear combination of probability distributions resulting from path-constrained random walks to explain, to some extent, a specific relation in heterogeneous information networks. This proposed method allows to express the weight of a link between two nodes knowing some other links in a graph. This could be useful for prediction or recommendation tasks. In particular, we have shown by working on Twitter data, that the hashtags posted by a specific user is mainly related to those posted by her direct neighborhood, especially the mention and reply neighborhood. This method has also shown that the retweet relation is not really useful for our purpose. Then we have shown the applicability of the method to bibliographic data in order to recover the co-author relationship. It has been found that (data separated into) some topics are more suited to our method and so, the functioning of coauthors seemed to differ from one topic to another.

Nevertheless, the main drawback of the method is its sensitivity to outliers. Hence, more robust least square alternatives could be envisaged such that Least Trimmed Squares or parametric alternatives.

Furthermore when there is no prior knowledge about the data, as for the Twitter data experiment, we had to provide all the meta-paths whose length is no longer than four. Even if it has been motivated by previous tests, this threshold is clearly data related. Hence, it could be interesting to build a method able to find relevant meta-paths by itself.

Finally, all data have been aggregated in time. Consequently, the chronology of the events is ignored. Since it is possible to extract the time stamp of tweets or to take into account the papers' publication date, a future work could be the integration of time by defining a random walk process on temporal graphs [14] or by counting the temporal paths [9] (plus normalisation). The walker can thus only follow time-respecting paths which can perhaps improve the quality of the model.

## ACKNOWLEDGEMENTS

This work is funded in part by the European Commission H2020 FETPROACT 2016-2017 program under grant 732942 (ODYCCEUS)

## References

[1] https://aminer.org/citation.

[2] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[3] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[4] Yuan Fang, Wenqing Lin, Vincent W Zheng, Min Wu, Kevin Chen-Chuan Chang, and Xiao-Li Li. Semantic proximity search on graphs with metagraph-based learning. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 277–288. IEEE, 2016.

[5] Mukul Gupta, Pradeep Kumar, and Bharat Bhasker. Dprel: a meta-path based relevance measure for mining heterogeneous networks. *Information Systems Frontiers*, pages 1–17, 2017.

[6] Jiazhen He, James Bailey, and Rui Zhang. Exploiting transitive similarity and temporal dynamics for similarity search in heterogeneous information networks. In *International Conference on Database Systems for Advanced Applications*, pages 141–155. Springer, 2014.

[7] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. Meta structure: Computing relevance in large heterogeneous information networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1595–1604. ACM, 2016.

[8] Ni Lao and William W Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[9] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61, 2018.

[10] Vincent Leroy, B Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 393–402. ACM, 2010.

[11] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[12] Sofus A Macskassy. On the study of social interactions in twitter. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[13] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69, 2017.

[14] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics reports*, 716:1–58, 2017.

[15] Xiaofeng Meng, Chuan Shi, Yitong Li, Lei Zhang, and Bin Wu. Relevance measure in large-scale heterogeneous networks. In *Asia-Pacific Web Conference*, pages 636–643. Springer, 2014.

[16] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

[17] Rudy Raymond and Hisashi Kashima. Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 131–147. Springer, 2010.

[18] Chuan Shi, Xiangnan Kong, Yue Huang, S Yu Philip, and Bin Wu. Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2479–2492, 2014.

[19] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2016.

[20] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 121–128. IEEE, 2011.

[21] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[22] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 322–331. IEEE, 2007.

[23] Qing-Song Xu and Yi-Zeng Liang. Monte carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 2001.

[24] Kun Yao, Hoi Fong Mak, et al. Pathsimext: revisiting pathsim in heterogeneous information networks. In *International Conference on Web-Age Information Management*, pages 38–42. Springer, 2014.

[25] Yu Zhou, Jianbin Huang, Heli Sun, and Yizhou Sun. Recurrent meta-structure for robust similarity measure in heterogeneous information networks. *arXiv preprint*, 2017.