UNIVERSITÉ DE LYON

DOCTORAL THESIS

Interactions in Information Spread

Gaël POUX-MÉDARD

Supervisors:
Prof. Julien VELCIN
Prof. Sabine LOUDCHER

Jury:
Prof. Christine LARGERON (Rapporteur)
A.R.Prof. Camille ROTH (Rapporteur)
D.R. Pierre BORGNAT
Prof. Fabrice ROSSI

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

ERIC lab
Doctoral school Computer Science and Mathematics

May 25, 2022

“ Psychohistory was the quintessence of sociology; it was the science of human behavior reduced to mathematical equations. The individual human being is unpredictable, but the reactions of human mobs could be treated statistically. [...]

The Three Theorems of Psychohistorical Quantitivity:

- *The population under scrutiny is oblivious to the existence of the science of Psychohistory.*
- *The time periods dealt with are in the region of 3 generations.*
- *The population must be in the billions for a statistical probability to have a psychohistorical validity.*

”

Isaac Asimov, *Foundation*, 1942

UNIVERSITÉ DE LYON

Abstract

Doctoral school Computer Science and Mathematics

Doctor of Philosophy

Interactions in Information Spread

by Gaël POUX-MÉDARD

Since the development of writing 5 000 years ago, human-generated data gets produced at an ever-increasing pace. This rate has been greatly influenced by technical innovations, such as clay tablets, papyrus, paper, press, and more recently the Internet. At the same time, new methods designed to handle and archive these growing information flows emerged: clay archives (Nippur, Mari), early libraries (Alexandria, Rome's Tabularia, Athens' Metroon), religious scriptoriums (abbeys, monasteries), modern libraries and, more recently, machine learning. Each of these aims at easing information retrieval.

Nowadays, archiving is not enough anymore. The amount of data that gets generated daily appeals for new information retrieval strategies. Instead of referencing every single data piece as in traditional archival techniques, a more relevant approach consists in understanding the overall ideas conveyed in data flows. To spot such general tendencies, a precise comprehension of the underlying data generation mechanisms is required.

In the rich literature tackling this problem, the question of information interaction remains nearly unexplored. Explicitly, few works explored the influence of previously generated data pieces on subsequent information creation mechanisms. In this manuscript, we develop a panel of new machine learning methods that explore this specific aspect of data generation.

First, we investigate the frequency of such interactions. Building on recent advances made in Stochastic Block Modelling, we explore the role of interactions in several social networks. We find that **interactions are rare** in these datasets.

Then, we wonder how interactions evolve over time; earlier data pieces should not have an everlasting influence on ulterior data generation mechanisms. We model this using dynamic network inference advances on social media datasets. We conclude that **interactions are brief** and that their influence typically decays in an exponential fashion.

Finally, as an answer to the previous points, we design a framework that jointly **models rare and brief interactions**. Doing so, we exploit a recent bridge between Dirichlet processes and Point processes. We improve on this advance and discuss the more general Dirichlet-Point processes. We argue that this new class of models readily fits brief and sparse interaction modelling. We conduct a large-scale application on Reddit and find that **interactions play a minor role** in this dataset.

From a broader perspective, our work results in a collection of highly flexible models and in a rethinking of core concepts of machine learning. Consequently, we open a range of novel perspectives both in terms of real-world applications and in terms of technical contributions to machine learning.

Acknowledgements

Au terme de cette thèse, tant de gens méritent d'être reconnus pour avoir su me supporter, dans tous les sens du terme. Je vais tenter ici de dépeindre ce cadre, si agréable, à l'intérieur duquel j'ai pu mener les travaux qui composent cet ouvrage.

En tout premier lieu, il convient évidemment de remercier les personnes sans qui cette thèse n'aurait pu avoir eu lieu, mes directeurs de recherche Julien et Sabine. En outre, et en dehors des convenances cette fois, j'aimerais sincèrement les remercier pour la confiance qu'ils m'ont accordée lors de ces trois années de thèse. Cette confiance, qui s'est notamment exprimée au travers de l'autonomie dont j'ai bénéficié, et d'un réel intérêt pour mes productions. Cette confiance qui m'aura permis de mener à bien mon projet de recherche sous une égide qui m'est chère, l'indépendance.

En étendant ce cadre, on retrouve les personnes qui ont réussi à rendre une vie de laboratoire pourtant ponctuée de confinements et d'isolements, stimulante et enrichissante malgré tout. Je tiens donc à sincèrement remercier Antoine, Arwa, Clément, Enzo, Habiba, Jean, Loïc, Margot, Martial, Robin, Adrien, (un autre) Antoine, Camille, Jairo, (un autre) Julien, et Stéphane.

En élargissant encore ce cadre, on trouve les personnes qui, sans avoir directement pris à part aux travaux présentés ici, leur ont tout de même permis de voir le jour, par le soutien moral qu'ils ont su m'apporter –en affection, en présence et en houblon. Merci Téo, Tony, Nora, Noémie, Nadir, Léa, Lucie, Lucas, Kenza, Delphine, Cyril, Claim, Chloé, Benoît, Audrey, (une autre) Audrey, et Amine !

Enfin, pour des raisons évidentes provenant du fond de mon cœur, qui s'ajoutent à la plupart des remerciements précédents, un grand merci à ma famille, merci Elwenn, merci Lucille, merci Maman, merci Papa, grazie Manuela, merci (une autre !) Audrey, merci mes grands-parents.

Afin que le cadre soit complet, je me dois pour finir de remercier les personnes qui, sans le savoir, m'ont mis sur la voie que j'emprunte aujourd'hui. Je me contenterai d'une citation que je juge pertinente au regard de ma situation : *Si je devais résumer ma vie aujourd'hui, je dirais que c'est d'abord des rencontres. Des gens qui m'ont tendu la main [...]. Et c'est assez curieux de se dire que les hasards, les rencontres forgent une destinée.* Ainsi, j'aimerais remercier (la même) Chloé, qui entre autres choses aura eu ce bon goût de me conseiller de lire Asimov, et qui en une simple phrase m'a mené cinq ans plus tard à rédiger ce manuscrit. Gràcies Marta, qui m'a permis à la fois de concrétiser cette petite idée née de la lecture précédente, mais également de m'avoir mis le pied à l'étrier de la recherche il y a quatre ans, et de m'y avoir donné goût par la même occasion. Enfin, merci à ces personnes dont je connais l'existence mais ignore le nom, qui en m'ouvrant une porte il y a cinq et trois ans, ou en m'en claquant une au nez il y a huit, six, cinq *et* trois ans, m'ont permis de guider mes pas là où j'en suis aujourd'hui.

Merci.

Contents

Chapter I

Introduction

I.1 General considerations

With the advent of the Internet, society realized that starting a manuscript as "With the advent of the Internet" is a banality at best, and obsolete at worst. It has now been thirty years that the amount of available online data grows exponentially. It has been twice as much that tools to automatically handle large corpora began to be developed. The impact of the Internet on societies is now a well-established fact. We know that large flows of data stream through it every second. We know the importance of developing automated means to make sense of these massive datasets. We know how crucial the understanding of the underlying mechanisms from which data emerges is. Or do we?

I.1.1 About these large flows of data

Most Internet users have at least an idea of how little of the total information flows appears on their usual platforms, be it Facebook, Reddit, Twitter, or any user generated content platform. Understanding the voices of 5 billion Internet users is not a human task. What most Internet users do not know, however, is how much data this represents. As an analogy, think of this child's dream of reading every book on the planet –which was eventually doomed after the invention of the press. Taking the 200,000 million books stored at the British Library as a good estimate of the available literature, such a task would imply reading roughly 5,000 books a day for 80 straight years. This amount of information represents a rough estimate of 150 terabytes of textual data. The task seems colossal, even by automated means – the largest textual model to date GPT-3 is trained on 45TB of text data. However, confronting today's reality, the same amount of textual data gets published on Twitter over the course of a year and a half. It remains a pretty long time, given the same amount of text gets sent over Whatsapp *every single day*. Moreover, these numbers are for text only and do not account for other types of content, such as audio, video, or images.

I.1.2 About the automated means to make sense out large corpora

To "make sense out of the data" can have various interpretations depending on the studied object. It is often used as a shortcut in scientific articles' abstracts. A quick query on any scientific search engine shows this expression systematically refers to a different aspect of data modelling: understanding tumour growth from medical reports, boosting a company's value from utility data, identifying depression in a pile of text messages, provoking Internet buzzes, etc. The common feature of these examples is that data are used as a means of an application. To "make sense" out of large datasets is to be understood as "make them usable" or "describe them in-depth". Being able to gather 150TB of Whatsapp messages a day is useless unless

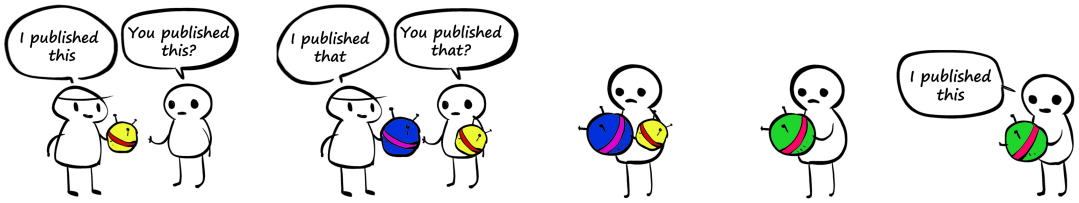

FIGURE I.1: A cartoon illustrating how the interaction between previous events can condition present events.

we have an application for it –be it descriptive or applied, medical or commercial, etc. Making sense of datasets boils down to extracting elements of interest from them. A lower-level description of "making sense" of it implies defining numerical quantities to compare data elements to each other. One of such quantities that will be extensively discussed in this manuscript is entities' group membership; a favoured way to describe datasets is to group its elements into clusters and analyse data at a more tractable level. Clusters containing different data elements are likely to tell different stories.

I.1.3 About understanding underlying data-generation mechanisms

This point makes use of the two previous ones. We would like to develop models that explain the emergence of data. For instance, I tweeted this news about the last Disney movie because I wanted to, as the result of a free choice. However, there likely is more to it. I may have tweeted because I heard a Disney song in a supermarket, because a Facebook friend talked about it a few days earlier, because I felt like eating popcorn in the dark after seeing an ad on TV and went to see a random movie, or because *someone* got influenced on her side and asked me to go. This decision has certainly been influenced by these previous exposures and their interactions (see Fig. I.1) –the extent of this influence appeals to philosophical notions about free will that are not discussed in the manuscript. Either way, the decision could be *explained* by underlying influence mechanisms –advertisement, social relations, hunger, social relations again. At the scale of individuals, understanding these processes is intractable. However, at the scale of 5 billion Internet users, statistics may be sufficient to accurately model what happens under the hood. This is what we call *information spread*. A piece of influence travels from one person to the other (people talking), from a media to a person (people browsing the Internet), or from a media to a media (news replication). Understanding the underlying mechanisms of information spread can be considered from the angle of individual transmissions: who/what spread what to who/what. On this matter, many works have modelled how pieces of information spread from one entity to another. Understanding these mechanisms is crucial for moderation purposes. For instance: how to surgically stop the spread of fake news by blocking a few spreader accounts, or by diffusing a denial from strategic spreaders?; how to nudge populations towards healthier behaviours by broadcasting the right content on the right platform at the right time –which was a central objective of Obama's Nudge Unit?; how to advertise a product using selected spreaders (or *influencers* in this case) that maximize buys?; etc.

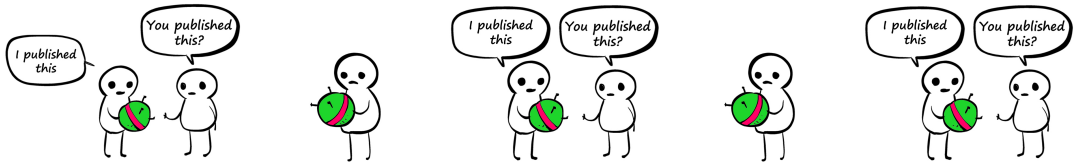

FIGURE I.2: **Independent spread assumption** — Pieces of information spread and replicate independently from each other.

I.2 Motivations

The point about understanding underlying data-generation mechanisms is the one driving the present work. As stated in the title, we propose to explore how interactions play a role in information spread. A rigorous definition of the interactions considered in this manuscript will be given further in the introduction (Section I.3.3); the key point here is that "*interacting*" is opposed to "*independent*". That said, it appears that most works on information spread consider spreading entities that are independent of each other. This is poorly illustrated in Fig. I.2.

I.2.1 Most existing models do not consider information interaction

Independent Cascade model

A seminal work that illustrates this paradigm is called the Independent Cascade model (Kempe, Kleinberg, and Tardos, 2003). In this literature, a node (or user) is said to be *infected* when she acts on a *piece of information* (retweeting a tweet, liking a post, commenting on news, etc.). In this work, an initial set of users is *infected*, meaning they act as initial spreaders for a given piece of information (e.g., a virus, a news, a tweet, etc.). Each spreader has a single chance to infect each of its neighbours in a network. After deciding whether each node gets infected by its contagious neighbour(s), the time goes forward, and the process repeats. Another model proposed in (Kempe, Kleinberg, and Tardos, 2003) is the Linear Threshold model. The process is roughly the same, except that a node gets infected after repeated exposures to contagious neighbours. At each time step, the viral charge of a node increases according to its neighbour's infection status, and to the strength of the link between them. In these processes, each node is infected conditionally to its links to other nodes in the network, but not according to the *cascade* of infections flowing through the network. Two cascades spreading simultaneously on a network are assumed to be independent.

Extensions and other independent-spreading models

Several more recent works could have illustrated that most research is oriented towards considering independent cascades spreading on networks. In (Saito et al., 2009), the authors propose to extend (Kempe, Kleinberg, and Tardos, 2003) to model the diffusion process in continuous time (instead of considering time steps). In (Larremore et al., 2012), the authors propose the avalanche cascade model, where one piece of information spreads from a single node. This work has been extended in (Poux-Médard, Pastor-Satorras, and Castellano, 2020) to highlight the importance of network structure in the description of independent diffusion processes. In (Bourigault, Lamprier, and Gallinari, 2016), the authors propose to embed spreaders in a latent space and to model information diffusion as heat diffusion in this latent space.

More elaborate models have been proposed to model diffusion processes. Some propose to consider nodes' metadata in the modelling (Saito et al., 2011), other to model the spreading processes according to the spreading content (Barbieri, Bonchi, and Manco, 2012; Du et al., 2013), or to do both jointly (Lagnier et al., 2013). In all these works, spreading processes are independent of each other.

Inferring the network from independent cascades

Reversing the problem, several works proposed to infer the underlying spreading network from the infection cascades. In (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011), the authors develop the NetRate model that considers infection timestamps to recover the underlying diffusion network using this single piece of information. With InfoPath (Gomez-Rodriguez, Leskovec, and Schölkopf, 2013b), they extend their previous method to model time-varying diffusion networks, and finally in (Gomez-Rodriguez, Leskovec, and Schölkopf, 2013a) they generalize their previous works NetRate and InfoPath in a single survival analysis framework, as well as several ulterior works based on their methodology –MoNet (Wang, Ermon, and Hopcroft, 2012), KernelCascade (Du et al., 2012). The method has then been extended to consider the spreading entities' content (Du et al., 2013; Barbieri, Manco, and Ritacco, 2017). However, here again, each spreading process is modelled independently from the others. Two cascades occurring jointly will not affect each other i.e., there is no interaction between information pieces.

I.2.2 Should we consider information interaction?

Before devoting –likely significant– efforts to consider information interaction in all the models introduced in this section, we should first conclude on its importance in spreading processes. Answering this question is the guiding thread of the whole work described in this manuscript.

Studying the role of information interaction can be broken down into three main parts –discussed at the very beginning of this manuscript. Firstly, we must **define and characterize information interactions in the large flows of data**. How to spot them, how to measure them, are there particular challenges to solve before being able to study them, etc.? Secondly, we can develop usable models **to make sense of large datasets**. Do we improve results on various tasks by considering interactions, do they have a significant role in our corpora, etc.? Finally, once information interactions have been characterized and shown to influence models' results, come the **understanding of the unveiled mechanisms** at stake. Providing interactions exist, where do they occur, can we unveil unexpected interaction patterns, do these results improve our understanding of spreading processes as a whole, etc.?

Providing insights on interactions through the lens of those guiding questions is the overall motivation of this manuscript. If interactions indeed play a significant role in information spread, the impact could be broad, as a whole class of information spread literature would have to be rethought from the perspective of interactions modelling. If interactions are shown to have lesser importance, our work would spare ulterior efforts on this problem. However, as it is the norm in research, we do not expect our conclusions to be so definitive and general. Therefore, our motivation is mainly about providing a methodology and models for interaction investigation studies, as well as about glimpsing a global conclusion on interactions in spreading processes with a case study.

I.3 Landscape of information interaction modelling

At the end of his –seminal– PhD thesis on the dynamics of diffusion networks, M. Gomez-Rodriguez states "*we have assumed contagions to propagate independently. However, this is over-simplistic, as noticed recently (Prakash et al., 2012; Myers and Leskovec, 2012). It would be interesting to relax this assumption.*" (Gomez-Rodriguez, 2013).

In this section, we review the main efforts that have been made for modelling interactions in information spread. Our goal here is to brush a global landscape of what is done on the topic without entering the technical specifics. This section does not substitute to the more technical state-of-the-art that inaugurates each chapter of this manuscript.

I.3.1 Theoretical studies

Several models have been proposed to investigate how information competes for user attention. This first section will treat theoretical works. Authors essentially set up the supposed rules for diffusion processes, simulate them, and compare them to ground-truth observations; there is no *learning* from the data.

Information overload as the consequence of micro-interactions

An early work on the topic (Weng, Flammini, and al., 2012) develops the concept of *information overload*. The overload is characterized by the entropy of the *meme* topics a user has retweeted. A meme is a piece of information that carries a semantic meaning on online platforms. Here, interactions are considered on a global scale: we do not know which are the interacting pieces of information, but we observe the overall effect of the interaction, which is a user's information overload. The authors consider users have a given affinity regarding certain memes and are exposed to them due to their ties to other people in a network. Affinity is defined as the Maximum Information Path measure (Markines and Menczer, 2009). Finally, the authors simulate a diffusion process accounting for both user interests and information overload on a synthetic Erdös-Renyi network (Erdős and Rényi, 1960). In their experiments, they achieve to manually tune their model parameters and recover aggregated measures that are similar to those observed on Twitter. This work has been followed by a large-scale quantitative study to describe the role of information overload (Gomez-Rodriguez, Gummadi, and Schölkopf, 2014). The authors find that the maximum information processing rate for tweeter users (in 2010) is 30 tweets per hour, beyond which a user is said to be overloaded. Overloaded users restrict their attention to specific information sources. The number of sources a user is likely to retweet reaches a threshold once the number of followees gets large. This study confirms the need for considering information overload in diffusion processes. However, considering information overload does not allow to explain how the overload happens. Interactions here are considered on a global scale, but the underlying agent-based interaction mechanisms remain unknown.

Modelling micro-interactions

Other works proposed to tackle information interaction modelling from a microscopic perspective. The idea is to consider each piece of information individually and observe how it relates to every other spreading entity. In (Beutel et al., 2012), the authors propose a diffusion model to infer information outbreaks under the assumption of pair interaction between pieces of information. The proportion of nodes

infected by information A, by information B, and by both A and B at time t is described using a set of differential equations –similar to SIR-like models. A node infected by a given piece of information can inhibit this node's sensitivity to the other one. They illustrate it using web browser (e.g., Firefox, Opera, Chrome, etc.) adoption: in most cases, one user will use one main browser at the time and will be much less likely to download another one once infected. The competition between the spreading information pieces is accounted for on a global scale –no network is considered. The authors derive some elementary properties of the process they defined and then show their equations approach real-world users' behaviour on the adoption of either Firefox or Chrome as web browsers. To obtain these results, the interaction parameter was tuned manually, as well as several other hyperparameters.

Modelling complex micro-interactions

Recently, the authors in (Zhu, Gao, and Zhang, 2020) proposed a complex model to account for cooperation and competition among information on social networks. This model considers interactions at a microscopic scale, meaning that the influence of each piece of information on the others is considered. The authors jointly model several attributes: user affinity, information complexity, bot spreaders (nodes that spread every information given to them), user memory, and social reinforcement. The authors define ad-hoc rules to model a spread that could consider all these parameters and specifically study the results of the simulations. The proposed modelling reproduces some emergent effects from this micro-model. However, no extensive comparison to real-world data is done.

We should *learn* models from the data

Note that we presented only a snapshot of current research on theoretical interaction modelling which is sufficient for our demonstration. Several other models tackle this problem in an analogous way (Wang et al., 2019). The fact that most works are done in a theoretical framework is decisive in the motivation of the present work. The models we just presented in this section first define the rules for interacting processes, and then tune parameters to reproduce global-scale observations on synthetic networks. However, a single effect can arise from several causes. The global observed retweeting behaviour could arise from information interacting processes, but it could also arise from other underlying processes –such as hidden ties from other media sources for instance (Myers, Zhu, and Leskovec, 2012). If most of those studies develop interesting models, we believe it is fundamental to follow the opposite process: first, we should learn the model from the data, and then analyse its characteristics, instead of first defining an ad-hoc model and only then comparing its results to the data. In the next section, we present works that tackle the problem of information interaction modelling from a machine learning perspective.

I.3.2 Data-driven studies

Clash of the Contagion

To our knowledge, Clash of the contagions (Myers and Leskovec, 2012) is the earliest attempt to learn the interaction intensity from the data. This work extensively refers to (Beutel et al., 2012), as it proposes to infer the pair-interaction parameter described in the previous section. Their main motivation is the prediction of retweets

on Twitter. It estimates the probability of retweeting an information piece given the last tweets a user has been exposed to, according to their position in the Twitter feed. To do so, they define a block-model trained on quadruplets *(tweet A, tweet B, Δt, tweet B retweeted?)* where Δt represents the time separation between A and B. Tweets are first grouped into clusters –one different cluster at each time slice–, and clusters interact with each other to determine whether yes or no the tweet B has been retweeted by the exposed user. The authors conclude that most interactions between tweets are weak, but that their overall effect cannot be neglected.

However, the method suffers various flaws. Most importantly, it is based on a questionable hypothesis on the prior probability of a retweet (in the absence of interactions). The probability of retweets in the absence of interactions is defined as equal to the frequency of retweets. Interactions are defined on this ground. We show later in this manuscript that this assumption does not hold, and thus makes conclusions about information interactions sloppy. Other technical problems have been encountered during the replication of their results. Typically, the approach does not have convergence guarantees and is ill-defined – typically because the model's "probabilities" are not constrained to be between 0 and 1. A note on our implementation to alleviate some of these problems is provided in Appendix, Section II.1.

Correlated cascades

Another work that tackles interaction modelling from a machine learning point of view proposed the Correlated Cascade model (Zarezade et al., 2017). In this work, the authors define a marked Hawkes process to model how existing pieces of information condition the appearance of ulterior pieces of information in a network. Explicitly, it models the rate at which each of two pieces of information flow through the network's edges depending on the (non-)presence of each information type. The Hawkes process models the intensity of the flows between each user, and the interaction term is tuned by a hyper-parameter β.

The final aim is to infer the latent diffusion network of an interacting spreading process. If the interaction parameter is not directly inferred as in (Myers and Leskovec, 2012), its tuning indirectly relies on observation from the data, as it plays a role in the inference of the diffusion network. The authors show that their model allows recovering global processes on real-world spreading processes on Twitter. In the conclusion, the authors formulate the open problem of learning several kernels and the interactions intensity parameter β, that we address in our work.

Further learning-based studies?

In the previous section, we only presented a snapshot of the available works on information interaction modelling from a theoretical perspective. However, to the best of our knowledge, the two models introduced in this section are quite an exhaustive list of machine learning efforts investigating this problem. This concurs with another recent survey on coevolution in information spread (Wang et al., 2019), which only cites (Myers and Leskovec, 2012; Zarezade et al., 2017) as examples of learning interactions in information spread.

On one hand, (Myers and Leskovec, 2012) proposes to model the interaction term between pair interactions, does not consider continuous-time modelling, suffers various formulation flaws and arguable assumptions. On the other hand, (Zarezade et al., 2017) models continuous-time processes and considers pair interactions but does not infer the interaction parameter, which must be tuned. This leaves plenty of space

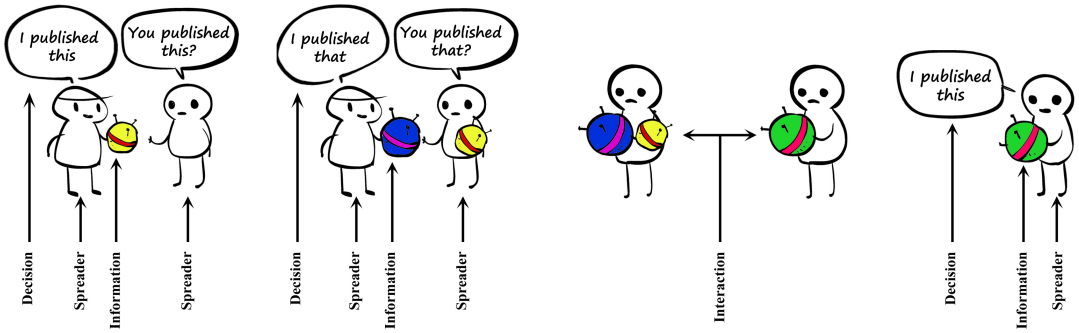

FIGURE I.3: **Illustration of the definitions** — Spreader, Information, Decision, and Interaction.

for our work to fit in. As we will discuss in the next section, we will fill these gaps by developing a framework that models the interaction parameters, over continuous times, and that can consider not only pair interactions but n-order interactions for a reasonable computational cost.

I.3.3 Definitions

In the works introduced in the section above, words such as "interaction" or "information" can have many different meanings. For instance, interactions can take place between users, pieces of information, a user and its environment, etc. In this section, we give a strict definition of key concepts present we use throughout this manuscript. In doing so, we frame our work into a specific research area. These concepts are illustrated in Fig. I.3.

Spreader — An agent that is likely to spread a piece of information to other spreaders. In the context of social media, spreaders are often referred to as *users*. For instance: a tweeter user, a member of a friends/family group, a journalist, an influencer, etc.

Information, or Entity — Any object that is susceptible to have an influence on spreaders. It carries a semantic meaning. For instance: a tweet, a meme, a song, a virus, a news article, etc.

Decision — The action of a spreader when facing an entity. The decision can be endogenous (the spreader acts on the entity alone, e.g. by sharing it, liking it, clicking on it, reacting to it, etc.) or exogenous (the spreader creates a new entity as a consequence of the first one, e.g. a denial, an answer to a mail, to a tweet, etc.). A diffusion is a set of individual decisions. For instance: a retweet cascade, an internet buzz, a Youtube trend, etc.

Interaction — When the joint effect of several entities does not equal the sum of their individual effect on spreaders. For instance, imagine a spreader that retweets (decision of retweeting noted x) a tweet A given only A with 10% chance, retweets the same tweet given only tweet B with 50% chance, but also retweets A given tweet A *and* tweet B with a 15% chance. We say there is interaction between A and B, because $P(x|A)P(x|B) = 5\% \neq P(x|A,B) = 10\%$. In this example, this interaction raises the decision probability by 5%. We sketch an illustration for this definition in Fig. I.4.

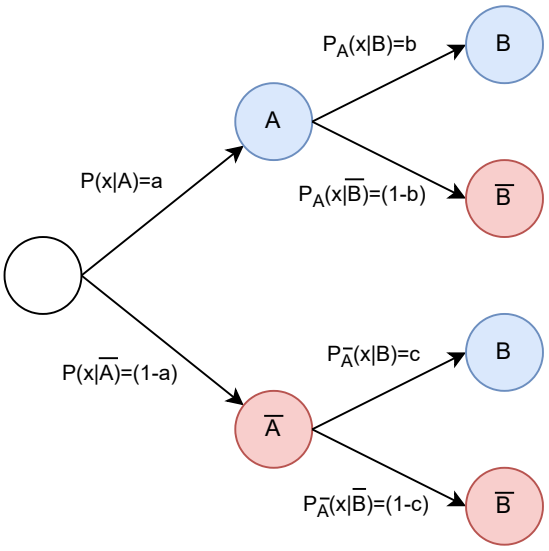

FIGURE I.4: **Definition of an interaction** — Probability tree for action x given the presence of A and B. We always have the following relation $P(x|A,B) = P(x|A)P_A(x|B)$. Now if $b = c$, we have $P(x|B) = P(x|A)P_A(x|B) + P(x|\bar{A})P_{\bar{A}}(x|B) = P_A(x|B) = P_{\bar{A}}(x|B)$. The independence relation follows: $P(x|A,B) = P(x|A)P(x|B)$. If $b \neq c$, the independence relation does not hold; there is an interaction between A and B regarding x.

I.4 About this manuscript

I.4.1 Problematic

We formulate our problem in the following terms:

How to model interactions between propagating information pieces to conclude on their significance in the spreading process?

We decompose this guiding thread into four more specific questions:

Question 1 — how frequent are interactions?
Do significant interactions occur between each spreading entity individually? Or are they rare? Can we define groups of interactions? How to model possibly rare phenomena?

Question 2 — how persistent are interactions?
Do interactions last in time? How long does the influence of a piece of information remain significant? How to unveil generic interaction temporal trends?

Question 3 — Can we efficiently model interactions?
Given an answer to our two first questions, can we develop a global model that would account for possibly rare *and* brief interactions? Could such a model handle large corpora? What challenges arise in its development, and how to overcome them?

Question 4 — Do interactions play a significant role in spreading processes?
Provided we develop a way to efficiently model interactions, we can conclude on the importance of interactions in real-world diffusion processes. In which corpus is it relevant to consider information interaction? Do interactions play a significant role in it? Do we unveil unsuspected interaction patterns?

TABLE I.1: **Contributions presented in this manuscript** — Our contributions are listed below in the order of their appearance in the text.

	SIMSBM Chap. II Q1	IMMSBM Chap. II Q1	SDSBM Chap. II Q1	InterRate Chap. III Q2	PDP Chap. IV Q3	PDHP Chap. IV Q2,Q3	MPDHP Chap. IV Q1-Q4
Self-interactions	x	x	x	x	x	x	x
Pair-interactions	x	x	x	x			x
N-interactions	x		x				
Clustering	x	x	x		x	x	x
Discrete time			x	x		x	x
Continuous time				x		x	x
Online inference					x	x	x

I.4.2 Plan and contributions

Our contributions in this manuscript are multiple. Overall, we develop a methodology to investigate the role of interactions in spreading processes. In doing so, we derive models that are based on radically different fields of machine learning. These models are applied to interaction modelling but can serve much more global purposes. All our contributions in terms of models are summarized in Table I.1.

I.4.2.a Chapter II – Stochastic Block Models (Question 1)

Publications — Section II.2.3: (Poux-Médard, Velcin, and Loudcher, 2021b)

In this chapter, we observe if there are regularities in the way interactions may happen by using recent advances in Stochastic Block Modelling (SBM). We assume that subsets of pieces of information exhibit similar interaction patterns.

Section II.2.2 — We derive a global framework that generalizes several existing Stochastic Block Models. It allows to consider an arbitrarily high number of labels as an input, and an arbitrarily high order of interaction between them, to make predictions on multi-label outputs.

Section II.2.3 — We consider a special case of this global framework. We investigate the role of interactions on several social media datasets (Twitter, Reddit, and Spotify). Our conclusions in this section are that most interactions are weak (or that significant interactions are rare), but that they play a non-negligible role on global scales. We emphasize the need for clustering pieces of information to unveil global interaction patterns.

Section II.3 — The models developed in the previous section are highly flexible but do not allow to model time. In this section, we will develop a simple way to incorporate time in the models discussed previously in the form of a Dirichlet prior on observations.

I.4.2.b Chapter III – Temporal diffusion networks (Question 2)

Publications — Section III.3: (Poux-Médard, Velcin, and Loudcher, 2021a)

In this chapter, we consider the temporal aspect of information interaction.

Section III — We develop a convex model that infers the temporal evolution of interactions influence. Typically, given two entities at different points in time, our model allows us to investigate how influential was the first entity concerning a possible action on the second one. The model, baptized InterRate, is convex and can be run in parallel. We specifically apply InterRate to interaction modelling on Twitter.

We find that the most significant interactions happen immediately after the appearance of an influential piece of information. The overall conclusion of this section is that interactions in spreading processes are brief. Typically, we find that the intensity of interactions on Twitter decreases exponentially as time goes forward. Besides, we find once again that most interactions are weak.

I.4.2.c Chapter IV – Dirichlet-Hawkes Processes (Question 3 and Question 4)

Publications — Section IV.4: (Poux-Médard, Velcin, and Loudcher, 2021c)

In this chapter, we report our steps towards a model that answers the challenges raised in the two previous chapters.

Section IV.3 — We first take the Dirichlet Process (DP) as a base, and generalize it as a special case of the Powered Dirichlet Process (PDP). Specifically, it allows to control the importance of the "rich-get-richer" assumption.

Section IV.4 — We then use the PDP and the Dirichlet-Hawkes Process (DHP) as a base to develop the Powered Dirichlet-Hawkes Process (PDHP). This model can model intra-cluster temporal interaction, meaning that information groups self-interact. It can also handle challenging situations where text is rare or where publication dynamics are intricate.

Section IV.5 — We extend PDHP to the Multivariate Powered Dirichlet-Hawkes Process (MPDHP) to model intra- *and* extra-cluster temporal interactions. The main output of this model is a temporal interaction network between clusters of documents.

Section IV.6 — Finally, we apply the MPDHP to a large-scale real-world Reddit news dataset and conclude on the role of information interaction in its underlying spreading processes.

TABLE I.2: **Codes and datasets**

Model	Link	External datasets
SIMSBM	`https://github.com/GaelPouxMedard/SIMSBM`	(Harper and Konstan, 2015) (Gutiérres-Roig et al., 2016)
IMMSBM	`https://github.com/GaelPouxMedard/IMMSBM`	(Hodas and Lerman, 2014) (Baumgartner et al., 2020)
SDSBM	`https://github.com/GaelPouxMedard/SDSBM`	(Kumar, Zhang, and Leskovec, 2019) (Clauss et al., 2021)
InterRate	`https://github.com/GaelPouxMedard/InterRate`	(Bereby-Meyer and Roth, 2006) (Hodas and Lerman, 2014) (Cao and Sun, 2019)
PDP	`https://github.com/GaelPouxMedard/PDP`	(Clauss et al., 2021)
PDHP	`https://github.com/GaelPouxMedard/PDHP`	(Baumgartner et al., 2020)
MPDHP	`https://github.com/GaelPouxMedard/MPDHP`	(Baumgartner et al., 2020)

I.4.3 Reproducible research

"Non-reproducible single occurrences are of no significance to science", in *The Logic of Scientific Discovery* (Popper, 1935)

"We may say that a phenomenon is experimentally demonstrable when we know how to conduct an experiment which will rarely fail to give us statistically significant results", in *The Design of Experiments* (Yates, 1935)

"Improving the reliability and efficiency of scientific research will increase the credibility of the published scientific literature and accelerate discovery.", in *A manifesto for reproducible science* (Munafò et al., 2017)

All the experiments presented in this manuscript are available on GitHub. The repositories contain the datasets along with the Python implementation used for running the experiments. Note that datasets which are not referenced here have been gathered by us –Spotify, PubMed, Reddit. We reference the material used for each of the models presented throughout this manuscript in Table I.2.

Chapter II

Stochastic Block Models – Interactions are rare

Abstract

A straightforward way to represent interactions is to embed them as a network. Entities are nodes of this network, and interactions between these entities are link between these nodes. These links can be of diverse types; they are labelled. The last years have seen a regain of interest in stochastic block modelling of such labelled networks, which canonical decomposition-based methods are not fit to tackle.

Section II.1, we first consider static stochastic block models and their use for interaction modelling. Existing models are not fit for modelling interactions: the number of entities that can interact is limited and higher-order interactions are not allowed, which is extremely restrictive when it comes to modelling highly interacting systems.

Section II.2, to answer these limitations, we show in Section II.2.2 that most of the state-of-the-art models are all special cases of a global framework, the Serialized Interacting Mixed membership Stochastic Block Model (SIMSBM). This generalization now allows modelling an arbitrarily large context as well as an arbitrarily high order of interactions.

We then consider a special case of SIMSBM in Section II.2.3 to model interactions between entities. This particular iteration is denoted by SIMSBM(2), or Interacting Mixed Membership SBM (IMMSBM). We investigate the role of interactions between entities (hashtags, words, memes, etc.) and quantify their importance in several real-world datasets.

Section II.3, we introduce a simple way to incorporate dynamic modelling into SIMSBM in the form of a temporal prior on the model's parameters. The proposed approach relies on the single assumption that dynamics are not abrupt. We demonstrate the interest of our method on several synthetic experiments and four real-world datasets.

Section II.4, we report our first conclusion: **interactions are sparse**. Significant interactions take place only between a limited fraction of cluster pairs, and between an even smaller fraction of entity pairs. However, their overall impact increases the predictive accuracy of the models. This underlines the necessity of considering clusters of entities to efficiently model such sparse interactions.

Published works: (Poux-Médard, Velcin, and Loudcher, 2021b) in Section II.2.3

II.1 Introduction

II.1.1 Motivation

Social networks such as Facebook, Reddit or WhatsApp let individuals share and compare ideas. Modelling the mechanisms of these exchanges can help us understand why and how various pieces of information (e.g., hashtags, memes, ideas, etc.) flow through communities. We refer to these pieces of information as *entities*. In particular, we suppose that entities can interact with each other. Someone's opinion on a given entity might be influenced by previous entities this person has been exposed to; we say there is an interaction between entities. For instance, a customer that bought a smartphone might be interested in side accessories such as headphones or selfie sticks, but a customer that bought a smartphone *and* a camera lens extension might be more interested in buying a professional camera. An approach without interactions would be less successful here, since each product can lead to a greater number of different recommendations. Considering interactions would narrow the field of possible outcomes. The same line of reasoning can be applied to the prediction of retweets (user exposures to tweet A and tweet B affects the retweet probability of C), music playlist building (which song should be added to a playlist given the last songs a user listened to), detection of controversial posts (which combinations of words trigger which answers), etc. Our goal is to infer such underlying relations between entities.

Up to now, little work has been done on investigating the role of interacting entities in users' decisions (retweet, share, comment answer, etc.), or *outcomes*. Several previous works on information diffusion theory consider a user acting on an isolated piece of information (Pastor-Satorras et al., 2015; Poux-Médard, Pastor-Satorras, and Castellano, 2020). On some occasions, theoretical frameworks have been developed to investigate how the presence of concurrent pieces of information affects the action a user exerts on them in a network (Beutel et al., 2012). However, a fundamental question that remains unanswered is *how* pieces of information interact in the informational landscape.

II.1.2 Overview of the proposed approaches

Representing interactions as a network

A straightforward way to represent interactions is to embed them as a network. Entities are nodes of this network, and interactions between these entities are link between these nodes. Given interactions can give rise to various outcomes (e.g., retweet of A, like of B, etc.), there must be one link per possible outcome – the links are labelled.

In real-world applications, the number of interacting entities can be considerable. Modelling each individual labelled link between all of these entities is infeasible in practice. However, the task becomes tractable if we suppose that sets of nodes behave similarly with respect to other sets of entities. The idea is to infer such sets, or *clusters*, and model only the labelled ties between those. A schematic representation of these assumptions is proposed in Fig. II.1.

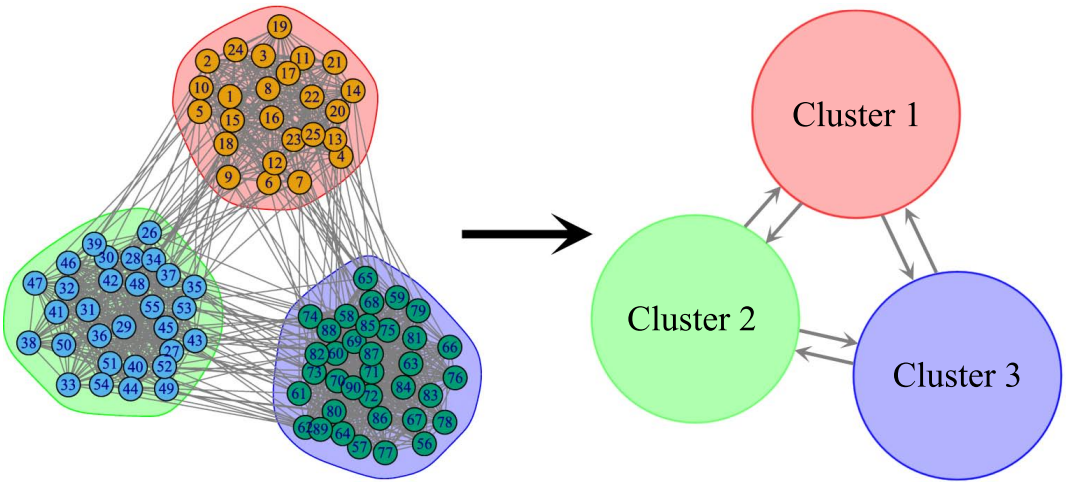

FIGURE II.1: **Illustration of the stochastic block modelling** — (Left) A set of 100 nodes is densely connected. There is a total of 100x100 links between these nodes to model. The SBM assumption is that sets of nodes behave similarly – here the nodes that share the same background colour (Right) The SBM approach proposes to model only the relations between these sets of nodes instead of each pair. In this case, it leaves us with 3x3 ties to model (we count the self-ties), as well as the node's membership.

Spotting regularities in the interactions

In general, clustering is a core concept of machine learning. Among other applications, it has proven to be especially fit to tackle real-world recommendation problems. A recommendation consists in guessing an outcome given a certain context. This context can often be represented as a high dimensional set of input entities. On retail websites, for instance, the context could be the ID of a user, the last product she bought, the last visited page, the current month, and so on; the outcome would be whether this user buys a given product. Clustering algorithms look for regularities in these datasets to reduce the dimensionality of the input context to its most defining characteristics. Continuing the online retail example, a well-designed algorithm would spot that a mouse, a keyboard, and a computer screen are somehow related buys and that the next buy is likely to be another computer device. Besides, as stated before, subsets of users are likely to share a similar interest in a given product if their buying history is similar. We can define such groups of people that share similar behaviours using clustering algorithms; this is called *collaborative filtering*. One of the most widely used approaches to perform this task efficiently relies on tensor decomposition.

Tensor decomposition approaches

Tensor decomposition approaches provide a variety of efficient mathematical tools for breaking a tensor into a combination of smaller components. One of the most popular tensor decomposition methods is Non-negative Matrix Factorization (NMF). Its application to recommender systems has been proposed in 2006 on Simon Funk's blog for an open competition on movie recommendation (Koren, Bell, and Volinsky, 2009). The underlying idea is to approximate a target 2-dimensional real-valued observations tensor $D \in \mathbb{R}^{+I \times J}$ (a positive real matrix in this case) as the product of two lower dimensional matrices $W \in \mathbb{R}^{+I \times K}$ and $H \in \mathbb{R}^{+K \times J}$ such that $D = WH$. This approach has seen numerous developments, such as an algorithm allowing its

online optimization (Fung, Li, and Cheung, 2007; Cao et al., 2007). Thanks to its low computational cost, this method is still today at the core of many real-world large scale recommender systems. However, a major drawback of NMF is that it can only consider two-dimensional data. Several extensions have been proposed to consider n-dimensional data. A straightforward generalization of NMF is the Tensor Factorization, that generalizes NMF to infer an n-dimensional matrix $D \in \mathbb{R}^{I \times J \times \dots}$ as the product of a core tensor $C \in \mathbb{R}^{K \times L \times \dots}$ with n smaller matrices $M_1 \in \mathbb{R}^{I \times K}$, $M_2 \in \mathbb{R}^{J \times L}$, and so on, such that $D = CM_1M_2...M_n$ (Karatzoglou et al., 2010). This approach allows to consider a larger context as input data. Several variants have been proposed based on similar ideas (Hidasi and Tikk, 2012; Bhargava et al., 2015).

Another popular decomposition method is the Singular Value Decomposition (SVD) (Klema and Laub, 1980). It generalizes the concept of eigenvalue decomposition to non-square matrices. As NMF, it has been used to a great extent in recommender systems. The idea is to approximate a positive real-valued matrix $D \in \mathbb{R}^{+I \times J}$ by the product of three matrices $U \in \mathbb{R}^{+I \times K}$, $S \in \mathbb{R}^{+K \times K}$ and $V \in \mathbb{R}^{+K \times J}$. Geometrically, SVD can be interpreted as the decomposition of the initial transformation matrix D as the composition of 3 elementary operations: one scaling S and two rotations U and V.

Yet another class of tensor decomposition is called Tensor rank or Canonical Polyadic (CP) decomposition. It is at the base of several popular decomposition methods that consider a sum of rank-one matrices instead of a product decomposition. In this case, an n-dimensional tensor is approximated as the sum of rank-one tensors (Harshman, 1970; Carroll and Chang, 1970). Several extensions based on CP have been proposed (Acar et al., 2010; Filipović and Jukić, 2015).

Limitations of tensor decomposition methods

All the decomposition methods introduced in the previous paragraph are based on the linear decomposition of a real-valued tensor D, which makes them unfit to tackle discrete problems. These methods can efficiently infer continuous outputs (the rating of a movie as in (Koren, Bell, and Volinsky, 2009) for instance, or the number of buys of a product) but must be tweaked to consider discrete outputs (the next buy on an online retail website for instance). In this case, a possible approach consists in mapping all possible discrete outputs as a continuous variable. This is straightforward as in the case of movie ratings, because the set of possible ratings (1, 2, 3, ...) can be ordered on a continuous scale. However, for recommending one of several products (mouse, keyboard, computer, ...), the mapping of possible outputs to a continuous value is not trivial. To consider discrete data in decomposition methods, we can add another dimension to the input tensor, whose size equals the number of possible outputs. Then, the algorithm optimizes the model based on the frequency of each of those items. This trick induces a strong bias and increases the complexity of the algorithm.

Bayesian network modelling

To answer this problem, recent years have seen a growing interest in the literature on Stochastic Block Modelling (SBM). The core idea is to represent the data in the form of a network and apply Bayesian network inference methods to the resulting graph. In particular, these methods assume a block structure. Each node of the network is associated with a block, and blocks relate to each other through a latent block interaction network, which can be labelled. The labels correspond to each possible

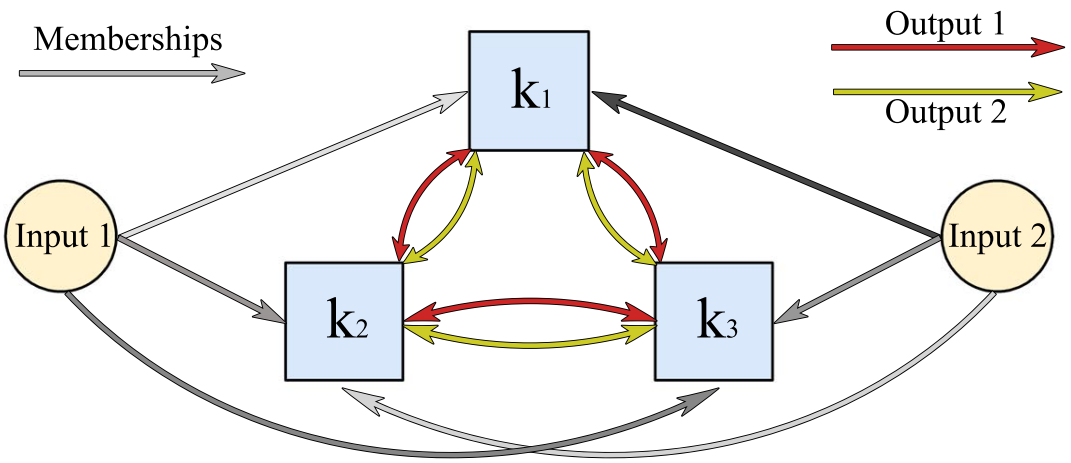

FIGURE II.2: **Illustration of the SIMSBM principle** — Two input entities are embedded as nodes, linked together by labelled edges. Edges represent the probability of an output given a combination of nodes. Instead of directly modelling such links, we assume a block structure to this network. Nodes belong to clusters to a certain extent, and only the labelled links between these clusters are modelled.

output, and must not be mapped on a continuous scale, as in (Carroll and Chang, 1970; Klema and Laub, 1980; Koren, Bell, and Volinsky, 2009). This makes the model fit to explicitly consider problems such as movie recommendation without the need for mapping movies as an additional dimension in the observation tensor. Besides, learning does not rely on linear algebra decomposition methods, but on Bayesian learning. It provides greater interpretability of the results and relies on solid statistical foundations; the model actually learns from the data. Besides, Bayesian modelling allows us to incorporate *a priori* knowledge on the data –and we will make extensive use of it in Section II.3.

We represent a schematic example of the type of Stochastic Block Model we consider in this chapter in Fig. II.2. In this case, we have two input entities that are embedded as nodes. They are linked together by labelled edges, which represent the probability of an output given the entities' combination. Instead of directly modelling such links, we assume a block structure to this network. Nodes belong to clusters to a certain extent, and only the labelled links between these clusters are modelled.

We extensively review this class of models in the sections below and present our proposed improvements to tackle interaction modelling. Explicitly, we first generalize the existing labelled stochastic block models so that they can model high order interactions, arbitrary large input contexts, and provide predictions on as many output labels as wanted (Section II.2). In a second time, we design a temporal plug-in for our model. It allows to model temporal variations of the block model's parameters with high accuracy at a minimal cost –in terms of data needed, simplicity of use and numerical complexity (Section II.3).

II.2 Static interactions

II.2.1 State of the art, limitations, and contributions

II.2.1.a Modelling static interactions

Accounting simultaneously for multiple pieces of information is motivated by numerous descriptive studies on multimodal networks structure (Sun and Han, 2012; Shi et al., 2016; Huan et al., 2017; Rashed et al., 2020). Typically, in (Sun and Han, 2012), the authors study interaction between multiple entity types *via* a heterogeneous network representation and define clusters of entities based on the structural properties of the resulting graph. However, as pointed out by the authors, this method is heavily influenced by the structural clustering method used –in this case a meta-path-based clustering (Sun, 2011). Moreover, defining edge weights in heterogeneous graphs is subjective and requires additional learning algorithms.

As seen in the introduction, several works proposed to model the diffusion of information as the result of an interaction between entities (Beutel et al., 2012). Following a similar idea, (Myers and Leskovec, 2012) investigated interactions between contagions on Twitter. The authors aim to find the interaction factors between different tweets in activity feeds. Their findings suggest that interactions between tweets play a determinant role in their retweet probability. The authors assume that there is an inherent virality for every tweet (that is an inherent probability to be retweeted) computed from the frequency of retweets, to which is added a small interaction term.

(Myers and Leskovec, 2012) paves the way to model interactions in information spread but presents various limitations that we are alleviated by using Stochastic Block Models. Firstly, the method proposed by the authors makes predictions solely based on tweets that have been observed in the feed of a given user. It therefore limits the application range of the model uniquely to systems based on the retweets (or share) concept, where information has to appear first to be spread. This model is hardly applicable to systems that are based on exogenous reactions (e.g., online forums, playlist building and recommender systems) where information can appear as a consequence of different entities ("Capital" + "Netherlands" $\rightarrow$ "Amsterdam"). An SBM-based modelling allows outcomes that are different from the input entities. Secondly, interaction is defined as a modulation of the frequency of retweets of a given tweet in any context (i.e., $P(X) = f_X + \Delta P_{inter}(X)$). We argue this can lead to false conclusions about interactions. Imagine that interactions lead to a shift of $\Delta P(X)$ on the actual virality V_X of a retweet. This interaction happens in a fraction s of all observations of a given tweet being retweeted. The virality $V_{X,hyp}$ as defined in (Myers and Leskovec, 2012) equals the frequency of retweets: $V_{X,hyp} = V_X(1-s) + (V_X + \Delta P(X))s = V_X + s\Delta P(X)$, which is by construct larger than the actual virality of a retweet by a term $s\Delta P(X)$. Therefore, defining interaction according to this quantity is wrong. Virality needs to be inferred by the model at the same time as the contribution of interactions to be properly defined, which the proposed SBM-based modelling allows.

II.2.1.b Stochastic Block Models

Stochastic Block Models

As already stated, recent years saw a growing interest for Stochastic Block Models (SBM) to tackle collaborative filtering problems in recommender systems (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019; Poux-Médard et al., 2021). These models first cluster input entities together and then analyse how clusters relate to each other. Each input entity can either be associated to one cluster (Single-Membership SBM, or SBM) (Holland, Laskey, and Leinhardt, 1983; Guimera, Llorente, and Sales-Pardo, 2012; Funke and Becker, 2019) or to a distribution over available clusters (Mixed Membership SBM, or MMSBM) (Airoldi et al., 2008). While the single membership SBM has been successfully applied to a wide range of problems (Guimera, Llorente, and Sales-Pardo, 2012; Guimerà and Sales-Pardo, 2013; Cobo-López, Godoy-Lorite, and Duch, 2018; Funke and Becker, 2019), inference is done using greedy algorithms, typically simulated annealing, making it unfit for large scale real-world applications (Godoy-Lorite et al., 2016).

Mixed Membership Stochastic Block Models

The Mixed-Membership SBM (**MMSBM**) is a major extension of Single-Membership SBM that has been proposed in the seminal work (Airoldi et al., 2008). In the frame of collaborative filtering for recommender systems, (Godoy-Lorite et al., 2016) proposed the **Bi-MMSBM** extension. This formulation generalizes the Matrix Factorization model. In this approach, entities of different type (e.g., users and movies in this case) are grouped into distinct clusters whose interaction result in an outcome (e.g., a rating on a movie). This model has later been extended as the **T-MBM** to consider triples of input entities instead of pairs (Tarrés-Deulofeu et al., 2019). It assumes that all the entities in a given triplet are linked together by a given label. This boils down to assuming data can be represented in the form of a tripartite network instead of a bipartite network.

The need for a more global framework

However, the existing literature does not answer all the challenges inherent to interaction modelling.

First, these models allow considering only interaction pairs, which are arguably not exhaustive enough to correctly model the interaction processes at stake (Steck and Liang, 2021). As stated in the introduction, interactions can involve an arbitrarily large number of entities. For instance, a Twitter user might want to retweet an entity based on its content, but also on the previous tweets she has been exposed to, the hour of the day, the last tweet she published, etc. No existing MMSBM variation proposes a solution for considering an arbitrarily large number of interacting entities.

Second, none of the existing models consider the case where interacting entities are of the same type. Entities of the same type carry the same semantic meaning (e.g., two movies are of the same type "movie", but a movie and a director are of different types). When considering input entities of the same type, permutation-invariant clustering must be accounted for: the probability of an output given entities (A, B) should not differ from the probability of this output given (B, A). This may not hold when the order of entities' apparition is considered, which is not the case here.

We consider as input an unordered set of entities, hence the need for permutation-invariant modelling.

Considering the interaction between entities of the same type is novel and important. When ignoring it, a user on Netflix is predicted to like a given movie because she is partially part of the group that liked the movie A and partially part of the group that disliked movie B, but all those groups are independent of one another (Koren, Bell, and Volinsky, 2009; Godoy-Lorite et al., 2016). The friendship between individuals is determined based on the independent groups of friends they belong to, but not based on the joint belonging to various groups (Airoldi et al., 2008; Jamali, Huang, and Ester, 2011). A drug might interact with another one, but the joint interaction of two drugs on a third one cannot be investigated (Guimerà and Sales-Pardo, 2013). Typically, in (Godoy-Lorite et al., 2016), embedding pieces of information of the same type in a bipartite graph seems irrelevant: an entity should not interact differently with another one (or belong to different clusters) because it is on the left or the right side of this bipartite graph. Instead, we need to enforce a permutation-invariant interaction.

II.2.1.c Contributions

In this section, we present the Serialized Interacting Mixed Membership Stochastic Block Model (SIMSBM). The SIMSBM is a global framework that generalizes several state-of-the-art models which tackle the problem of discrete recommendation by a Bayesian network approach –including but not limited to (Airoldi et al., 2008; Koren, Bell, and Volinsky, 2009; Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019) and the IMMSBM model presented in Section II.2.3.

First, we present the proposed framework and develop an Expectation Maximization (EM) optimization procedure (Section II.2.2.a). Then, we review previous works on Stochastic Block Models that are used in collaborative filtering and detail for each one how to recover it as a special case of our framework (Section II.2.2.c). Experiments are then conducted on 6 real-world datasets and compared to standard baselines of the literature (Section II.2.2.d). We show our formulation allows us to obtain better recommendations than existing methods either by adding layers to the modelled multipartite graph or by adding higher-order interaction terms in the modelling.

Then, we proceed to an in-depth study of a special case of the SIMSBM applied to interactions modelling, named IMMSBM. We consider the case where two input entities of the same type interact with each other and influence the probability of an output (Section II.2.3.a). We consider four real-world datasets that are expected to comprise underlying interaction processes: Twitter (entities are tweets and outcomes are retweets), Reddit (entities are words in a post and outcomes are words in an answer), Spotify (entities are songs in a user-generated playlist and outcomes are songs added next) and PubMed (entities are symptoms and outcomes are diagnosed diseases) (Section II.2.3.c). We compare to similar works that do not consider interactions; we also discuss and discard a core assumption made in (Myers and Leskovec, 2012) on interaction modelling in information spread (Section II.2.3.d). We investigate in detail the role of interactions in the proposed corpora. Finally, we conclude that significant interactions between spreading entities are rare (Section II.2.3.e).

II.2.2 SIMSBM – A global MMSBM framework

II.2.2.a SIMSBM – Serialized Interacting MMSBM

Overall idea

The goal of the SIMSBM is to recommend an *output entity*, that is one o in O possible outputs entities, given a context –for instance, o can take any label in $O = \{1, 2, 3\}$. To do so, it considers data in the form of a multipartite network. A multipartite network is a generalization of the bipartite network (two layers of nodes with no intra-connection within layers) to any number N of layers.

The network's nodes are the context elements. They are called *input entities* f_n, each of which is one of F_n possible input entities –for instance, f_1 can take one value in $F_1 = \{A, B, C\}$, f_2 can take one value in $F_2 = \{D, E, F\}$, etc. One hyper-edge between the N input entities (or nodes) of a context represents the probability of a given output entity o, that is $P(o \in O | f_1 \in F_1, ..., f_N \in F_N)$.

The SIMSBM clusters input entities (or nodes) together and infers edges between these clusters to form a smaller multipartite network. Each node f_n is associated with a certain extent to each cluster k_n among K_n possible clusters for layer n. The adjacency matrix of this network is noted $p(o \in O | k_1 \in K_1, ..., k_N \in K_N)$ –note the lowercase p for the clusters-interaction network. This projection is illustrated in Fig. II.3. The notations used throughout this section are given in Table. II.1.

Input data

Formally, input data of the SIMSBM is noted R°. Its entries are tuples of form $(\vec{f}, o)$. The vector $\vec{f} = (f_1, ..., f_N)$ represents a context, whose entries f_n are the input entities that can be one of F_n possible input entities; outputs are designated by o and can be any of the O possible outputs. Each F_n and O are unordered sets containing the labels each f_n and o can take. Each input entity is represented as a node in a layer of the multipartite network (circles in Fig. II.3). The number of layers of the multipartite graph is then $N = |\vec{f}|$; each layer n comprises F_n nodes. For each node in each layer, the SIMSBM infers a vector $\vec{\theta}_i \in [0, 1]^K$ that represents its probability to belong to each of K possible clusters (i.e., associate each circle to a distribution over the squares in Fig. II.3).

Input entities are of a given *type*; entities of the same type carry the same semantic meaning and are drawn from the same set of possible values. We note $a(f)$ the function that associates an input entity f to its given type, and $K_{a(f_n)}$ the number of available clusters for this type. Entities of the same type can interact with each other but are forced to share the same cluster membership matrix. For instance, consider a user rating a movie according to the pair of actors starring in it: the context vector takes the shape $(user, actor1, actor2)$. In this example, one entity is of type "user", and the two others are of type "actor", and each of the three entities is embedded in its own layer of a multipartite graph. When creating clusters, the SIMSBM enforces that the two layers of type "actor" share the same membership matrix, while the layer accounting for the type "user" has its own membership matrix. This structure is exactly the same as the one depicted in Fig. II.3. This is needed to get results that are permutation independent, meaning in our example that $P(o|(user, actor1, actor2)) = P(o|(user, actor2, actor1))$. Our formulation as a multipartite network allows us to consider higher-order interactions, in contrast to (Godoy-Lorite et al., 2016) which only considers pair interactions.

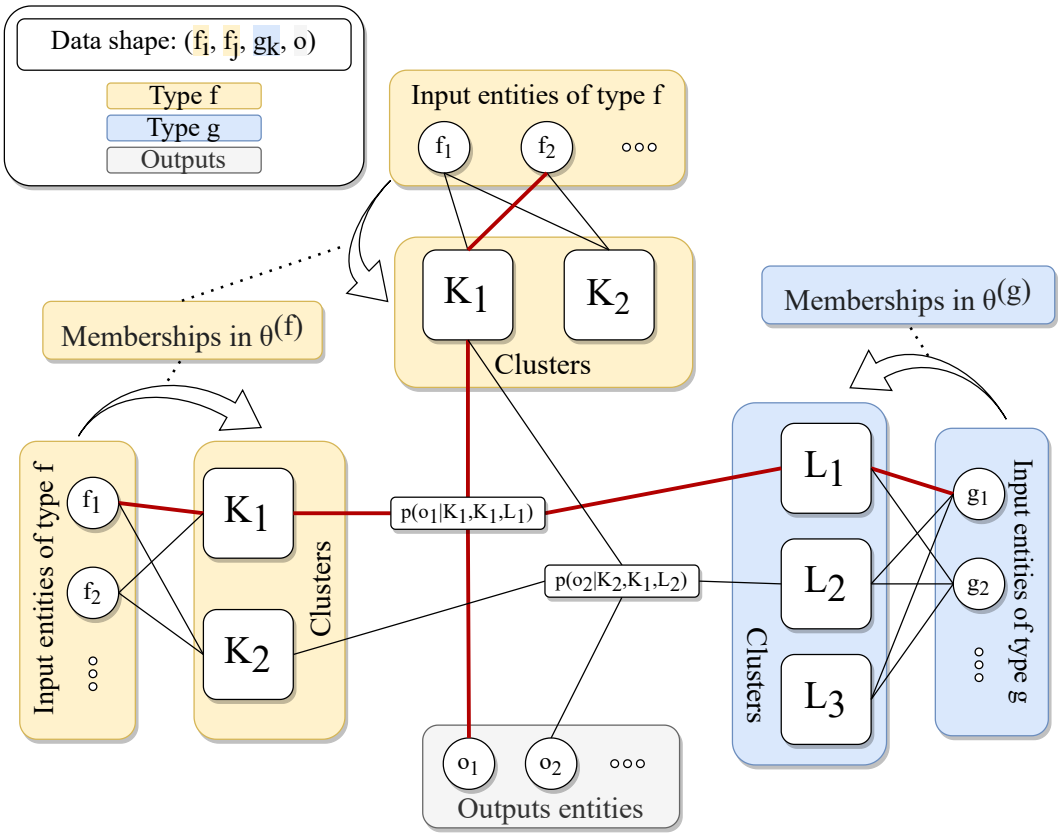

FIGURE II.3: **Illustration of the SIMSBM** — For input entities of type f and g, where entities of type f interact with each other as pairs and entities of type g do not interact with each other. The membership of an entity f_i of type f to a cluster K_n is encoded into the membership matrix entry $\theta^{(f)}_{f_i,K_n}$. The interaction between clusters is embedded in a multipartite network, whose adjacency tensor is $\mathbf{p}$. A weighted edge between several clusters and one output represents the probability of this output given the context clusters –only two such edges are represented here. **In red**, we represent the probability of o_1 given f_1 belonging to K_1, f_2 belonging to K_1 and g_1 belonging to L_1, which is equal to $\theta^{(f)}_{f_1,K_1}\theta^{(f)}_{f_2,K_1}\theta^{(g)}_{g_1,L_1}p(o_1|K_1,K_1,L_1)$.

Model parameters

The membership of an entity must sum to 1 over all the $K_{a(f_n)}$ available clusters, hence the following constraint:

$$\sum_{k}^{K_{a(f_n)}} \theta^{(a(f_n))}_{f_n,k} = 1 \ \forall f_n \tag{II.1}$$

Once the node membership is known, the SIMSBM infers the clusters multipartite network, whose weighted hyper-edges stand for the probability of an output o given a combination of clusters.

Note that there are two possible views on this: we can consider that each possible o is associated with its own multipartite network, or that each o is a node in an additional output layer, as in Fig. II.3. These views are strictly equivalent.

The hyper-edge corresponding to the clustered context $\vec{k} = (k_1, ..., k_N)$ associated with the output o is written $p_{\vec{k}}(o) \in \mathbb{R}^{K_{a(f_1)}\times...\times K_{a(f_N)}\times O}$ –it is one entry of the multipartite network's adjacency matrix. Note that the clustered context $\vec{k}$ can take any

TABLE II.1: Notations for the SIMSBM

f_n	An input entity, can take any value in F_n
F_n	Set of possible input entities for layer n
$\vec{f}$	Context vector $(f_1, ..., f_N)$
o	Output entity, can take any value in O
N	Number of input layers $\lvert\vec{f}\rvert$
R°	Data, a list of (N+1)-plets $(f_1, ..., f_N, o)$
$a(f_n)$	type of entity f_n
$K_{a(f_n)}$	Number of available clusters for type $a(f_n)$
$\theta^{(a(f_n))}$	Membership matrix for entities of type $a(f_n)$
$\mathbf{p}(o)$	Clusters' multipartite network for output o
$\vec{K}$	Every possible clusters permutation $\{(k_1, ..., k_N)\}_{k_1=1 \text{ to } K_1;...;k_N=1 \text{ to } K_N}$
$\vec{k}$	One permutation of clusters indices $(k_1, ..., k_N) \in \vec{K}$
$c_v(x)$	Count of element x in vector $\vec{v}$
C_{f_n}	Total count of f_n in R°

value among all the possible permutations $\vec{K} = \{K_1, ..., K_N\}_{K_1=1,...,K_{a(f_1)};...;K_N=1,...,K_{a(f_N)}}$. As we want SIMSBM to infer a distribution over possible outputs in a given context, the edges of the multiparted graph are related by the following constraint:

$$\sum_o p_{\vec{k}}(o) = 1 \quad \forall \vec{k} \in \vec{K} \tag{II.2}$$

Finally, the probability of an output o given a context of input entities $\vec{f}$ can be written as:

$$P(o|\vec{f}) = \sum_{\vec{k} \in \vec{K}} p_{\vec{k}}(o) \prod_{n \in N} \theta^{(a(f_n))}_{f_n,k_n} \tag{II.3}$$

From Eq. II.3, we can define the log-likelihood of the model as:

$$\ell := \log P(R^\circ|\theta, p) = \sum_{(\vec{f},o) \in R^\circ} \log \left(\sum_{\vec{k} \in \vec{K}} p_{\vec{k}}(o) \prod_{n \in N} \theta^{(a(f_n))}_{f_n,k_n} \right) \tag{II.4}$$

II.2.2.b Inference

In this section, we derive an Expectation-Maximization algorithm for inferring the model's parameters $\mathbf{p}, \theta$. Such an algorithm guarantees the convergence towards a local maximum of the likelihood function (Neal and Hinton, 1998).

E-step

The derivation presented in this paragraph follows a well-known general derivation of the EM algorithm, which can be found in (Bishop, 2006) for instance.

We recall that one entry of the dataset R° takes the form of a tuple $(\vec{f}, o)$, where $\vec{f}$ is the features input vector in which an output o has been observed. Each feature in $\vec{f}$ is associated with a cluster. We note $\vec{k} \in \vec{K}$ the set of latent variables accounting for each cluster allocation of a given feature vector among the space of $\vec{K}$ possible allocation combinations. The probability of an output given a combination of clusters is given by $p_{\vec{k}}(o)$

The total log-likelihood (Eq. II.4) is expressed as the sum of each observation's individual log-likelihood: $\log P(R^\circ|\theta,p) = \sum_{(\vec{f},o)\in R^\circ} \log P(\vec{f},o|\theta,p)$. Without loss of generality, we focus here on a single observation for clarity.

We assume a distribution $Q(\vec{k})$ on the latent variables associated with one observation in the dataset R°; this distribution is yet to be defined. Because $\vec{k}$ takes values in $\vec{K}$, we have $\sum_{\vec{k}\in\vec{K}} Q(\vec{k}) = 1$. Given this normalization condition, we can decompose Eq.II.4 for any distribution $Q(\vec{k})$ as:

$$\begin{aligned}
\log P(\vec{f},o|\theta,p) &= \underbrace{\log P(\vec{f},o,\vec{k}|\theta,p) - \log P(\vec{k}|\vec{f},o,\theta,p)}_{\text{This difference does not depend on } \vec{k}} \\
&= \Big(\log P(\vec{f},o,\vec{k}|\theta,p) - \log P(\vec{k}|\vec{f},o,\theta,p)\Big) \underbrace{\sum_{\vec{k}\in\vec{K}} Q(\vec{k})}_{=1} \\
&= \sum_{\vec{k}\in\vec{K}} Q(\vec{k}) \log P(\vec{f},o,\vec{k}|\theta,p) - \sum_{\vec{k}\in\vec{K}} Q(\vec{k}) \log P(\vec{k}|\vec{f},o,\theta,p) \\
&= \sum_{\vec{k}\in\vec{K}} Q(\vec{k}) \log \frac{P(\vec{f},o,\vec{k}|\theta,p)}{Q(\vec{k})} - \sum_{\vec{k}\in\vec{K}} Q(\vec{k}) \log \frac{P(\vec{k}|\vec{f},o,\theta,p)}{Q(\vec{k})}
\end{aligned} \tag{II.5}$$

In this equation, the first term of the last line is proportional to the expectation of the complete log-likelihood $P(\vec{f},o,\vec{k}|\theta,p)$ with respect to the latent variables $\vec{k}$ – minus the entropy of the distribution over $\vec{k}$. This explains the name "Expectation step".

We note that the second term in the last line of Eq.II.5 is the Kullback-Leibler (KL) divergence between $P(\vec{k}|\cdot)$ and $Q(\vec{k})$, noted $KL(P||Q)$. The KL divergence obeys $KL(P||Q) \geq 0$, and is null iif P equals Q. Therefore, the expectation of the complete log-likelihood is interpreted as a lower bound on the log-likelihood $\log P(\vec{f},o|\theta,p)$.

The goal of the E-step is to find the expression of $Q(\vec{k})$ that maximizes the expectation of the complete log-likelihood according to the latent variables $\vec{k}$, which is the same as maximizing the lower bound of the log-likelihood $\log P(\vec{f},o|\theta,p)$. Given that the log-likelihood does not depend on $Q(\vec{k})$ and that $KL(P||Q) \geq 0$, it is maximized when $KL(P||Q) = 0$, which occurs for $Q(\vec{k}) = P(\vec{k}|\vec{f},o,\theta,p)$. In this case, the expectation of the complete log-likelihood equals the log-likelihood itself. Its expression reaches a global maximum with respect to the latent variables $\vec{k}$ for fixed parameters θ and p.

Given the definition of the SIMSBM, the derivation of $P(\vec{k}|\vec{f},o,\theta,p)$ is straightforward (see Eq.II.4). The probability of one combination of clusters $\vec{k}$ among $\vec{K}$ possible combinations given an input features vector $\vec{f}$ and an output o is proportional to $p_{\vec{k}}(o)\prod_{n\in N}\theta^{(a(f_n))}_{f_n,k_n}$. Therefore, we can write the now-known distribution $Q(\vec{k})$, noted $\omega_{\vec{f},o}(\vec{k})$, as:

$$\begin{aligned}
\omega_{\vec{f},o}(\vec{k}) := Q(\vec{k}) &= P(\vec{k}|\vec{f},o,\theta,p) \\
&= \frac{p_{\vec{k}}(o)\prod_{n\in N}\theta^{(a(f_n))}_{f_n,k_n}}{\sum_{\vec{k'}\in\vec{K}} p_{\vec{k'}}(o)\prod_{n\in N}\theta^{(a(f_n))}_{f_n,k'_n}}
\end{aligned} \tag{II.6}$$

By substituting $Q(\vec{k})$ and $P(\vec{k}|\vec{f},o,\theta,p)$ by $\omega_{\vec{f},o}(\vec{k})$ in Eq. II.5, we get an expression for the log-likelihood which is maximized with respect to the latent variables. Explicitly:

$$\ell = \sum_{(\vec{f},o)\in R^\circ} \sum_{\vec{k}\in\vec{K}} \omega_{\vec{f},o}(\vec{k}) \cdot \log\left(\frac{p_{\vec{k}}(o)\prod_{n\in N}\theta_{f_n,k_n}^{(a(f_n))}}{\omega_{\vec{f},o}(\vec{k})}\right) \tag{II.7}$$

The maximization step follows by maximizing Eq.II.7 with respect to the parameters θ and p, holding $\omega(\cdot)$ constant.

M-step

We take back Eq. II.4 and add Lagrangian multipliers ϕ to account for the constraints on θ. We maximize of the resulting constrained likelihood ℓ_c according to each latent variable:

$$\begin{aligned}
\frac{\partial \ell_c}{\partial \theta_{mn}^{(a(m))}} &= \frac{\partial}{\partial \theta_{mn}^{(a(m))}}\left[\ell - \sum_i \phi_i^{(a(i))}\left(\sum_k \theta_{ik}^{(n)} - 1\right)\right] \\
\Leftrightarrow\quad \phi_m^{(a(m))} &= \sum_{(\vec{f},o)\in\partial m}\sum_{\vec{k}\in\vec{K}} \frac{c_{\vec{k_{i_m}}}(n)\omega_{\vec{f},o}(\vec{k})}{\theta_{mn}^{(a(m))}} \\
\Leftrightarrow\quad \theta_{mn}^{(a(m))}\phi_m^{(a(m))} &= \sum_{(\vec{f},o)\in\partial m}\sum_{\vec{k}\in\vec{K}} c_{\vec{k_{i_m}}}(n)\omega_{\vec{f},o}(\vec{k})
\end{aligned} \tag{II.8}$$

The term $c_{\vec{k_{i_m}}}(n)$ arises because of the non-linearity induced by the interaction between input entities of the same type. Let i_m be the indices where entity m appears in the input vector $\vec{f}$. The corresponding entries of the permutation vector $\vec{k}$ are noted $\vec{k_{i_m}}$. Then, $c_{\vec{k_{i_m}}}(n) = |\{1|\vec{k}_i = n\}_{i\in i_m}|$ is the count of n in $\vec{k}_{i_m}$. When n appears $c_{\vec{k_{i_m}}}(n)$ times in a permutation comprising $\vec{k_{i_m}}$, so will a term $\log\theta_{nm}^{c_{\vec{k_{i_m}}}(n)}$, whose derivative is $\frac{c_{\vec{k_{i_m}}}(n)}{\theta_{nm}}$, hence this term arising. Note that $c_{\vec{k_{i_m}}}(n) = 0$ nullifies the contribution of permutations $\vec{k}$ where n does not appear in $\vec{k_{i_m}}$. Therefore, we can restrict the sum over $\vec{k}$ in Eq. II.8 to the set $\partial n = \{\vec{k}|\vec{k}\in\vec{K}, n\in\vec{k}_{i_m}\}$. We also defined the set $\partial m = \{(\vec{f},o)|(\vec{f},o)\in R^\circ, m\in\vec{f}_{i_m}\}$.

Using Eq. II.1 and Eq. II.8, we compute $\phi_m^{(a(m))}$:

$$\begin{aligned}
\sum_n^{K_{a(m)}} \phi_m^{(a(m))}\theta_{mn}^{(a(m))} &= \sum_{(\vec{f},o)\in\partial m}\sum_n^{K_{a(m)}}\sum_{\vec{k}\in\vec{K}} c_{\vec{k_{i_m}}}(n)\omega_{\vec{f},o}(\vec{k}) \\
= \phi_m^{(a(m))} &= \sum_{(\vec{f},o)\in\partial m} \underbrace{\sum_{\vec{k}\in\vec{K}}\omega_{\vec{f},o}(\vec{k})}_{=1\text{ (Eq. II.6)}} \underbrace{\sum_n^{K_{a(n)}} c_{\vec{k_{i_m}}}(n)}_{=c_{\vec{f}}(m)} \\
&= \sum_{(\vec{f},o)\in\partial m} c_{\vec{f}}(m) = C_m
\end{aligned} \tag{II.9}$$

When summing over n, $c_{\vec{k_{i_m}}}(n)$ successively counts the number of times each n appears in $\vec{k_{i_m}}$, which equals the length of $\vec{k_{i_m}}$. Therefore $\sum_n c_{\vec{k_{i_m}}}(n) = |\vec{k_{i_m}}| = c_{\vec{f}}(m)$ is the number of times input entity m appears in the entry $(\vec{f},o)$ considered, which

does not depend on $\vec{k}$. C_m is the total count of m in the dataset. Note that this differs from the derivation proposed in (Tarrés-Deulofeu et al., 2019), where nonlinear terms are not accounted for.

The derivation of the maximization equation for $\mathbf{p}$ is very similar. Let the set $\partial s = \{(\vec{f}, o) | (\vec{f}, o) \in R^\circ, o = s\}$. We solve:

$$\begin{aligned}
&\frac{\partial \ell_c}{\partial p_{\vec{r}}(s)} = \frac{\partial}{\partial p_{\vec{r}}(s)} \left[\ell - \sum_{\vec{k}} \psi_{\vec{k}} \left(\sum_o p_{\vec{k},o} - 1 \right) \right] = 0 \\
&\Leftrightarrow \psi_{\vec{r}} = \sum_{(\vec{f},o) \in \partial s} \frac{\omega_{\vec{f},o}(\vec{r})}{p_{\vec{r}}(s)} \\
&\Leftrightarrow \sum_n \psi_{\vec{r}} p_{\vec{r}}(s) = \psi_{\vec{r}} = \sum_{(\vec{f},o) \in R^\circ} \omega_{\vec{f},o}(\vec{r})
\end{aligned} \tag{II.10}$$

Finally, combining Eq. II.8 with Eq. II.9, and the two last lines of Eq. II.10, the maximization equations are:

$$\begin{cases}
\theta_{mn}^{(a(m))} = \frac{\sum_{(\vec{f},o) \in \partial m} \sum_{\vec{k} \in \partial n} c_{k_{im}^{\rightarrow}}(n) \omega_{\vec{f},o}(\vec{k})}{C_m} \\
p_{\vec{r}}(s) = \frac{\sum_{(\vec{f},o) \in \partial s} \omega_{\vec{f},o}(\vec{r})}{\sum_{(\vec{f},o) \in R^\circ} \omega_{\vec{f},o}(\vec{r})}
\end{cases} \tag{II.11}$$

From Eq. II.11 we see that for a given $(K_{a(f_1)}, ..., K_{a(f_N)})$, one iteration of the EM algorithm runs in $\mathcal{O}(|R^\circ|)$.

II.2.2.c SIMSBM generalizes several state-of-the-art models

The formulation of the SIMSBM allows a great flexibility on modelling choices. Now, we develop how our formulation allows to recover several state-of-the-art models. Throughout this section, we denote input entities of distinct types by different letters (e.g., f_1 is not of the same type as g_1), and the model's output as o. The set of corresponding membership matrices for each type is noted as $\Theta = \{\theta^{(f)}, \theta^{(g)}, ...\}$ and one edge of the multipartite clusters-interaction tensor is noted $(p_{k(f_1),k(f_2),...}(o))$ where $k(f_i)$ is one of the possible cluster indices for an input entity of type f.

Nomenclature

We must define a nomenclature to refer to each special case of the SIMSBM –what input entity types are considered, and how many interactions for each type. We use the notation SIMSBM(number interactions type 1, number interactions type 2, ...). For instance, SIMSBM(2,3) represents a case where the SIMSBM considers two types of input entities, with the first one interacting as pairs with other entities of the same type, and the second one interacting as triples with entities of the same type. The corresponding data has a shape $(f_1, f_2, g_1, g_2, g_3, o)$ where f and g are the two considered types.

MMSBM

The historical MMSBM has been proposed in (Airoldi et al., 2008) and is at the base of most models discussed in this section. MMSBM takes pairs (f_1, o) as input data. We can recover this model with our framework by setting $\Theta = \{\theta^{(f)}\}$. The multipartite network then becomes "unipartite", which is a simple one-layer clustering

of entities. The probability of an output is defined by entities' cluster membership only. The tensor p then takes the shape $p_{f_1}(o)$. Using the SIMSBM notation, this corresponds to SIMSBM(1).

Bi-MMSBM

The Bi-MMSBM has first been proposed in (Godoy-Lorite et al., 2016) and has since been applied on several occasions (Tarrés-Deulofeu et al., 2019). In this modelling, data is made of triplets (f_1, g_1, o). Each entity is associated with a node on a side of a bipartite graph (f_i's on one side, g_i's on the other) and edges represent the probability of an output o. We recover the Bi-MMSBM with our model by setting $\Theta = \{\theta^{(f)}, \theta^{(g)}\}$ and the bipartite clusters network tensor $(p_{k(f_1),k(g_1)}(o))$. This corresponds to SIMSBM(1,1).

T-MBM

The T-MBM is a model proposed in (Tarrés-Deulofeu et al., 2019) that goes a step further than (Godoy-Lorite et al., 2016) by adding a layer to the bipartite network used to model quadruplet data (f_1, f_2, g_1, o). This model considers interactions between entities of the same type (as in Section II.2.3, see below) by clustering f_1 and f_2 using the same membership matrix, but does not account for nonlinear terms. We recover the T-MBM modelling by setting $\Theta = \{\theta^{(f)}, \theta^{(g)}\}$ and $(p_{k(f_1),k(f_2),k(g_1)}(o))$. Our formulation allows to go further by adding an arbitrary number of layers to the multipartite networks proposed in (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019). This corresponds to SIMSBM(2,1).

IMMSBM

The IMMSBM we propose in Section II.2.3 models interactions between entities of the same type to predict an output. The data takes the form (f_1, f_2, o). Each input entity is still associated with one node on either side of a bipartite graph, except that here the membership matrix is shared between the two layers. The links between each pair of clusters represent the probability of an output o. We recover the IMMSBM with our model by setting $\Theta = \{\theta^{(f)}\}$ and the bipartite clusters network tensor $(p_{k(f_1),k(f_2)}(o))$. Importantly, our formulation allows to consider interactions between n input entities instead of simply pair interactions. This corresponds to SIMSBM(2).

Indirect generalizations

We did not detail the generalization of other families of block models because our algorithm does not readily fit these cases. However, it is worth mentioning that MMSBM has been developed as an alternative to Single Membership SBM (Yuchung and George, 1987) that allows more flexibility (Airoldi et al., 2008). Our model reduces to most existing SBM by modifying the definition of the entries of $\theta^{(n)}$. In the Single Membership SBM, Eq. II.1 reads $\theta^{(n)}_{f_n,k} = \delta_{k,x}$ where x is one of the K_n possible values for k and δ is the Kronecker's delta. This means the membership vector of each input entity equals 1 for one cluster only, and 0 anywhere else. Therefore, the optimization process is not the same as we described. In practice, optimization is done with greedy algorithms such as the simulated annealing (Cobo-López, Godoy-Lorite, and Duch, 2018; Poux-Médard et al., 2021).

TABLE II.2: Datasets considered. The number of discrete values each input or output entity type can take is given between parentheses. We also provide the number of clusters for each entity type used in the experiments.

	Type of the input entities	# interactions	Type outputs	$\lvert R^\circ \rvert$	# clusters
MrBanks 1	{Player (280), Situation (7), Gender (2), Age (6)}	Situations: 3	User guess (2)	16k	{5,5,3,3}
MrBanks 2	{Player (280), Situation (7)}	Situations: 3	User guess (2)	16k	{5,5}
Spotify	{Artists (143)}	Songs: 3	Artist (740)	50k	{20}
PubMed	{Symptoms (13)}	Symptoms: 3	Disease (280)	2M	{20}
Imdb 1	{User (2502), Casting (809)}	Casting: 2	Rating (10)	1M	{10,8}
Imdb 2	{User (2502), Director (255), Casting (809)}	None	Rating (10)	700k	{10,10,10}

It has also been shown in (Godoy-Lorite et al., 2016)-Eq.7 that the Bi-MMSBM model generalizes matrix factorization. Therefore, it follows that SIMSBM also generalizes it. The underlying idea is to remove the multipartite network tensor p and define clusters that are shared by both sides of the bipartite network. This way, clusters do not interact with each other because they are not embedded into a multipartite network; input entities on one side of the bipartite network belonging to one cluster are solely linked to entities on the other side belonging to this same cluster.

II.2.2.d Experiments

Range of application

As shown in the previous section, our formulation generalizes several existing models from the state-of-the-art. Therefore, it is readily applicable to any of the datasets considered in these works. This includes recommendation datasets (movies (Godoy-Lorite et al., 2016), songs), medical datasets (symptoms-disease networks, drug interaction networks (Guimerà and Sales-Pardo, 2013; Tarrés-Deulofeu et al., 2019)) and social behaviour datasets (social dilemmas (Cobo-López, Godoy-Lorite, and Duch, 2018; Poux-Médard et al., 2021), e-mail networks (Godoy-Lorite, Guimerà, and Sales-Pardo, 2016; Tarrés-Deulofeu et al., 2019)). In general, it applies to datasets where there is a given number of input entities leading to a set of possible outputs. In this section, we propose to illustrate an application of our model on 6 different datasets.

Datasets

The **MrBanks datasets** has been gathered from a social experiment led in Barcelona in 2013, detailed in (Gutiérres-Roig et al., 2016). The experiment takes the form of a game where a player must guess whether a stock market curve will go up or down at the next time step. To do so, she can access various pieces of information, from which we selected the most relevant subset based on the description in (Gutiérres-Roig et al., 2016; Poux-Médard et al., 2021): direction of the market on the previous day (up/down), whether she guessed right (yes/no), and an expert's advice who is correct 60% of the time (up/down/not consulted). Those are the 7 interacting pieces of information that define a situation. If the model considers pair interactions for instance, a situation can be defined as "market went down and user guessed wrong", or "market went up and expert advised up". A triplet interaction allows one to get the full picture according to the selected pieces of information. In addition, we have access to the player's age and gender. The goal is to predict whether the user will guess up or down given the available information.

For the **Spotify dataset** we collected user-made playlists on Spotify using the Spotipy python API. Our goal is to predict which next artist the user will add to the

TABLE II.3: Results for every dataset presented. The letters in superscript represent the model SIMSBM generalizes in this particular configuration: MMSBM (Airoldi et al., 2008)=[a]; Bi-MMSBM (Godoy-Lorite et al., 2016)=[b]; IMMSBM (Section II.2.3)=[c]; T-MBM (Tarrés-Deulofeu et al., 2019)=[d]. The standard error on the last digits over all 100 runs is indicated between parenthesis – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$. The models presented in this chapter are underlined. Overall, we see that our formulation allows to improve results on every dataset.

			F1	P@1	AUCROC	AUCPR	RAP	NCE
MrBanks 1	**Ply, Sit (3), Gen, Age**	SIMSBM(1,1,1,1)	0.7124(2)	0.6549(3)	0.7071(2)	0.7141(3)	0.8274(1)	0.1726(1)
		SIMSBM(1,2,1,1)	0.7107(2)	0.6696(5)	0.7120(4)	0.7158(5)	0.8348(3)	0.1652(3)
		SIMSBM(1,3,1,1)	**0.7348(2)**	**0.7172(5)**	**0.7610(4)**	**0.7646(4)**	**0.8586(3)**	**0.1414(3)**
		TF	0.6795	0.6037	0.4702	0.4967	0.8019	0.1981
		NMF	0.7178	0.6976	0.7232	0.7182	0.8409	0.1591
		KNN	0.7023	0.6648	0.6859	0.6623	0.8324	0.1676
		NB	0.6867	0.6382	0.6323	0.6250	0.8191	0.1809
		BL	0.6795	0.6037	0.5000	0.5215	0.8019	0.1981
MrBanks 2	**Ply, Sit (3)**	SIMSBM(1,1)[b]	0.7032(1)	0.6700(3)	0.7049(2)	0.7018(2)	0.8350(2)	0.1650(2)
		SIMSBM(1,2)[d]	0.7032(2)	0.6679(5)	0.7028(4)	0.7010(4)	0.8340(3)	0.1660(3)
		SIMSBM(1,3)	**0.7290(3)**	**0.7067(6)**	**0.7547(5)**	**0.7530(6)**	**0.8533(3)**	**0.1467(3)**
		TF	0.6775	0.5953	0.5054	0.5259	0.7976	0.2024
		NMF	0.7137	0.6908	0.7246	0.7128	0.8397	0.1603
		KNN	0.7100	0.6699	0.7126	0.6856	0.8349	0.1651
		NB	0.6802	0.6512	0.6329	0.6225	0.8256	0.1744
		BL	0.6775	0.5953	0.5000	0.5181	0.7976	0.2024
Spotify	**Artists (3)**	SIMSBM(1)[a]	0.1741(4)	0.2155(7)	**0.7908(6)**	0.1603(3)	0.3827(4)	0.0786(3)
		SIMSBM(2)[c]	0.3156(5)	**0.3348(4)**	0.7661(5)	0.2545(3)	**0.4528(3)**	0.0938(6)
		SIMSBM(3)	**0.3243(4)**	0.3209(3)	0.7384(6)	**0.2613(3)**	0.4366(3)	0.1079(7)
		TF	0.026 20	0.004 200	0.4805	0.015 90	0.096 20	0.1550
		NMF	0.037 10	0.065 80	0.5650	0.040 30	0.1762	0.2557
		KNN	0.3201	0.3009	0.7079	0.2400	0.3941	0.5212
		NB	0.046 30	0.084 60	0.7005	0.057 60	0.2264	**0.076 30**
		BL	0.026 20	0.053 20	0.5000	0.013 50	0.1879	0.096 90
PubMed	**Symptoms (3)**	SIMSBM(1)[a]	0.2915(2)	0.5576(4)	0.7475(1)	0.2658(1)	0.4641(1)	0.2033(1)
		SIMSBM(2)[c]	0.3127(1)	0.5704(1)	0.7613(1)	0.2840(1)	0.4838(1)	0.1991(1)
		SIMSBM(3)	**0.3219(1)**	**0.5790(1)**	**0.7666(1)**	**0.2895(1)**	**0.4937(1)**	**0.1983(1)**
		TF	0.1607	0.1003	0.5605	0.1777	0.1370	0.5118
		NMF	0.1606	0.029 30	0.5368	0.2158	0.2321	0.2959
		KNN	0.2414	0.3251	0.6154	0.2324	0.2891	0.7730
		NB	0.2600	0.1618	0.7054	0.2389	0.2036	0.3058
		BL	0.1607	0.1003	0.5000	0.1026	0.2464	0.2834
Imdb 1	**Usr, Cast (2)**	SIMSBM(1,1)[b]	**0.3212(1)**	**0.2434(1)**	**0.6265(1)**	**0.2502(1)**	0.4360(1)	0.3504(1)
		SIMSBM(1,2)[d]	0.2546(1)	0.1006(38)	0.4998(3)	0.1509(1)	0.2928(42)	0.4527(51)
		TF	0.2546	0.2300	0.4960	0.1485	0.4568	0.2702
		NMF	0.1329	0.059 30	0.5007	0.1531	0.1549	0.8087
		KNN	0.2578	0.1899	0.5489	0.1679	0.3290	0.5328
		NB	0.2555	0.2351	0.5308	0.1607	**0.4619**	**0.2596**
		BL	0.2546	0.2300	0.5000	0.1508	0.4605	0.2586
Imdb 2	**Usr, Dir, Cast**	SIMSBM(1,1,1)	**0.3896(1)**	**0.3437(2)**	**0.7593(1)**	**0.3293(2)**	**0.5705(1)**	**0.1654(1)**
		TF	0.2547	0.2238	0.5039	0.1513	0.4549	0.2636
		NMF	0.1127	0.048 30	0.5005	0.1529	0.1406	0.8319
		KNN	0.2596	0.1890	0.5501	0.1681	0.3268	0.5248
		NB	0.2558	0.2373	0.5362	0.1617	0.4632	0.2571
		BL	0.2547	0.2286	0.5000	0.1507	0.4598	0.2574

playlist, given the previous artists he already added. We consider the last 4 artists added by the user and their interaction to guess the next one. Note that it often happens for an artist to be added several times in a row.

The **PubMed dataset** is made of medical reports we gathered using the PubMed API. We use provided keywords to isolate symptoms and diseases in the text, as in (Zhou et al., 2014). Our goal is to guess which diseases are discussed in the report given the symptoms that are listed in the document. Our guess is that a combination of symptoms helps narrow the set of possible diagnoses.

Finally, the **Imdb datasets** are provided and discussed in (Harper and Konstan, 2015). The original dataset comes with information about movies such as the lead actors starring in them and the movie's director. It also provides a list of users' ratings on movies. We aim to predict which rating a movie will get according to several combinations of parameters and their interactions: who directed the movie, who played in it, who gave the rating, etc.

The datasets we consider here are summarized in Table II.2. 90% of each dataset's documents are used as a training set, and the other 10% are used as an evaluation set. Each iteration of the SIMSBM is run 100 times on every dataset. The EM algorithm stops once the relative variation of the likelihood falls below 10^{-4} for 30 iterations in a row. We present the average results over all the runs. The number of clusters has been chosen based on the existing literature on similar datasets (Imdb (Godoy-Lorite et al., 2016), MrBanks (Poux-Médard et al., 2021)) and heuristic methods (Section II.2.3.c) for demonstration purposes; dedicated work would be needed to infer their optimal number for every dataset.

Finally, when SIMSBM is evaluated on a dataset containing more interactions than it is designed to consider, the model is trained on the lower-order corresponding dataset. For instance, imagine a dataset considering one type interacting three times. This dataset is made of one observation only $(1, 2, 3, o)$. A SIMSBM iteration that considers pair interactions will then be trained on triplets $(1, 2, o)$, $(1, 3, o)$ and $(2, 3, o)$, and evaluation will be performed accordingly.

Baselines and evaluation

Evaluation is done by considering test set entries $\left(\vec{f}, o_{true}\right)$. We ask the SIMSBM to provide a probability for each possible output $o \in O$ given the unobserved input context $\vec{f}$. We then compare the resulting probability distribution to o_{true}. We evaluate our results according to the maximum **F1** score, the precision at 1 (**P@1**), the area under the ROC curve (**AUCROC**), the area under the Precision-Recall curve (**AUCPR**, or Average Precision). Since the problem is about multi-label classification, we consider the weighted version of these metrics –metrics are computed individually for each class, and averaged with each class's weight being equal to the number of true instances in the class. The presented results are averaged over all 100 runs. We also consider the rank average precision (**RAP**) and the normalized covering error (**NCE**, only here lower is better).[1] We compare to several standard baselines:

- BL: the most naive baseline, where each output is predicted according to its frequency in the training set, without any context.
- NB[1]: the Naive Bayes baseline assumes conditional independence between the input entities and updates the posterior probability according to Bayes law.

[1] We used the sklearn Python library implementation.

- KNN[1]: K-nearest-neighbours. The output probabilities for a given entities array are defined according to a majority vote among the most similar entities arrays.
- NMF[1] and TF: Tensor Factorization baselines. For TF, we use the implementation provided by authors of (Karatzoglou et al., 2010). As discussed in the introduction, to make these methods fit our problem, we have to define a continuous quantity to train the model. Instead of requiring an additional model to map possible outputs into a continuous space, we train the model on the frequency of outputs in a given context. Since NMF can only consider one entity as an input, we consider each different context as an independent entity. Outputs are added as an additional dimension to the data matrix instead of being a proper objective –their frequency is now the objective. The TF baseline is run for the same number of clusters as for the SIMSBM.
- MMSBM (Airoldi et al., 2008), Bi-MMSBM (Godoy-Lorite et al., 2016), IMMSBM (Section II.2.3), T-MBM (Tarrés-Deulofeu et al., 2019): as discussed before, each of these models are particular cases of SIMSBM. For presentation purpose, for each model, we keep the SIMSBM notation and indicate in superscript which of these it reduces to in this context. MMSBM=^{a}; Bi-MMSBM=^{b}; IMMSBM=^{c}; T-MBM=^{d}.

II.2.2.e Discussion

We present our main results in Table II.3. In this table, we see that our formulation systematically outperforms the proposed baselines, as well as the ones it generalizes. In most cases, the possibility to add a layer or to consider higher-order interactions improves the performance over the existing baselines (MMSBM, Bi-MMSBM, IMMSBM and T-MBM). About the Spotify dataset, as stated before, artists are often added to a playlist in a row, leading to the probability of the next artist being the same as the one immediately before her to be higher. In this context, adding interaction terms adds noise to the modelling. This explains why the triple interactions version of SIMSBM does not perform better than its pair-interactions (Godoy-Lorite et al., 2016) or no-interaction (Airoldi et al., 2008) iterations.

We also ran SIMSBM on the datasets considered in (Godoy-Lorite et al., 2016) and (Poux-Médard et al., 2021) in the corresponding model's specifications and obtained comparable results as theirs (provided in Appendix, Section I.1). In the case of (Poux-Médard et al., 2021), the authors propose to describe a given situation in form of a unique key, where each key is independent of the others. Interestingly, the formulation with triple interactions improves the results on the same dataset the authors provided. This is because the constituents of a situation are not independent anymore but instead behave as elementary interacting pieces of context, which provides a more accurate description of reality: a situation is not considered as a whole anymore, but instead as the combination of several pieces of information.

II.2.2.f Conclusion

In this section, we developed a global framework, SIMSBM, that generalizes several models from the literature as particular cases, such as MMSBM, Bi-MMSBM, IMMSBM and T-MBM. Our derivation accounts for the nonlinearity induced by the interactions, which has not been taken into account in state-of-the-art works.

This results in a highly flexible model that can be applied to a broad range of problems, as shown through systematic evaluation of the proposed formulation on several real-world datasets. In particular, we cited throughout the text several experimental studies conducted in medicine, social behaviour and recommendation using special cases of our model; using alternative iterations of the SIMSBM framework may help further improve the description and understanding of the interacting processes at stake between an arbitrary greater number of input entities.

To summarize, we developed a framework that allows to consider MMSBMs for any number of entity types that can have arbitrarily high-order interactions. This answers the two challenges raised in Section II.2.1.b. We now propose to restrict our study of interactions in information spread to a special case of the SIMSBM –the IMMSBM.

II.2.3 IMMSBM – A study of pair interactions

This work has been published, see (Poux-Médard, Velcin, and Loudcher, 2021b).

About this section

As stated at the end of Section II.2.2, the IMMSBM introduced here is a special case of the SIMSBM, noted SIMSBM(2). As such, the motivations and derivations detailed in Section II.2.2.a are readily applicable to the case at hand.

In particular, Section II.2.3.a and Section II.2.3.b may seem redundant at first glance. However, these sections' argumentation is specifically focused on interaction modelling. Besides, we present an alternative, simpler, derivation of the EM algorithm for SIMSBM using Jensen's inequality in Section II.2.3.b.

The novel results of SIMSBM(2), or IMMSBM, applied to interactions modelling are discussed from Section II.2.3.c to Section II.2.3.e.

II.2.3.a IMMSBM – Interacting MMSBM

MMSBM to model interactions

In this section, we consider a specific iteration of the SIMSBM framework, referred to as IMMSBM (for Interacting Mixed Membership Stochastic Block Model). Studying interactions using a model built on the mixed-membership SBMs allows us to assume that each entity does not have only one membership. This is in line with what is expected for real-life situations, where entities can belong to several clusters simultaneously (a song can be rock *and* blues, a word can be used in two topics, etc.). The IMMSBM requires no prior information on the system and its numerical implementation is possible via the scalable Expectation-Maximization algorithm detailed in the previous section. The EM complexity scales linearly with the size of the dataset. Our method offers a better predictive power than non-interacting baselines.

The goal of the model is to predict the most likely result of an interaction between two entities (i and j in Fig. II.4). Typically, IMMSBM yields the probability of a tweet getting retweeted given a user's previous exposure to pairs of tweets, or the product the most likely to be bought given item pairs in a user's purchase history. Another example about assisted medical diagnosis: observing the words "fatigue" and "cough" in a medical report is more likely to imply the observation of "flu" than "anaemia", despite "anaemia" being often associated with the "fatigue" symptom. The model will group data into clusters (each user's membership is encoded in the matrix θ, see Fig. II.4) that interact symmetrically with each other (interactions tensor

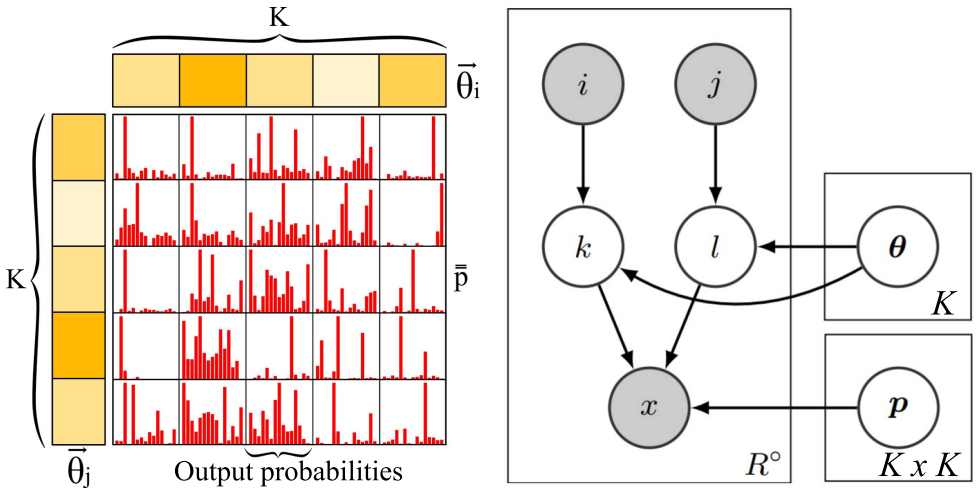

FIGURE II.4: **Illustration of the model — (Left)** Schema of IMMSBM for a single pair of entities (i, j) (which could be "fever" and "cough" for instance). Input entities are grouped into K clusters in different proportions; the proportion to which they belong to each cluster is quantified by a θ matrix (dimension $[I \times K]$ where I is the input space). The clusters then interact to generate a probability distribution over the output entities defined by the interactions tensor $\mathbf{p}$ (dimension $[K \times K \times O]$ where O is the output space). **(Right)** Alternative representation of the IMMSBM as a graphical model. To generate each output, for each observation (i, j, x) in the set R°, a cluster (k and l) is drawn for each input entity (i, j) from a distribution encoded in the matrix θ. The generated output x is drawn from a multinomial distribution conditioned by the previously drawn clusters k and l encoded in $\vec{p}$.

$\mathbf{p}$), resulting in a probability over the possible outputs to appear (histograms Fig. II.4-left). We have no prior knowledge of the content of the groups, and we need to set the number of clusters K.

Model

We refer to the interacting entities as input entities $(i, j) \in I^2$, and to an output as $x \in O$. I is the input space (the entities that interact: products, symptoms, or songs for instance) and O is the output space (the entities resulting from the interaction in an answer, diagnosed diseases for instance). We illustrate in Fig. II.4-left how the input space and output space are related according to IMMSBM: input entities are clustered, and the interaction between those clusters yields a probability distribution over the possible output. Note that I and O can be disjoint, unlike in (Myers and Leskovec, 2012). In the case of retweet prediction, I accounts for tweets in a user's history, and O for retweeted tweets –that do not necessarily appear in the user's feed. As an alternative visualization, we present the graphical generative model of the IMMSBM in Fig. II.4-right. The observed data then takes the form of triplets (i, j, x) signifying that the combined presence of input entities i and j leads to the output entity x.

We assume there are regularities in the studied datasets. Given subsets of inputs may exhibit a similar behaviour. In the medical dataset example, if symptoms such as "fever" and "pallor" often come in pairs, they are considered as similar regarding the diagnosis. They belong to the same cluster. We define the membership matrix θ associating each input entity with clusters in different proportions, such that θ_i is a $[1 \times K]$ vector with $\sum_k^K \theta_{i,t} = 1$. Note that unlike in single membership stochastic block models an entity does not have to belong to only one group (Funke and Becker,

2019). Given the possible semantic variation of entities (polysemy of words in natural languages -e.g., "like", "swallow"-, symptoms with various causes in medicine -"headache", "fever"-, etc.), an approach *via* a mixed-membership clustering is more in line with reality.

We then define the cluster interactions tensor $p_{k,l}(X_{k,l} = x)$ whose dimensions are $[K \times K \times O]$ as the probability that the interaction between clusters $k \in K$ and $l \in K$ gives rise to the output $x \in O$. By definition, $\sum_x p_{k,l}(X_{k,l} = x) = 1 \; \forall k, l$. The role of those two quantities is schematized in Fig. II.4.

We choose to consider only one membership matrix θ for all of the inputs, instead of one per input entry as in (Godoy-Lorite et al., 2016). It enforces the model to be permutation-independent with respect to the input entities. Observation (i, j, x) is equivalent to (j, i, x). This follows the idea of (Beutel et al., 2012) where it is assumed that the interaction between two viruses is symmetric, meaning the interaction influences both viruses with the same magnitude. There is no need to consider a different clustering for inputs i and j, which motivates the use of a single membership matrix θ. This differs from other recent works on interaction modelling that do not consider permutation-invariant clustering (Christakopoulou and Karypis, 2014; Steck and Liang, 2021) or the non-linearity induced by symmetric interactions (Myers and Leskovec, 2012; Tarrés-Deulofeu et al., 2019).

We now propose to define the entities interactions tensor $P_{i,j}(X_{i,j} = x)$, representing the probability that the interaction between inputs i and j implies the output x as:

$$P_{i,j}(X_{i,j} = x) = \sum_{k,l} \theta_{i,k}\theta_{j,l} p_{k,l}(X_{k,l} = x) \tag{II.12}$$

For the sake of brevity, from now on we will refer to $p_{k,l}(X_{k,l} = x)$ as $p_{k,l}(x)$. We define the likelihood of the observations given the parameters as:

$$P(R^\circ|\theta, p) = \prod_{(i,j,x)\in R^\circ} \sum_{k,l} \theta_{i,k}\theta_{j,l} p_{k,l}(x) \tag{II.13}$$

where R° denotes the set of triplets in the training set (input, input, output). Note that the remaining triplets $R \setminus R^\circ$ are used as test set.

From the definitions above, it is straightforward to show that IMMSBM corresponds to the special case SIMSBM(2), detailed in the previous section.

II.2.3.b Inference of the parameters

Expectation step

We detailed a general derivation of the E-step of EM algorithm for mixture models in Section II.2.2.a. Here, we present an alternative (and quicker) derivation of the E-step using Jensen's inequality. The two methods yield identical results. Taking the logarithm of the likelihood as defined in Eq. II.13, denoted ℓ, we have:

$$\begin{aligned}
\ell &= \sum_{(i,j,x)\in R^\circ} \ln \sum_{k,l} \theta_{i,k}\theta_{j,l} p_{k,l}(x) \\
&= \sum_{(i,j,x)\in R^\circ} \ln \sum_{k,l} \omega_{i,j,x}(k,l) \frac{\theta_{i,k}\theta_{j,l} p_{k,l}(x)}{\omega_{i,j,x}(k,l)} \\
&\geq \sum_{(i,j,x)\in R^\circ} \sum_{k,l} \omega_{i,j,x}(k,l) \ln \frac{\theta_{i,k}\theta_{j,l} p_{k,l}(x)}{\omega_{i,j,x}(k,l)}
\end{aligned} \tag{II.14}$$

We used Jensen's inequality to go from the 2nd to 3rd line. The inequality in Eq. II.14 becomes an equality for:

$$\omega_{i,j,x}(k,l) = \frac{\theta_{i,k}\theta_{j,l}p_{k,l}(x)}{\sum_{k',l'}\theta_{i,k'}\theta_{j,k'}p_{k',l'}(x)} \tag{II.15}$$

where $\omega_{i,j,x}(k,l)$ is interpreted as the probability that the observation (i,j,x) is due to i belonging to the group k and j to l, that is the expectation of the likelihood of the observation (i,j,x) with respect to the latent variables k and l. Therefore, Eq. II.15 is the formula for the expectation step of the EM algorithm. As stated earlier, an alternative derivation of the expectation step is discussed in (Bishop, 2006) and in Section II.2.2.b.1.

Maximization step

This step consists in maximizing the likelihood using the parameters of the model θ and p, independently of the latent variables. To take into account the normalization constraints, we introduce the Lagrange multipliers ϕ and ψ. Following this, the constrained log-likelihood reads:

$$\ell_c = \ell - \sum_i(\phi_i\sum_t\theta_{i,t} - 1) - \sum_{k,l}(\psi_{k,l}\sum_x p_{k,l}(x) - 1) \tag{II.16}$$

We first maximize ℓ_c with respect to each entry θ_{mn}.

$$\begin{aligned}
\frac{\partial\ell_c}{\partial\theta_{mn}} =& \frac{\partial\ell}{\partial\theta_{mn}} - \phi_m = 0 \\
=& \sum_{\partial m}\left(\sum_l\frac{\omega_{m,j,x}(n,l)}{\theta_{mn}} + \sum_k\frac{\omega_{j,m,x}(k,n)}{\theta_{mn}}\right) - \phi_m \\
=& \frac{1}{\theta_{mn}}\sum_{\partial m}\left(\sum_t\omega_{m,j,x}(n,t) + \omega_{j,m,x}(t,n)\right) - \phi_m \\
\Leftrightarrow\ \theta_{mn} =& \frac{1}{\phi_m}\sum_{\partial m}\sum_t\omega_{m,j,x}(n,t) + \omega_{j,m,x}(t,n)
\end{aligned} \tag{II.17}$$

where $\partial m = \{(j,x)|(m,j,x) \in R^\circ\}$ stands for the set of observations in which the entry m appears. Note that summing over this set in line 2 of Eq. II.17 implies that the relation between two inputs entities is symmetric –if $(i,j,x) \in R^\circ$ implies $(j,i,x) \in R^\circ$. Summing the last line of Eq. II.17 over $n \in K$ then multiplying it by ϕ_m, we get an expression for ϕ_m:

$$\phi_m\sum_n\theta_{mn} = \phi_m = \sum_{\partial m}\sum_{t,n}\omega_{m,j,x}(n,t) + \omega_{i,m,x}(t,n) = \sum_{\partial m}2 = 2\cdot n_m \tag{II.18}$$

Where n_m is the total number of times that m appears as an input in R°. Finally, plugging back this result in Eq. II.17, we get:

$$\theta_{mn} = \frac{\sum_{\partial m}\left(\sum_t\omega_{m,j,x}(n,t) + \omega_{i,m,x}(t,n)\right)}{2.n_m} \tag{II.19}$$

Following the same line of reasoning for p, we get:

$$p_{r,s}(l) = \frac{\sum_{\partial l} \omega_{i,j,l}(r,s)}{\sum_{(i,j,l')\in R^\circ} \omega_{i,j,l'}(r,s)} \tag{II.20}$$

The set of Eq. II.19 and Eq. II.20 constitutes the maximization step of the EM algorithm. They hold only if the input entities interactions are symmetric (e.g., when $\{(j,x)|(m,j,x) \in R^\circ\} = \{(i,x)|(i,m,x) \in R^\circ\}$), which is what we aimed to do. It is worth noting that the proposed algorithm offers linear complexity with the size of the dataset $\mathcal{O}(|R^\circ|)$ provided the number of clusters is constant, while guaranteeing convergence to a local maximum.

Instantaneous derivation as a special case of SIMSBM

The EM equations we just derived could be directly retrieved, given that IMMSBM is corresponds to the special case SIMSBM(2). We define the membership matrices set $\Theta = \{\theta^{(f)}\}$, the interaction tensor $p_{k(f_1),k(f_2)}(o)$ and consider data shaped as (f_1, f_2, o). Plugging these into Eq. II.6 and Eq. II.11 yields identical expressions as Eq. II.15, Eq. II.19 and Eq. II.20.

II.2.3.c Experiments

Datasets and evaluation protocol

We assess the performance of IMMSBM on 4 different datasets. The **PubMed** dataset is built with 15,809,271 medical reports collected from the PubMed database as a good approximation for human-disease network (Zhou et al., 2014). This dataset is not explicitly about recommender systems but provides an intuitive way to understand how our recommendation approach works by suggesting likely diseases given a collection of symptoms. The **Twitter** dataset with 139,098 retweets gathered in October 2010 associated with the 3 last tweets in the feed preceding each retweet (Hodas and Lerman, 2014). The task is to infer which tweets a user is the most likely to retweet. A possible application would be a personalized recommendation of such tweets in the "Trends for you" Twitter section. The **Reddit** dataset with the entirety of posts in the subreddit r/news in May 2019 (225,485 message-answer relationships in total). We aim at predicting the content of the answer given the incoming message. A possible application would be similar to what Gmail does when suggesting automated answers to an email given keywords present in the message. Finally, **Spotify** dataset is built with 2,000 music playlists associated with keywords "English" and "rock" of random Spotify users. We predict the next song a user will add to a playlist given this user's history. A RS application would suggest a ranked list of songs to the user is likely to add to a playlist. Each dataset is formed by associating every pair of inputs in a *message* (i.e., a list of symptoms, a user's feed, a Reddit post, and a playlist's last artists) with an *answer* (i.e., a disease, a retweet, a Reddit answer, an artist added to a playlist). This is illustrated in Fig. II.5. The building process of datasets is further detailed in Appendix, Section I.2, together with direct links to access them for possible replication studies.

From the raw datasets, we form the test set by randomly sampling 10% of the (message→answer) data entries. The 90% entries left are used as a training set. The number of clusters is determined using the elbow method (see Fig. II.6). The number of clusters considered for each corpus: 30 for PubMed, 15 for Twitter, 30 for Reddit,

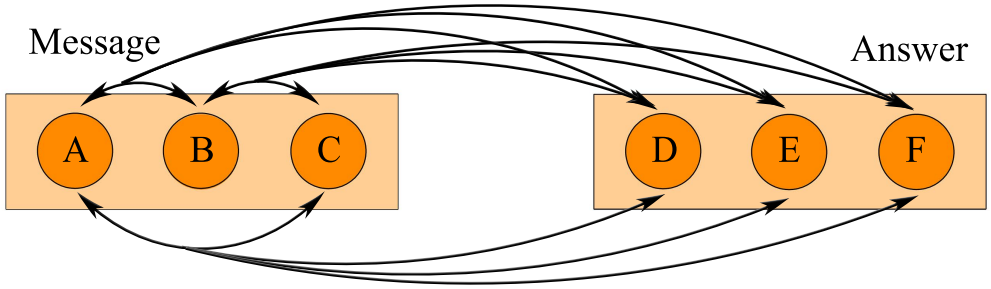

FIGURE II.5: **Illustration of dataset generation** — Each dataset is organized as a list of message-answer pairs. Each message and each answer can comprise several entities (words, URLs, etc.). We consider every possible pair of entities in the message, and link each of them to each output in the answer to create our triplets dataset (i, j, x)

and 15 for Spotify. We perform 100 independent runs, each with independent random initialization of the parameters θ and p. The EM loop stops once the relative variation of the likelihood between two iterations is less than 0.001%.

Baselines

- **Naive baseline** – The naive baseline is simply the frequency of the outputs in the test set. This naive classifier predicts the value of every output independently of the inputs.

- **MMSBM** – We use the classical MMSBM as a second baseline. In this formulation, interactions are not taken into account (Airoldi et al., 2008). Instead of considering triplets (input, input, output), we consider pairs (input, output). We then train this MMSBM baseline whose log-likelihood is defined as $\ell_{BL} = \sum_{(i,x)\in R^\circ} \ln \sum_k \theta_{ik} p_k(x)$ on the same datasets as in the main experiments. We make 100 independent runs with random initialization and discuss the results of the highest likelihood run. This baseline provides a way to quantify the importance of interactions by comparing to the case where these are taken into account. We expect the baseline model to find results that are equivalent to the diagonal probabilities of the main model (i.e., similar to $P_{i,i}(x) = \sum_{k,l} \theta_{i,k}\theta_{i,l} p_{k,l}(x)$). This is because the diagonal $P_{i,i}(x)$ is supposed to account for the apparition of x given only the presence of i. Furthermore, this model provides insight into the generalization of the assumption made in (Myers and Leskovec, 2012), stating that the probability of an output is mostly equal to the frequency of this output.

- **Perfect modelling** - Upper limit to prediction – We compare our results to an upper limit to predictions. In most situations, the dataset simply does not allow for perfect performances. Consider as an example a case where the test set contains twice the triplet (“fever”, “pallor”, “influenza”) and once the triplet (“fever”, “pallor”, “anaemia”): a model yielding a single value for the input pair (“fever”, “pallor”) cannot make a prediction better than 66%. In Appendix, Section I.3, we develop a general method to derive this upper limit to predictions.

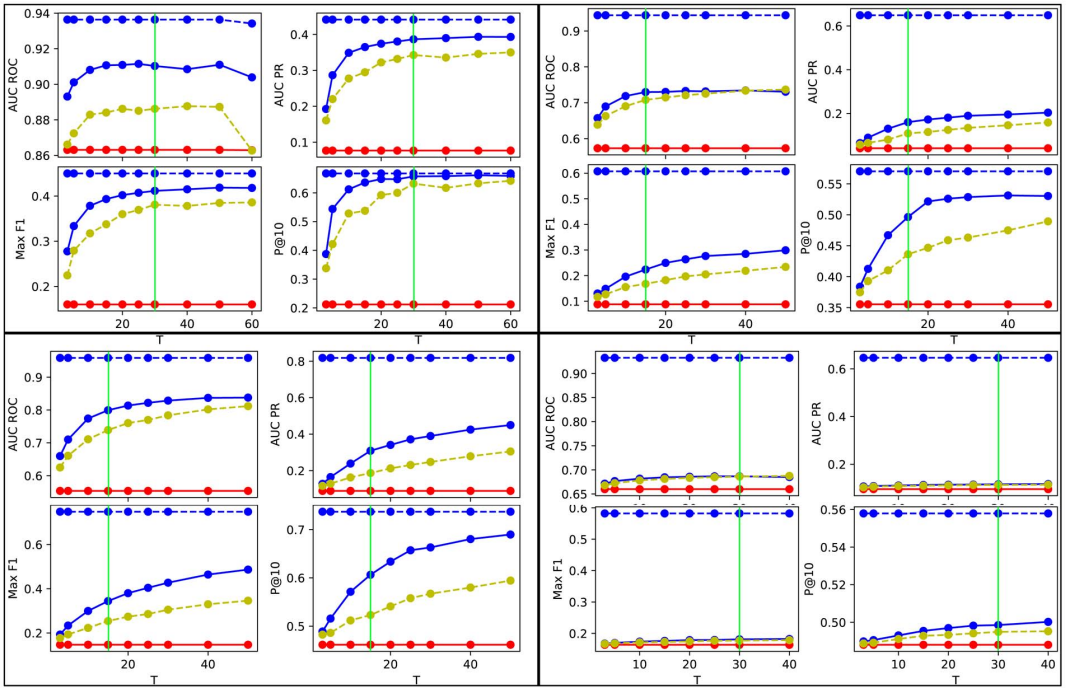

FIGURE II.6: **Choosing the number of clusters** — Performance variations on all the metrics for every dataset considered. Dashed blue line: upper limit to performances ; blue line: IMMSBM ; yellow dashed line: MMSBM ; red line: naive baseline. Top left: PubMed ; top right: Spotify ; bottom left: Twitter ; bottom right: Reddit. The vertical green line shows the selected number of clusters; it is chosen using the AIC criterion, which matches with the elbow of the various metrics considered.

Results

The metrics we use to assess the performance of our model are the **max-F1** score, the area under the receiver operating characteristic curve (**AUCROC**) curve and the precision@10 (**P@10**). For all of these quantities, the closer to 1 they are, the better the performance.

For evaluation, we adopt a guessing process as follows. For every pair of inputs, we compute the probability vector for the presence of every possible output. Then we predict all of the probabilities larger than a given threshold to be "present", and all the others to be "absent". Comparing those predictions with the observations in the test set, we get the confusion matrix for the given threshold. We then lower the threshold and repeat the process to compute the various metrics.

We recall that we do not compare our approach to (Myers and Leskovec, 2012) because its formulation does not allow us to make a prediction on exogenous outputs – when the output is not part of the input pair.

In Table II.4, we show the performances of our model compared with the baselines introduced in Section II.2.3.c.2. We see that IMMSBM outperforms the proposed baselines in most cases. As expected, taking interactions between entities into account leads to improved accuracy in the prediction of the test outputs. This correlates to the conclusions drawn in (Myers and Leskovec, 2012), stating the importance of interactions in real-world phenomenon modelling.

We notice however that accounting for interactions did not lead to a significant improvement in performance over the two baselines on the Reddit corpus. The lack of improvement can be imputed to the dataset itself. The Reddit dataset contains few

TABLE II.4: Experimental results for the four metrics considered, from each model applied to each corpus. We see that our model outperforms the proposed baselines in every dataset for almost every evaluation metric – the error bars overlap for the AUC (ROC) on the PubMed corpus. The given error corresponds to the standard deviation over the 10 runs – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$. The naive baseline and upper limit results are constant over the runs and therefore have no variance. The models presented in this chapter are underlined.

		P@10	Max-F1	AUC ROC
PubMed	Naive	0.212	0.160	0.863
	MMSBM	0.627(2)	0.393(2)	**0.911(0)**
	IMMSBM	**0.656(1)**	**0.411(1)**	**0.911(2)**
	Up.lim.	0.668	0.450	0.936
Twitter	Naive	0.462	0.147	0.554
	MMSBM	0.529(5)	0.254(5)	0.741(4)
	IMMSBM	**0.610(4)**	**0.349(6)**	**0.800(1)**
	Up.lim.	0.737	0.748	0.959
Reddit	Naive	0.488	0.164	0.660
	MMSBM	0.495(0)	0.177(0)	0.686(0)
	IMMSBM	**0.499(0)**	**0.181(0)**	**0.687(0)**
	Up.lim.	0.558	0.582	0.933
Spotify	Naive	0.355	0.088	0.573
	MMSBM	0.426(6)	0.167(3)	0.707(2)
	IMMSBM	**0.502(6)**	**0.228(5)**	**0.723(2)**
	Up.lim.	0.570	0.607	0.944

observations for every possible pair, due to the wide range of available vocabulary of natural language (Loreto et al., 2016). Likely, the model has not been trained with enough data to learn significant regularities in pair interactions. This can also be seen during the building of the test set: approximately one-half of the pairs have never been observed in the training set. As future perspectives, it might be interesting to answer this problem by considering a corpus of pre-clustered entities instead of independent named entities, hence reducing the vocabulary range and adding to the regularity of the dataset.

Our results show that taking interactions between entities into account is particularly relevant in the case of the PubMed corpus (98.2% of the maximum reachable precision@10 vs 93.8% for the non-interacting baseline). It seems reasonable to consider that a diagnosis is better determined by joint observation of given symptoms, and not only by the sum of their individual effect. The interaction aspect is especially relevant given the small number of observed symptoms (322) used to predict the possible diseases (4,442).

For the Twitter corpus, we confirm the results of (Myers and Leskovec, 2012) on the importance of interactions between URLs in their spread. Our model consequently outperforms the non-interacting baseline.

Finally, our model performs better than the baselines on the Spotify dataset. In particular, it achieves better prediction for the top-10 artists one would listen to (+7.6%). A good P@10 precision is of key interest in the application of any model to playlist building and recommender systems in general. Taking into account artists' interaction clearly added to the level of prediction details.

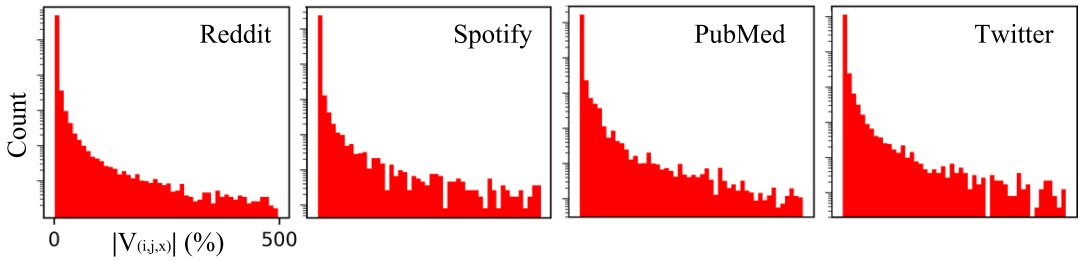

FIGURE II.7: **Relative impact of interactions** — Histogram of the relative impact of interactions on the base virality of outputs. Overall, most interactions do not lead to any notable change in the probability of an output.

II.2.3.d Discussion

Global impact of interactions

IMMSBM infers virality along with interaction terms and yields better results than state-of-the-art methods (see Table II.4), which provides solid ground for analysing the effect of information interaction. Close analysis of interactions between pieces of information has been little considered in literature –what lexical fields, groups of symptoms, musical genres, and kinds of tweets interact with each other. We can evaluate the importance of interactions between entities based on inferred virality. We study two quantities for each corpus: the overall relative impact of the interactions on the probability of an outcome and the contribution of each pair of clusters in the modification of outcome probabilities.

To evaluate the global impact of the interactions, we compute the relative change of probability according to the inferred virality for each triplet (i, j, x), noted $V_{i,j,x}$ and average this quantity over all the triplets in the corpus. We note this quantity $\bar{V}$:

$$\bar{V} = \frac{1}{|R^\circ|} \sum_{(i,j,x)\in R^\circ} \underbrace{\frac{|P_{i,i}(x) - P_{i,j}(x)|}{P_{i,i}(x)}}_{V_{i,j,x}} \tag{II.21}$$

where $P_{i,j}(x) = \sum_{k,l} \theta_{i,k}\theta_{j,l} p_{k,l}(x)$ denotes the probability of outcome x given the entities i and j; as shown in the previous section, the diagonal elements $P_{i,i}(x)$ account for the virality of i on x.

First, we report in Fig. II.7 the distribution $V_{i,j,x}$ over all triplets in every dataset. In this figure, we plot the histogram of the relative impact of interactions on the base virality of outputs. Overall, we confirm the conclusions of (Myers and Leskovec, 2012) that most interactions do not lead to any notable change in the probability of an output. However, a non-negligible part of them leads to changes in probability up to 500% the base virality. The overall impact of interactions if the weighted average of this histogram, calculated in Eq. II.21. These results are shown in Fig. II.8.

For every corpus, the interaction between entities exerts a non-negligible overall influence $\bar{V}$ on the probability of outputs. Those results confirm previous work done on interactions modelling, stating the importance of taking interactions into account when analysing real-world datasets (Myers and Leskovec, 2012). Interactions increase the virality of an output by a factor of 2.58 in the PubMed corpus, 2.78 in the Twitter corpus, 1.73 in the Spotify corpus and 1.57 in the Reddit corpus. Interactions have a greater effect on output probabilities for PubMed and Twitter corpora, and a lesser role for the Spotify and Reddit ones. Besides, our model applied to a dataset where interactions do not play any role ($\bar{V} = 0$) reduces to the

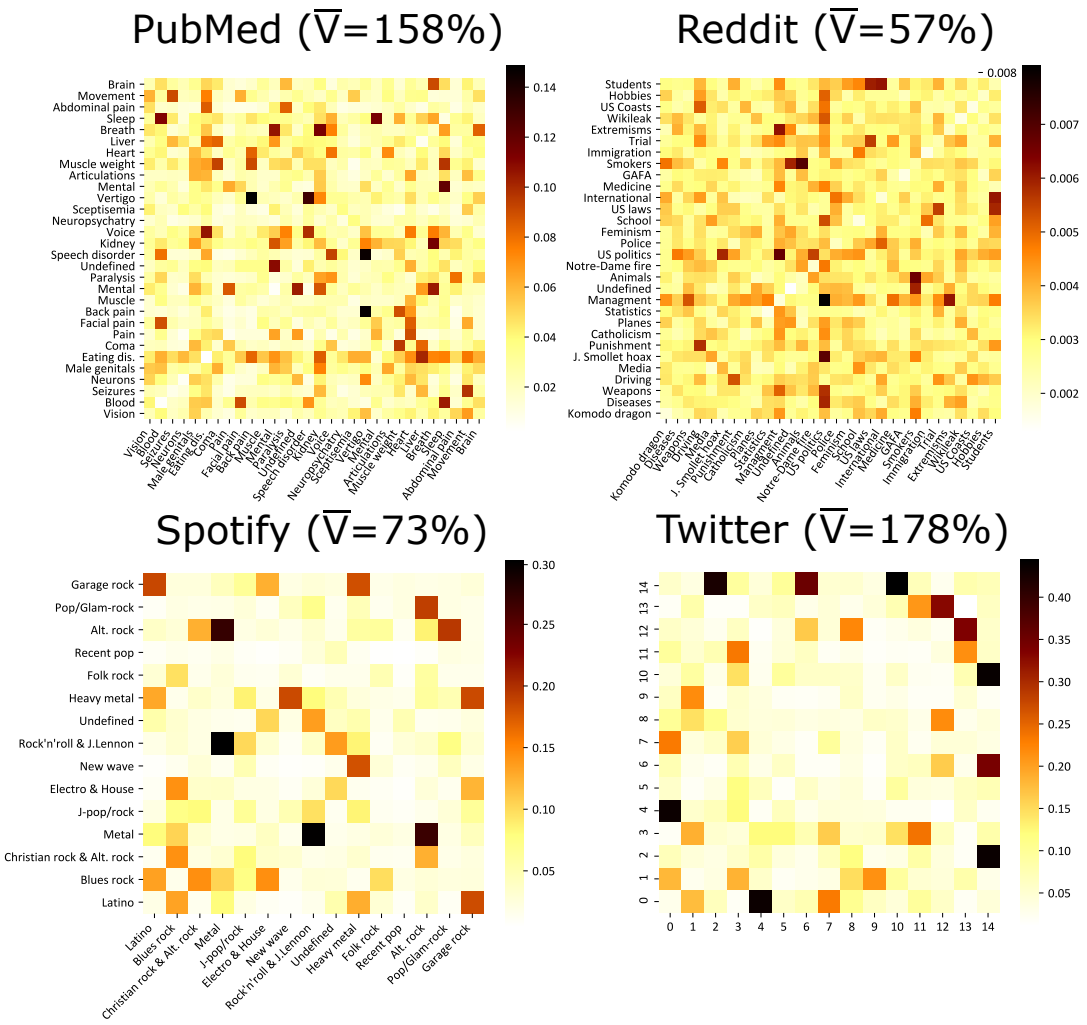

FIGURE II.8: **Importance of interactions** - Contribution of each pair of clusters $V_{k,l}$ and average impact of the interactions $\bar{V}$ (on the right) in outcome probabilities for each corpus. Clusters typically interact with a limited number of others; these interactions still play a significant role in outcomes probabilities. The cluster have been annotated manually.

non-interacting MMSBM baseline. This metric therefore allows us to assert the importance of the interactions in a given corpus.

Which clusters interact

To evaluate clusters pair-interaction, we consider the following quantity:

$$V_{k,l} = \frac{\sum_{(i,j,x)\in R^\circ} \theta_{i,k}\theta_{j,l}|p_{k,l}(x) - P_{i,i}(x)|}{\sum_{(i,j,x)\in R^\circ} \theta_{i,k}\theta_{j,l}}$$

This quantity is the weighted average of the absolute change in output probability with respect to virality due to the interaction between every pair of clusters (k,l) for each possible pair of entities. The results are shown in Fig. II.8. Clusters have been annotated manually.

We see that most of the clusters do not interact with each other; the interactions essentially take place between a limited number of clusters. Typically, a cluster interacts significantly with only one or two other clusters in every corpus ("Vertigo"

and "Speech disorder" in PubMed, "Students" and "Schools" in Reddit, etc.). We also notice that in each corpus, the model forms some non-interacting cluster with low values of $V_{k,l}$ ("Neuropsychiatry" in PubMed, "Recent pop" in Spotify, etc.); for those, the probability of an output is essentially equal to the virality of this output. We also notice that the diagonal of the $V_{k,l}$ matrices comprises low values. The interaction of a group with itself leads to an output probability close to its entities' virality. To picture how this makes sense, we can imagine diagnosing a disease based on two ear-related symptoms ("earache" and "hearing disorders"): the diagnosis is likely to be related to the ear as we would have guessed with only one symptom (its probability equals the virality). Now imagine two symptoms of different kinds ("earache" and "speech disorder"): the diagnosis is then likely to be related to the brain and less to the ear, so the interaction lowers the base probability (virality) of the "ear disease" output and increases the one of the brain disease.

Being able to see in detail the extent to which interactions exert an influence in a corpus and between which categories they take place opens new perspectives in research. Models that allow explaining the underlying mechanisms are of interest for applied social sciences (Guimera, Llorente, and Sales-Pardo, 2012; Cobo-López, Godoy-Lorite, and Duch, 2018; Poux-Médard et al., 2021).

Entropy of membership

We now consider the membership entropy of the entities. It measures how entities' membership is spread over the clusters. When it is low, entities belong to a small number of clusters to a great extent; when it is high, it means entities belong to every cluster to roughly the same extent. We use the normalized Shannon entropy of memberships of user i as a metric, noted $S_i^{(m)}$:

$$S_i^{(m)} = \frac{1}{\log_2 \frac{1}{K}} \sum_k^K \theta_{i,t} \log_2 \theta_{i,t} \tag{II.22}$$

Here the lowest entropy reachable is 0, which corresponds to an entity belonging to only one group ; the largest is 1 corresponding to belonging to every cluster evenly (with probability $\frac{1}{K}$).

Overall, the entropy of memberships is low. The average entropy values per corpus are: 0.320 for PubMed (equivalent to belonging on average to 2-3 clusters), 0.324 for Twitter (2 clusters), 0.561 for Reddit (6-7 clusters) and 0.364 for Spotify (2-3 clusters). The small number of entities spread among clusters means that the clustering done by our model is easy to interpret –which eased the manual annotation of the clusters presented in the previous section.

II.2.3.e Conclusion

In most previous approaches to information spread, the effect of interactions between diffusing entities has been neglected. Here, we proposed an in-depth study of the IMMSBM (corresponding to the special case SIMSBM(2)) that allows us to investigate the detail of interactions strength. Note that the aim of IMMSBM includes but is not restricted to interaction modelling. Throughout this section, we also illustrated the interest of IMMSBM for recommender systems (Spotify and PubMed datasets).

Our conclusions specific to interactions in information spread come from the Twitter dataset. We show that their effect is not trivial (average relative change of

178%) and that taking them into account increases predictive performance (+0.06 AUCROC over the non-interacting baseline). However, these interactions appear to be sparse: only 5 pairs of clusters out of 105 possible pairs seem to have significant interactions. In most cases, virality seems to be enough to predict an output with good accuracy.

However, all the models discussed in this section exhibit a major flaw: they are all static. The data collected on Twitter and Reddit used in our experiments spans approximately over a month. There is no a priori reason for the underlying interaction mechanisms to remain the same over this period. Slicing this data into time intervals would provide deeper insights into the interaction mechanisms at stake and their evolution over time.

II.3 Dynamic interactions

II.3.1 Introduction

Dynamic networks are powerful tools to visualize and model interactions between different entities that can evolve over time. The network's nodes represent the interacting entities, and ties between these nodes represent an interaction. In many real-world situations, the strength of the ties can vary over time –on music streaming websites for instance, users' affinity with various musical genres can vary greatly over time (Kumar, Zhang, and Leskovec, 2019; Villermet et al., 2021). Such network is said **dynamic**.

Now, every interaction does not have the same importance. A music listener can like both Rock and Jazz, but might prefer one over the other. This person's tie to each musical genre does not have the same intensity; each tie is associated with a number, representing the strength of the interaction. The network is said to be **dynamic and valued**.

However, in many real-world situations, valued networks are not enough to fully represent a given situation. The same music listener as before can have different opinions on musical genres; she can like it, dislike it, be bored of it, like to listen to it only in the morning, or at night, etc. Each of these relations can be represented by its own tie in the network, each having its own value. The network is said to be **dynamic, valued and labelled**.

Inferring dynamic, valued and labelled networks

Networks are high-dimensional objects, whose exact inference is a difficult problem –as stated in the previous section. Several ways to achieve this task have been proposed, the Stochastic Block Models (SBM) family being one of the most popular approaches, see Chapter II and (Holland, Laskey, and Leinhardt, 1983; Guimerà and Sales-Pardo, 2013; Cobo-López, Godoy-Lorite, and Duch, 2018). The underlying assumption is that certain sets of nodes behave similarly. They can be described using a single mathematical object instead of individual ones: the so-called clusters. Instead of modelling every edge for every node in a network, only the edges between these sets are modelled, which makes the task much more tractable. Each cluster is then associated with a labelled edge, and each node is associated with a cluster. A variant of SBM that allows more expressive power is called the Mixed-Membership SBM (MMSBM), where each node can belong to several clusters in different proportions at the same time (Yuchung and George, 1987; Airoldi et al., 2008; Godoy-Lorite

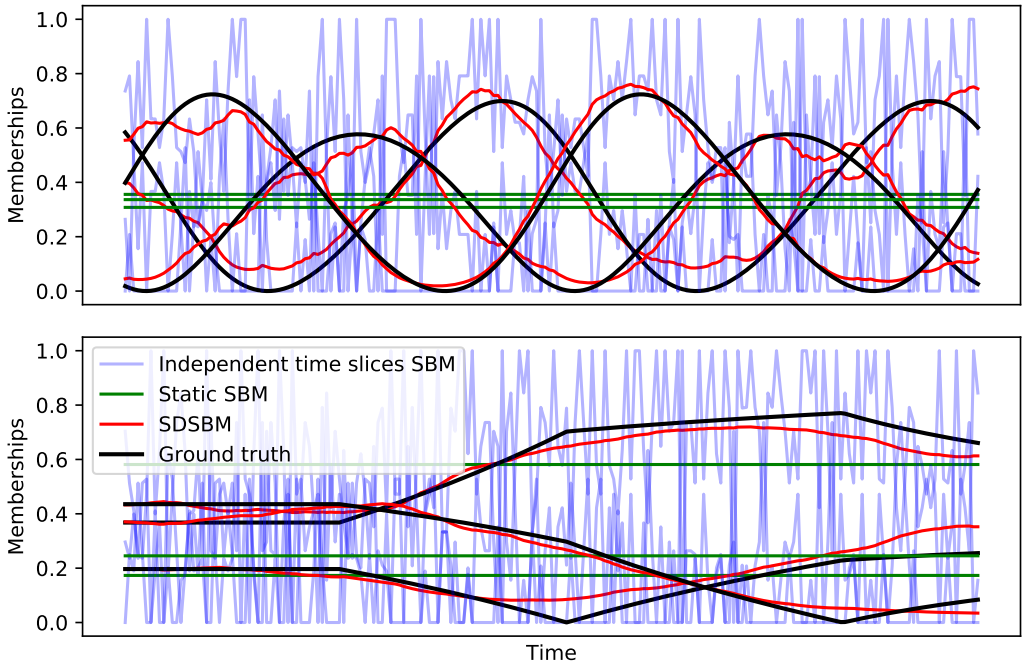

FIGURE II.9: **Users' attachment to groups can vary over time** — A music listener could cyclically prefer Rock, Jazz, or Pop music (top), or listen to either of these without any specific pattern (bottom). For 200 epochs each containing only 5 observations, our approach (in red) infers any smooth dynamic membership pattern and does it more accurately than static models (in green (Godoy-Lorite et al., 2016)) and models that consider each time slice independently (in blue (Tarrés-Deulofeu et al., 2019)).

et al., 2016; Tarrés-Deulofeu et al., 2019). A major advantage of this model's family is that it yields readily interpretable results (interpretable clusters), unlike most existing neural-network-based modelling of labelled networks (Fan et al., 2021).

Overview of the proposed approach

The goal of this section is therefore to provide a way to infer networks that are dynamic, valued and labelled by assuming a block structure – by using a mixed-membership SBM. After a careful review of dynamic network inference literature, it seems that no prior work tackles such a task. Although some previous works handle similar problems, their adaptation to the case at hand is not trivial. Besides, the solution we propose here is conceptually much simpler than those present in the literature. Last but not least, our method is readily pluggable into most existing MMSBM for labelled and valued networks (such as SIMSBM and its special cases, comprising the IMMSBM and each of (Airoldi et al., 2008; Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019)) as their temporal extension.

We develop an EM optimization algorithm that scales linearly with the size of the dataset, demonstrate the effectiveness of our approach on both synthetic and real-world datasets, and further detail a possible application scenario.

II.3.2 State of the art and limitations

II.3.2.a Notations

We consider a network of I nodes and O labels. All the clustering models discussed in this section can be represented using a matrix $\theta \in \mathbb{R}^{I \times K}$ accounting for memberships over a set of K possible clusters for each of I nodes, and a block-interaction matrix $p \in \mathbb{R}^{K \times K \times O}$ linking each of K clusters to every of O labels. Both θ and p can vary over time. The network is said to be unlabelled when $O = 1$, and binary (as opposed to valued) if an edge can only exist or not exist.

II.3.2.b Dynamic unlabelled networks - Single-membership

Single-membership SBM considers a membership matrix such as $\theta \in \{0; 1\}^{I \times K}$: each membership vector equals 1 for one cluster, and 0 everywhere else. Literature also speaks of "hard" clustering. The authors in (Xu and Hero, 2014; Xu and Hero, 2013) proposed to model a binary unlabelled dynamic network using a label-switching inference method along with a Sequential Monte Carlo algorithm (Jin et al., 2021). Both the membership and the interaction matrices can vary over time, thus supposing two independent underlying Markov processes (Jin et al., 2021).

In (Yang et al., 2010; Tang and Yang, 2014), the authors propose to model a binary dynamic and unlabelled network. The cluster interaction matrix p can vary over time while keeping the memberships θ fixed over time. The entries of p are drawn from a Dirichlet distribution and expressed as a Chinese Restaurant Process. This process converges to a Dirichlet distribution and allows to infer a potentially infinite number of clusters. This model is therefore non-parametric and inferred using an MCMC algorithm.

A novel way of tackling the problem has been proposed in (Matias and Miele, 2017; Matias, Rebafka, and Villers, 2018), where the authors propose to model the cluster interaction and membership matrices as Poisson processes, that explicitly model the temporal dependency without slicing the time dimension into episodes. The method allows to infer varying membership *and* interaction matrices for dynamic binary or valued networks, but their results have shown that allowing both to vary simultaneously leads to identifiability and label switching issues (Funke and Becker, 2019). This conclusion seems reasonable, given none of these SBM algorithms can reach a global optimum in the likelihood function. During optimization, a model with both membership θ and interaction p matrices is all the more likely to get stuck into a local maximum if both can vary over time.

Finally, we can mention the existence of SBM variants that account for dynamic degree-correction (Wilson, Stevens, and Woodall, 2019) or that enforce a scale-free characteristic (Wu et al., 2019). However, all these methods consider unlabelled networks and consider a hard clustering which does not allow for as much expressive power as the Mixed-Membership approaches.

II.3.2.c Dynamic unlabelled networks - Mixed-membership

Mixed-membership SBM considers a membership matrix such as $\theta \in \mathbb{R}^{I \times K}$, where each membership vector is normalized to 1. Literature also refers to it as "soft" clustering.

Similar to (Yang et al., 2010; Tang and Yang, 2014), a method for inferring dynamical binary unlabelled networks has been proposed in (Fan, Cao, and Da Xu, 2015). The membership vector of each piece of information is drawn from a Chinese

Restaurant Process (CRP) according to the number of times a node has already been associated with each cluster before. The resulting process yields a distribution over an infinity of available clusters. The formulation as a CRP arises naturally because the prior on membership vectors is typically a Dirichlet distribution; CRP naturally converges to this distribution. The block-interaction matrix p does not evolve with time. The article shows a complexity analysis that suggests the methods run with a complexity of $\mathcal{O}(N^2)$ which makes it unfit for large-scale real-world applications.

The work the most closely related to ours is (Xing, Fu, and Song, 2010). This seminal work proposed the dMMSB as a way to model dynamic binary unlabelled networks using a variational algorithm (Lee and Wilkinson, 2019). To do so, the authors modify the original MMSBM (Airoldi et al., 2008) to consider a logistic normal distribution as a prior on the membership vectors $\vec{\theta}_i$. This choice allows to model correlations between membership vectors' evolution (Ahmed and Xing, 2007). The membership vectors are then embedded in a state-space model, which is a space where we can define a linear transition between two time points for a given variable. The authors define such a trajectory for the membership vectors as a linear function of the previous time point. The trajectory is estimated and smoothed using a Kalman Filter. This approach is the most closely related to ours, as it proposes to consider the temporal dependency directly in a prior distribution over memberships, noted $P(\theta)$.

However, this model is not fit for the task at hand. It considers unlabelled and binary networks (Lee and Wilkinson, 2019), and extension to labelled and valued networks is not trivial. The proposed optimization algorithm requires a loop until convergence at each EM iteration, making it unable to handle large datasets. It is not designed to let the clusters interaction matrix p depend on time, which we alleviate here. And most importantly, it assumes a linear transition between time steps in the state space, while we do not assume any kernel function in our proposed approach. (Xing, Fu, and Song, 2010) has been extended to consider $P(\theta)$ as a logistic normal mixture prior (Ho, Song, and Xing, 2011), which improves its expressive power. However, it does not address the aforementioned points.

II.3.2.d Static labelled networks - Mixed-membership

Recent years saw a rise of Bayesian methods for inferring static valued labelled networks using MMSBM variants (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019; Poux-Médard et al., 2021). Note that in (Tarrés-Deulofeu et al., 2019), the authors consider a temporal slicing of the data and consider each slice as independent from the others; a time slice is considered as a node in a tripartite network. We will compare our approach to this modelling later. We do not present the details of these works here, as we will develop their in-depth functioning in Section II.3.3.

The method we propose here uses these works as a base. This section focuses on making the prior probability of both θ and p time-dependent to model these parameters' dynamics. We provide a ready-to-use temporal plug-in for each of the works we have just presented in this section and in Section II.2. It applies to dynamical, valued, and labelled networks in a mixed-membership context, and inference is conducted with a scalable variational EM algorithm.

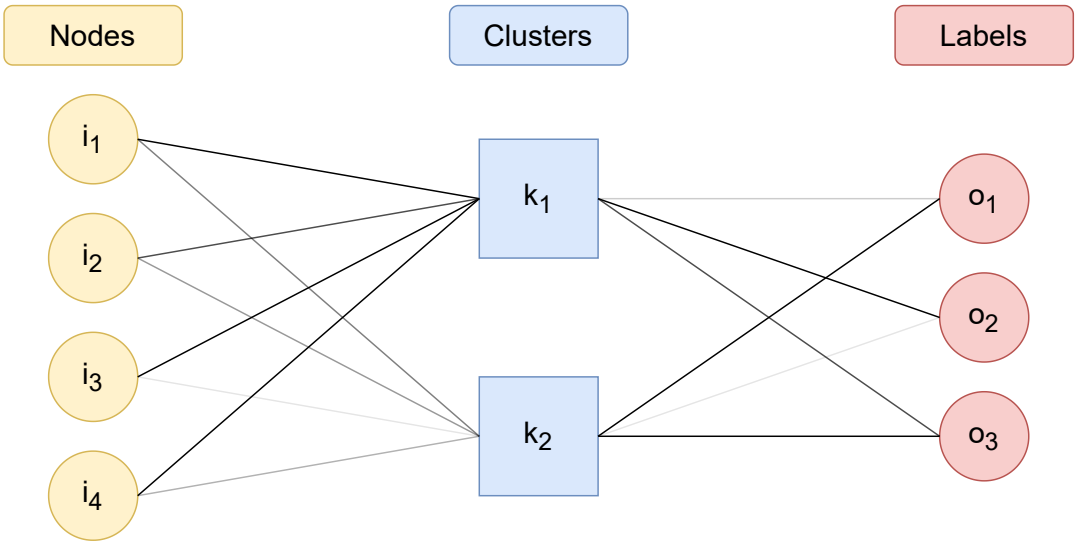

FIGURE II.10: Illustration of the SIMSBM(1), which is the base model coupled to the SDSBM prior. Nodes are associated with clusters, and clusters are associated with labels. Ties between each entity represent a membership and can take values between 0 and 1.

II.3.3 SDSBM – Simple Dynamic labelled MMSBM

II.3.3.a Base model

In this section, we present the simplest form of a labelled MMSBM, or SIMSBM(1), for demonstration purposes. We illustrate the structure of this model in Fig. II.10. Its derivation trivially extends to (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019; Poux-Médard et al., 2021), the IMMSBM, and any of the SIMSBM iterations discussed Section II.2.2 (we detail our work's inclusion in these more complex models in Appendix, Section I.6).

We consider a set of I nodes that can be associated with O possible labels on a discrete time interval, or epoch, written t. We assume that nodes can be efficiently represented as a mixture of K available clusters at each time step, each of which is in turn linked to the labels. The membership of each of I nodes into each of the K possible clusters at time t is encoded in the membership matrix $\theta^{(t)} \in \mathbb{R}^{I\times K}$. One vector $\theta_i^{(t)}$ represents the probability that i belongs to any of the K clusters at time t, and is normalized as:

$$\sum_{k\in K} \theta_{i,k}^{(t)} = 1 \; \forall i, t \tag{II.23}$$

The probability of each cluster to be associated with each label at time t is encoded in the matrix $p^{(t)} \in \mathbb{R}^{K\times O}$. An entry $p_k^{(t)}(o)$ represents the probability that cluster k is associated with label o at time t, and thus is normalized as:

$$\sum_{o\in O} p_k^{(t)}(o) = 1 \; \forall k,^{(t)} \tag{II.24}$$

Finally, the probability that a node i is associated with label o at time t (this can be seen as the probability an edge between i and o exists at time t) is written:

$$P^{(t)}(i \to o) = \sum_{k\in K} \theta_{i,k}^{(t)} p_k^{(t)}(o) \tag{II.25}$$

Given a set R° of observed triplets (i, o, t), the model's posterior distribution can be written (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019):

$$P(\theta, p|R^\circ) \propto P(R^\circ|\theta, p) \prod_t P(\theta^{(t)}) P(p^{(t)}) \tag{II.26}$$

$$= \prod_{(i,o,t)\in R^\circ} \sum_{k\in K} \theta_{i,k}^{(t)} p_k^{(t)}(o) \prod_t \left(\prod_i P(\theta_i^{(t)}) \prod_k P(p_k^{(t)}) \right)$$

Now, before we describe the optimization procedure, we must choose the priors $P(\theta^{(t)})$ and $P(p^{(t)})$.

II.3.3.b Simple Dynamic prior

We formulate the prior distribution over $\theta^{(t)}$ and $p^{(t)}$ following a simple assumption: the parameters at a given time are unlikely to vary abruptly at small time scales — *the apple does not fall far from the tree*. It means an entry $\theta_{ik}^{(t_1)}$ should not differ so much from $\theta_{ik}^{(t_2)}$ for every t_2 close enough to t_1. Such entries close to a reference time are called **temporal neighbours**.

Our *a priori* knowledge on each entry $\theta_i^{(t)}$ and $p_k^{(t)}$ is that they should not differ significantly from their temporal neighbours. This is a fundamental difference with (Xing, Fu, and Song, 2010), where the next parameters values are estimated using a Kalman Filter that only considers the previous time step. Moreover, the authors assume a linear transition function, while we do not make such a hypothesis. An illustration of the proposed approach is given in Fig. II.11, where the prior probability of a membership vector depends on its temporal neighbours (in white).

Dirichlet distribution

Since each entry $\theta_i^{(t)}$ and $p_k^{(t)}$ is normalized to 1, we consider a Dirichlet distribution as a prior, which naturally yields normalized vectors such that $\sum_n x_n = 1$. It reads:

$$Dir(x|\alpha) = \frac{1}{B(\alpha)} \prod_n x_n^{\alpha_n - 1} \tag{II.27}$$

where $B(\cdot)$ is the multivariate beta function. In Eq. II.27, the vector α is called the concentration parameter and must be provided to the model. Importantly to the model introduced in this section, it allows to control the mode and the variance of the Dirichlet distribution.

We consider a concentration parameter such as $\alpha = 1 + \beta\alpha_0$, so that for $\beta = 0$ we recover a uniform prior over the simplex, and $\alpha > 1$ so that the prior has a unique mode. The most frequent value drawn from Eq. II.27 (or mode) is $\mathbb{M}(x_n) = \frac{\alpha_n - 1}{\sum_n'(\alpha_n' - 1)} = \frac{\alpha_{0,n}}{\sum_n' \alpha_{0,n}}$. We recover a uniform prior for $\beta = 0$; the variance vanishes with $\beta \gg 1$ as $\frac{1}{\beta}$. The effect of various values of β on the prior distribution is illustrated in Fig. II.11.

Prior's mode

Our main assumption states that $\theta_i^{(t)}$ and $p_k^{(t)}$ do not vary abruptly. To enforce this, we define their prior probability mode with respect to their close temporal neighbours. The hyper-parameter β controls the variance of the prior –that is, how much

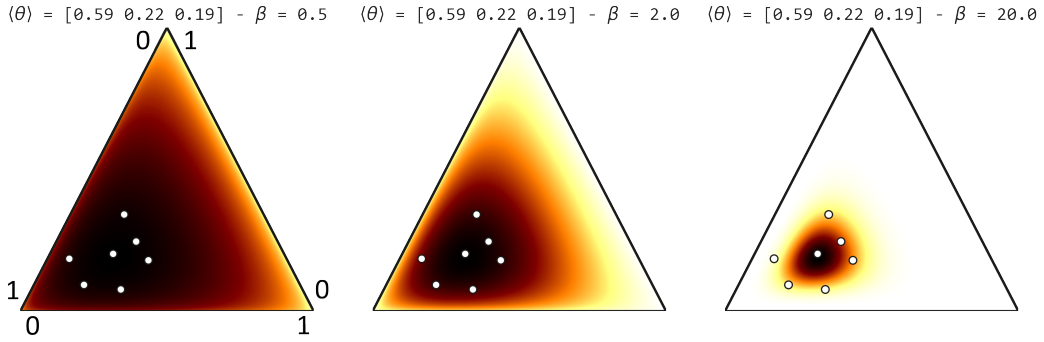

FIGURE II.11: **Prior probability on a membership vector for various values of β according to temporal neighbourhood** — Darker means higher probability. Projected on a simplex tri-space (each of 3 axes ranges from 0 to 1). The white dots represent the temporal neighbours of the considered 3D vector. Their average is given as $\langle\theta\rangle$ using a uniform weight function $\kappa(t,t')$ for illustration purpose. β controls the variable's prior variance around its neighbours.

it should impact the inference procedure. We express the Simple Dynamic prior parameters for $\theta_i^{(t)}$ as:

$$\alpha_{i,k}^{(t,\theta)} = 1 + \beta \underbrace{\left(\frac{\sum_{t' \neq t} \kappa(t,t') \theta_{i,k}^{(t')}}{\sum_{t' \neq t} \kappa(t,t')} \right)}_{\langle \theta_{i,k}^{(t)} \rangle} \tag{II.28}$$

where $\kappa(t,t')$ is a weight function, and $\alpha^{(t,\theta)}$ corresponds to the concentration parameter for θ at time t. In following experiments, we define the weight function as $\kappa(t,t') = \frac{N_{t'}}{|t-t'|}$, where $N_{t'}$ is the number of observations made at time t', so that temporal neighbours' influence decrease as the inverse of temporal distance. We illustrate the influence of this particular kernel function on the prior probability on membership at all times in Fig. II.12. In particular we see that with this expression, the prior probability variance collapses to 0 when the considered time is very close to a temporal neighbour.

The mode of the prior over a variable is then the average value of its the temporal neighbours weighted by $\kappa(t,t')$, noted $\langle\theta_{i,k}^{(t)}\rangle$. Note that this holds because $\sum_k \langle\theta_{i,k}^{(t)}\rangle = 1 \; \forall i,t$. Besides, the prior variance is a decreasing function of β; when $\beta = 0$ the prior is uniform over the simplex, and when $\beta \to \infty$ the variance goes to 0, as illustrated Fig. II.11. The same reasoning holds for $p_k^{(t)}$, with prior parameters $\alpha_{k,o}^{(t,p)} = 1 + \beta \langle p_k^{(t)}(o) \rangle$.

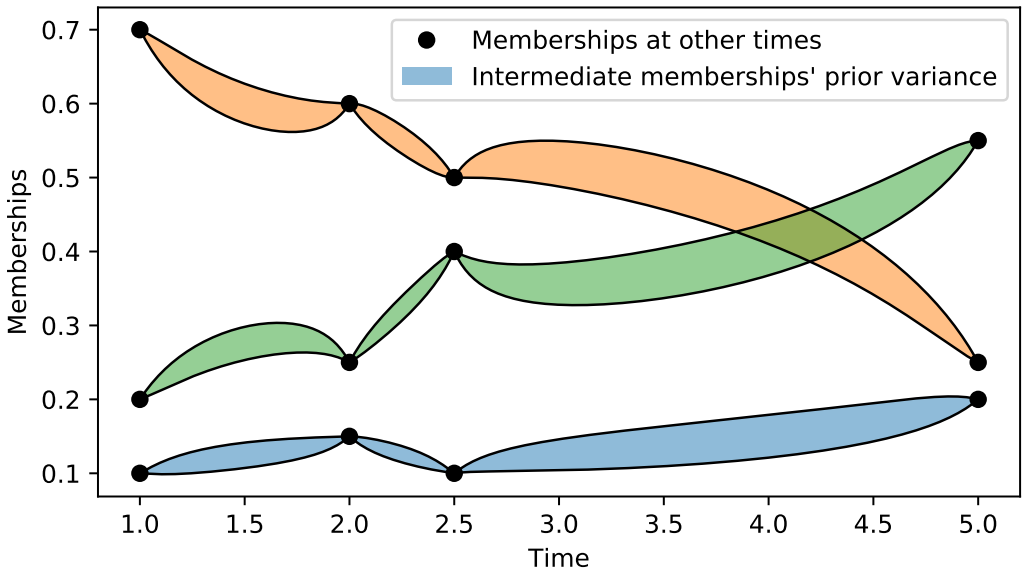

FIGURE II.12: **Prior probability's variance on memberships at all times according to the temporal neighbourhood** — Variance of the prior over a membership entry (filled curves, we represented 3 such entries as illustration) as a function of time, given some temporal neighbours (black dots). This illustration considers an averaging kernel as $\kappa(t,t') = \frac{1}{|t-t'|}$. When inferring a parameter $x^{(t)}$ at a time t, the variance of its prior probability $P(x^{(t)})$ depends on t relative to the temporal neighbours. Here for instance, the variance is null at $t = 2$ because $\kappa(t,t')$ diverges, and so does $\alpha^{(t)}$, hence the variance collapsing to 0.

Priors expression

Finally, we give the final log-priors on $\theta_i^{(t)}$ and $p_k^{(t)}$:

$$P(\theta_i^{(t)} | \{\theta_{i,k}^{(t')}\}_{t' \neq t}) \propto \prod_k \theta_{i,k}^{(t)\,\beta\langle\theta_{i,k}^{(t)}\rangle} \tag{II.29}$$

$$P(p_k^{(t)}(o) | \{p_k^{(t')}(o)\}_{t' \neq t}) \propto \prod_o p_k^{(t)}(o)^{\beta\langle p_k^{(t)}(o)\rangle}$$

We omitted the normalisation factor for clarity –it does not influence the inference procedure.

II.3.3.c Inference

E step

We develop an EM inference procedure for maximizing the log-posterior distribution defined in Eq. II.26. The expectation step computes the expected probability of a latent variable (here a cluster k) being chosen given each entry of R°. Since such latent variables do not appear in the priors expressions, the expectation step remains unchanged by the introduction of the Simple Dynamic Priors; in general, prior distributions do not intervene in the computation of the expectation step (Bishop, 2006). The E step for such labelled networks has already been derived on many occasions. Therefore, we give the expectation step equation without explicit derivation (full

derivation is however provided in Appendix, Section I.4, and in Section II.2.2.b.1 and Section II.2.3.b.1).

The expectation of the latent variable k given an observation $(i, o, t) \in R^\circ$, written $\omega_{i,o}^{(t)}(k)$, is defined as:

$$\omega_{i,o}^{(t)}(k) = \frac{\theta_{i,k}^{(t)} p_k^{(t)}(o)}{\sum_{k'} \theta_{i,k'}^{(t)} p_{k'}^{(t)}(o)} \tag{II.30}$$

Using this expression, we can rewrite the log-likelihood $\log P(R^\circ|\theta, p)$ as (Bishop, 2006; Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019):

$$\log P(R^\circ|\theta, p) = \sum_{(i,o,t)\in R^\circ} \sum_{k\in K} \omega_{i,o}^{(t)}(k) \log \frac{\theta_{i,k}^{(t)} p_k^{(t)}(o)}{\omega_{i,o}^{(t)}(k)} \tag{II.31}$$

M step

Taking back the first line of Eq. II.26 and substituting with Eq. II.29 and Eq. II.31, we get an unconstrained expression of the posterior distribution. We introduce Lagrange multipliers to account for the constraints of Eq. II.23 ($\phi_i^{(t)}$) and Eq. II.24 ($\psi_i^{(t)}$), and finally compute the maximization equations with respect to the model's parameters. Starting with the membership matrix entries $\theta_{i,k}^{(t)}$:

$$\begin{aligned}
&\frac{\partial \left(\log P(\theta, p|R^\circ) - \sum_{i',t'} \phi_{i'}^{(t')} (\sum_{k'} \theta_{i',k'}^{(t')} - 1) \right)}{\partial \theta_{i,k}^{(t)}} = 0 \\
&\Leftrightarrow \sum_{o\in\partial(i,t)} \frac{\omega_{i,o}^{(t)}(k)}{\theta_{i,k}^{(t)}} + \frac{\beta \langle \theta_{i,k}^{(t)} \rangle}{\theta_{i,k}^{(t)}} - \phi_i^{(t)} = 0 \\
&\Leftrightarrow \sum_{o\in\partial(i,t)} \omega_{i,o}^{(t)}(k) + \beta \langle \theta_{i,k}^{(t)} \rangle = \phi_i^{(t)} \theta_{i,k}^{(t)} \\
&\Leftrightarrow \sum_{o\in\partial(i,t)} \underbrace{\sum_k \omega_{i,o}^{(t)}(k)}_{=1 \text{ (Eq. II.30)}} + \beta \underbrace{\sum_k \langle \theta_{i,k}^{(t)} \rangle}_{=1 \text{ (Eq. II.23)}} = \phi_i^{(t)} \\
&\Leftrightarrow \frac{\sum_{o\in\partial(i,t)} \omega_{i,o}^{(t)}(k) + \beta \langle \theta_{i,k}^{(t)} \rangle}{N_{i,t} + \beta} = \theta_{i,k}^{(t)}
\end{aligned} \tag{II.32}$$

where $\partial(i,t) = \{o|(i,\cdot,t) \in R^\circ\}$ is the subset of labels associated with both i and t, and $N_{i,t} = |\partial(i,t)|$ is the size of this set. Note that for $\beta = 0$ we recover the M-step of standard static MMSBM models, see (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019) and Section II.2.

The derivation of the M-step for the entries $p_k^{(t)}(o)$ is identical and yields (see Appendix, Section I.5, for details):

$$p_k^{(t)}(o) = \frac{\sum_{(i,t)\in\partial o} \omega_{i,o}^{(t)}(k) + \beta \langle p_k^{(t)}(o) \rangle}{\sum_{(i,o,t)\in R^\circ} \omega_{i,o}^{(t)}(k) + \beta} \tag{II.33}$$

II.3.3.d Discussion

Easy to use

We briefly review some key points of the Simple Dynamic prior. Its introduction induces minor changes in the existing works on MMSBM for labelled networks. Compared to (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019) and Section II.2, its introduction boils down to simply adding a term $\beta\langle x\rangle$ to the numerator of maximization equations, and the corresponding normalizing term β to the denominator. This way, our approach is ready-to-use to make these models, and variants built on them, dynamic –explicit derivations are provided in Appendix, Section I.6.

Flexible dynamic modelling

The prior allows us to consider that some parameters are dynamic and that others are not. For instance, when several membership matrices are involved, as in (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019)), setting $\beta = 0$ for some makes them time-invariant (or universal). The SD prior also allows choosing whether the block-interaction tensor p is dynamic.

In general, β does not have to be identical for every membership matrix, or even every entry i of each of them. Moreover, it is not mandatory for β to be constant over time. A dynamic parameter $\beta(t)$ is especially relevant when epochs are not evenly spaced over time; $\beta(t)$ would typically lower the variance (by increasing) when temporal neighbours are closer (right plot in Fig. II.11).

To summarize, β allows controlling the time scale over which variables may vary. This allows greater modelling flexibility, allowing to jointly model universal variables ($\beta = 0$) and dynamical ones ($\beta \neq 0$).

Tuneable temporal dependence

Finally, the choice of the averaging kernel function $\kappa(t, t')$ is important. It allows choosing the range over which the inference of a variable should rely on its temporal neighbours. A formulation as the inverse of time difference seems relevant: the weight of a neighbour appearing at a time δt later should diverge as $\delta t \to 0$, so that continuity is ensured. Besides, it allows controlling the smoothness of the curve with respect to time by tuning the weight function as $\kappa(t, t') = \frac{N_{t'}}{|t-t'|^a}$ where $a = 1, 2, ...$ for instance, where $N_{t'}$ is the number of observations in the time slice t'.

Overall, the Simple Dynamic prior works by inferring the variables using both microscopic and mesoscopic temporal scales. If a time slice t has very few observations but some of its neighbours have a greater number of them for instance; learning the parameters at t is helped mostly by the population of its closest ($\frac{1}{|\Delta t|}$) and most populated ($N_{t'}$) epochs, and less influenced by further and less populated epochs. This is what is illustrated in Fig. II.12.

II.3.3.e Experiments

Synthetic data

In this section, we address several situations in which our method (abbreviated SDSBM for Simple Dynamic MMSBM) could be useful. Experiments are run for $I = 100$, $K = 3$ and $O = 3$, which are standard testing parameters in the literature of dynamic networks inference (Fan, Cao, and Da Xu, 2015; Matias and Miele, 2017).

We choose to infer a dynamic membership matrix $\theta^{(t)}$ and to provide a universal block-interaction matrix p $\forall t$. Note that the model yields good performances when p also has to be inferred, but due to identifiability and label-switching issues raised in (Matias and Miele, 2017), there is no unbiased way to assess the correctness of the inferred memberships values. An experiment where p is inferred jointly to θ is provided and discussed in Appendix, Section I.7. The expression we use for this matrix p is given Eq. II.34. We systematically test two variation patterns for θ: a sinusoidal pattern (Fig. II.9-top) and a broken-line pattern (Fig. II.9-bottom). Each pattern is generated with a different coefficient for each item; the memberships still sum to 1 at all times.

$$p = \begin{bmatrix} 1-s & s & 0 \\ 0 & 1-s & s \\ s & 0 & 1-s \end{bmatrix} \tag{II.34}$$

To the best of our knowledge, the only attempt to model dynamic parameters in labelled valued and dynamic networks using a MMSBM is (Tarrés-Deulofeu et al., 2019). In this work, each epoch is modelled independently from the others. We refer to this baseline as the "No coupling" or "**NC**" baseline. For reference, we also compare to a baseline that does not consider the temporal dimension and infers a single universal value for each variable (classical **MMSBM**) (Godoy-Lorite et al., 2016) and the models Section II.2.

We systematically perform a 5-fold cross-validation. The model is trained on 80% of the data, β is tuned using 10% as a validation set, and the model is evaluated on the 10% left. We choose as metrics the AUCROC and RMSE on the real values of θ (black line in Fig. II.9). The procedure is repeated 5 times; the error bars reported in the experimental results represent the standard error over these folds.

SDSBM unveils complex temporal patterns

In Fig. II.13a., we consider 1,000 observations for each item $i \in I$ and vary the number of epochs from 10 to 1,000. In the expression of p, s is set to 0.05. For both the sinusoidal and line-broken memberships, the model shows better predictive performances (in terms of AUCROC) than the proposed baselines. Interestingly, the SDSBM performances remain stable as the number of epochs increases unlike the NC baseline, which means it alleviates a bias of the temporal modelling proposed in (Tarrés-Deulofeu et al., 2019). The RMSE with respect to the true parameters remains low over the whole range of the tested number of epochs. The RMSE increases as the number of epochs grows because the number of parameters to estimate increases with it; this makes the inference more subject to local variations, which in turn mechanically increases the RMSE. Overall, SDSBM recovers dynamic variations of the membership vectors with superior performance; a sample of the inferred dynamic memberships is shown in Fig. II.9.

SDSBM works with little data

A major problem that arises when considering temporal data is the scarcity of observations, because slicing implies reducing the number of observations in each slice. This concern mostly arises in social sciences, where data retrieval cannot be automated and requires long-lasting human labour. Here, we demonstrate that our method works in challenging conditions when data is scarce. In Fig. II.13b., we vary the number of observations available for each item from 100 to 10.000, distributed

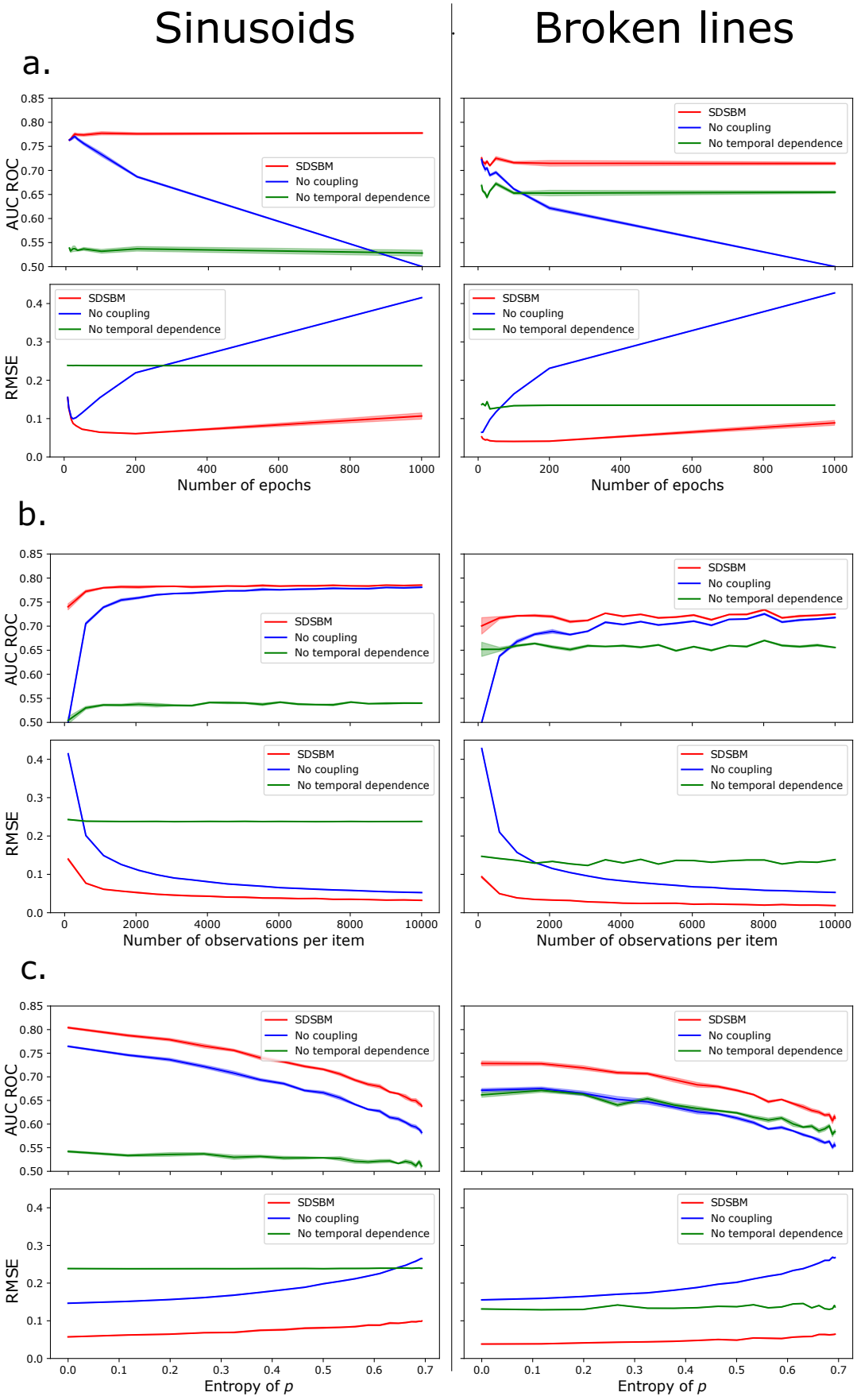

FIGURE II.13: **Results on synthetic data — (a.)** SDSBM retrieves the correct dynamic memberships and is little influenced by the data slicing. **(b.)** SDSBM works well on tiny datasets. **(c.)** SDSBM retrieves correct dynamic memberships in challenging situations.

TABLE II.5: **Numerical results on real-world datasets** — Metrics abbreviations stand for the area under the ROC curve (ROC), the Average Precision (AP), the Normalized Coverage Error (NCE). Metrics for models stand for Simple Dynamic SDM (SDSBM), No Coupling baseline (NC) and the classical static mixed membership SBM (MMSBM). Overall, our approach allows for a higher predictive power. The standard error over the folds is given in standard notation – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$. The models presented in this chapter are underlined.

		ROC	AP	NCE
Epi	SDSBM	**0.9025(11)**	**0.3700(17)**	**0.1151(11)**
	NC	0.8420(22)	0.3435(36)	0.1582(19)
	MMSBM	0.8597(12)	0.2141(16)	0.1451(13)
Lastfm	SDSBM	**0.8942(8)**	**0.0168(1)**	**0.1284(11)**
	NC	0.8393(5)	0.0157(2)	0.1785(7)
	MMSBM	0.8647(5)	0.0115(2)	0.1493(4)
Wiki	SDSBM	**0.9759(2)**	**0.0659(9)**	**0.0459(3)**
	NC	0.9092(7)	0.0608(10)	0.1195(8)
	MMSBM	0.9576(7)	0.0622(4)	0.0565(8)
Reddit	SDSBM	**0.9803(3)**	**0.4295(54)**	**0.0312(3)**
	NC	0.8508(5)	0.3598(17)	0.1846(7)
	MMSBM	0.9798(2)	**0.4269(40)**	0.0322(3)

over 100 epochs. Thus, in the most challenging situation, there is only one observation per epoch used to determine I dynamic memberships over 3 clusters. In the expression of p, s is set to 0.05. We see Fig. II.13b. that for both patterns, the predictive power of SDSBM remains high in such conditions. Moreover, the RMSE on the true dynamic memberships in this case is fairly low and decreases rapidly as the number of observations increases. When the number of observations is high, the "no coupling" baseline (Tarrés-Deulofeu et al., 2019) reaches the performances of SDSBM. This is because as the number of observations in each slice goes to infinity, the models need to rely less on temporal neighbours. However, even for 10.000 observations per item (100 observations per epoch), SDSBM yields better results than the proposed baselines. As an illustration, the results presented in Fig. II.9 have been inferred using only 5 observations per epoch.

SDSBM handles highly stochastic interaction patterns

Finally, we control the deterministic character of the block-interaction matrix p by varying s. We express such character as the mean entropy of p $\langle S(p) \rangle$ with respect to its possible outputs: $\langle S(p) \rangle = \frac{1}{K} \sum_{k \in K} \sum_o p_k(o) \log p_k(o)$. The maximum entropy for the proposed expression of p is reached for $s = 0.5$. We consider 1,000 observations spread over 100 epochs. We show in Fig. II.13c. that the predictive performance of all three methods drops as the entropy increases. This is expected, as the observations are generated from the true model with a higher variance; each observation becomes less informative about the underlying generative structure as s grows. However, the RMSE on the real parameters inferred using SDSBM remains low even at the maximum entropy, meaning the model recovers the correct membership parameters.

Real-world data

Finally, we demonstrate the validity of our approach on real-world data to argue for its usefulness and scalability. SDSBM builds on previous works on labelled

MMSBM and shares the same linear complexity $\mathcal{O}(|R^\circ|)$ with $|R^\circ|$ the size of the dataset (Godoy-Lorite et al., 2016). For our experiments, we consider the recent and documented datasets from (Kumar, Zhang, and Leskovec, 2019), namely the **Reddit** dataset (10.000 users, 984 labels, ∼670k observations), the **LastFm** dataset (980 users, 1000 labels, ∼1.3M observations) and the Wikipedia (**Wiki**) dataset (8227 users, 1000 labels, ∼157k observations). The Reddit and Wikipedia datasets contain 1 month of data; we slice them in 1 day long temporal intervals. The LastFm dataset spans over approximately 5 years; we slice it into periods of 3 days each. In addition, we build an additional dataset (**Epi**) about historical epigraphy data Clauss et al., 2021. The dataset is made of 117.000 Latin inscriptions comprising one or several of 18 social statuses (slave, soldier, senator, etc.) and its location as one of 62 possible regions, along with an estimated dating spanning from 100BC to 400AD. The goal is to guess the region where a status has been found, with respect to time. The goal is to recover statuses diffusion in territories newly conquered by the Roman Empire. We slice this dataset in epochs of one year each.

Evaluation is again conducted using a 5-fold cross-validation with 80% of training data, 10% of validation data and 10% of testing data for each fold. For each pair (i, o_{true}) in the test set, we query the probability for every output o given i and build the confusion matrix by comparing them to o_{true}. In Table II.5, we present the results of our method compared to the proposed baselines for various metrics: Area under the ROC curve (**AUCROC**), Average Precision (**AP**) and Normalized Coverage Error (**NCE**). The first two metrics evaluate how well models assign probabilities to observations, and the latter evaluates the order in which possible outputs are ranked. Overall, we see that our method exhibits a greater predictive power, except for the Reddit dataset where the static SBM performs as well as SDSBM. We explain this by the lack of significant temporal variation over the considered interval. This could be expected, since the dataset comprises roughly 80% of repeated actions (Kumar, Zhang, and Leskovec, 2019), meaning that users do not significantly explore new communities over a month. This result shows that SDSBM also works well in the static case. On the other datasets, SDSBM performs better often by a large margin, especially for the AUCROC, meaning that SDSBM is efficient at distinguishing classes from each other. We recall that the model used here is deliberately simplistic for demonstration purposes; low metrics do not mean the Simple Dynamic prior does not work, but instead that it should be coupled to a more complex model.

As an illustration of what SDSBM has to offer, we plot in Fig. II.14 a possible visualization of the membership's evolution over time for the epigraphy dataset. On the left and the right, we show the items that are considered in the visualization. The time goes from left (100BC) to right (500AD), and the flows represent the membership transfers between epochs. The grey bars represent the clusters. We manually annotated them by looking at their composition –explicit clusters composition is given in Appendix, Section I.8. From this figure, we can recover several historical facts: military presence in Rome was scarce for most of the times considered; Italy concentrates less military presence as time goes (due to its spread over the now extended empire), until the 3rd-century crisis that led to its re-militarization; most of the slaves that have been accorded an inscription are located in Italia throughout time; the religious functions (Augusti) are evenly spread on the territory at all times; the libertii (freed slaves) inscriptions are essentially present in Rome and Italy, etc. Obviously, dedicated works are needed to support these illustrative claims, and we believe SDSBM can provide such extended comprehension of the processes at stake.

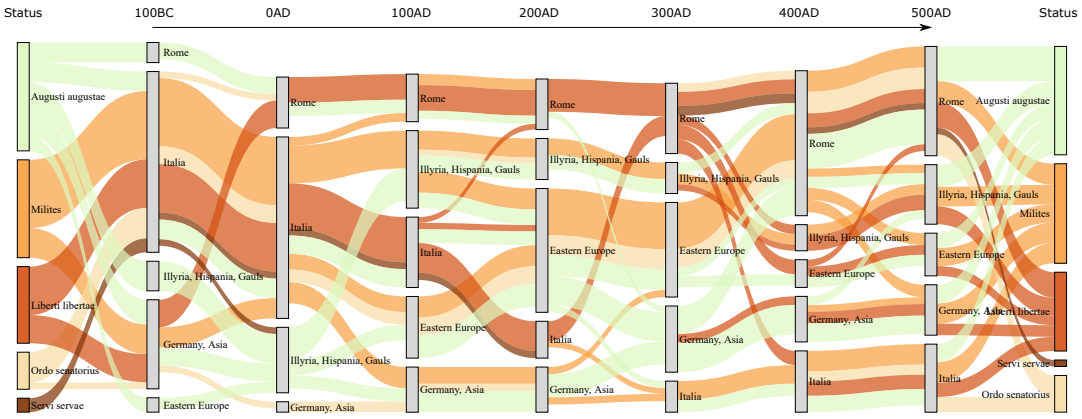

FIGURE II.14: **Geographic evolution of status distribution from Latin graves (100BC - 500AD)** — We applied the SDSBM to the Epigraphy dataset. We recall that our goal is to predict a roman region (e.g., Illyria, Hispania, etc.) given a status (e.g., Slave, Senator, etc.) and a year. We plot the temporal evolution of statuses membership to the five manually labelled clusters (in grey). For clarity, we removed small membership transfers from the data, which explains why the total cluster's population may vary from one time to another. This plot allows us to visualize some global historical trends about the evolution of the Roman Empire (e.g., 3rd-century crisis, the spread of military presence in Europe, Italy's demilitarization, etc.).

II.3.3.f Conclusion

We introduced a simple way to model time in dynamic valued and labelled networks by assuming a dynamic Mixed-Membership SBM. Our method defines the Simple Dynamic prior, ready to plug into any iteration of SIMSBM –see Section II.2.2.a. Time is considered under the single assumption that a network's ties do not vary abruptly over time.

We assessed the performance of the proposed method by defining the SDSBM and evaluating it in several controlled situations on synthetic datasets. In particular, we show that our prior shows stable performances with respect to the dataset slicing, and that it works well under challenging conditions (small amounts of data or high entropy blocks interaction matrix). We also evaluated SDSBM on large scale real-world datasets and showed how accounting for time increases yields better results than the two proposed baselines. Finally, we illustrate an application interest on a dataset of Latin inscriptions that indirectly narrates the evolution of the Roman Empire.

In the discussion section, we argued that our temporal prior offers great modelling flexibility: uneven slicing of observations over time, heterogeneous dynamic time scales for items (or clusters), time-dependent blocks-interaction matrix, informativeness of the prior, and temporal neighbours' dependence with respect to the averaging function. Future works exploring these directions on real-world data may help retrieve meaningful clusters for useful applications. On a further note, we believe that a key interest of our approach is the amount of data needed to get satisfactory predictive performances. As mentioned above, this point is fundamental to several social sciences, and we believe our approach could ease the incorporation of automated learning methods in these fields.

II.4 Conclusions

Global SIMSBM framework

In this section, we first developed a global framework, SIMSBM, that generalizes several models from the literature as particular cases, such as MMSBM, Bi-MMSBM, IMMSBM and T-MBM.

This results in a highly flexible model that can be applied to a broad range of problems. In particular, we cited throughout the text several experimental studies conducted in medicine, social behaviour and recommendation using special cases of our model.

Our framework answers the two challenges raised in Section II.2.1.b: it extends MMSBMs to any number of entity types that can have higher-order interactions.

Case study with SIMSBM(2)

We then proposed to study interactions in information spread by considering a special case of the SIMSBM: SIMSBM(2), or IMMSBM. We demonstrate that SIMSBM(2) allows us to investigate the detail of interactions strength in several datasets.

Our conclusions on interactions in information spread show that their effect is not trivial and that taking them into account increases predictive performance. However, these interactions are sparse: only 5% of significantly interacting pairs of clusters. In most cases, non-interacting models seem to provide an accurate enough description of output predictions – typically a spreading action.

Modelling dynamic memberships of interacting entities

The previous conclusions have been made by supposing the underlying interaction mechanisms to remain stable over one month. While this assumption holds on some datasets, most of the time there is a need to model their temporal evolution. In this perspective, we introduced a simple temporal prior, ready to plug into any of the SIMSBM iterations. The time is considered under the single assumption that the network's ties do not vary abruptly over time.

The performance of the proposed method is evaluated on several real-world datasets and shows that taking the time into account improves the results on each of them over the equivalent static model. On a further note, a key interest of this approach is the small amount of data needed to get satisfactory performances. This point is fundamental to several social sciences and extends beyond the scope of interactions modelling.

Interactions are sparse

The overall conclusion of this section is the following: **interactions are sparse**. Significant interactions take place only between a limited fraction of cluster pairs, and between an even smaller fraction of entity pairs. This underlines the necessity of considering clusters of entities to efficiently model such sparse interactions.

Time range of interactions?

However, all the models discussed consider static interactions. The dynamic version of SIMSBM proposed in Section II.3 allows us to consider static interactions whose expression evolves with time, but the interactions themselves are not dynamic. To

illustrate, a reasonable assumption is that the strength of interactions is not constant over time; a tweet does not have the same influence on a user if she saw it ten minutes or ten days earlier. The SDSBM approach accounts for the long-term changes in the evolution of the interacting mechanisms, but not for their immediate temporal evolution. A Twitter user might be influenced by tweets about sports at some point, and about politics at some other point, but being always influenced in the same way by either of them; the membership might change but not the interaction dynamics of each tweet. Modelling the temporal evolution of interaction strength using MMSBMs might be possible. However, other methods that consider time as a continuous variable instead of slicing it into episodes seem more relevant to the task. In the next section, we propose to investigate how entities' interaction strength evolves with time.

Chapter III

Temporal diffusion networks – Interactions are brief

Abstract

We saw in the previous chapter that despite being sparse, interactions between pieces of information (the *entities*) globally play a substantial role in individuals' actions: the adoption of a product, the spread of news, a strategy choice, etc. However, the sparsity of significant interactions could be due to temporal considerations; interactions may fade with time, making them harder to spot for static models.

Section III.1, we discuss and justify the role of time in interactions modelling.

Section III.2, we show that this aspect of the underlying interaction mechanisms remains unexplored in the literature.

Section III.3, we introduce an efficient method to infer both the entities' interaction network and its evolution according to the temporal distance separating interacting entities. We develop a convex model using multi-kernel inference, named InterRate. Here, time is considered as a continuous variable, unlike what we proposed in Section II.3. The temporal evolution of interaction intensity is what we call the *interaction profile*.

Section III.4, we consider a timestamped sequence of exposures to entities (e.g., URL, ads, situations) and the actions a user exerts on them (e.g., share, click, decision). We study how users exhibit different behaviours according to *combinations* of entities they have been exposed to in the past. We show that the joint effect of two exposures on a user is more than the disjoint sum of their individual effect –there is an interaction. InterRate allows for non-parametric convex optimization and can be solved in parallel.

Section III.5, we show our method recovers state-of-the-art conclusions on interaction processes on three real-world datasets. It outperforms the proposed baselines in the inference of the underlying data generation mechanisms. Finally, we show that interaction profiles can be visualized intuitively, making the interpretation of the model easier.

The overall conclusion of this section is that **interactions are brief**. In real-world diffusion processes, the interaction between two entities is significant only when the two entities are close to each other in time. For instance, a tweet does not have the same influence on the last tweet a user saw if the first appeared ten minutes or ten days earlier in her news feed. Typically, the intensity of an interaction decreases exponentially with the time separating the interacting entities. Our conclusion emphasizes the necessity of modelling the temporal aspect of interactions in social networks.

Published works: (Poux-Médard, Velcin, and Loudcher, 2021a)

III.1 Introduction

III.1.1 Temporal evolution of interactions

In Chapter II, we showed that interactions play a significant role in information spread. However, the models introduced there are all static: an interaction between several entities has a constant intensity over time. Many everyday examples contradict this assumption. The writing of a scientific article is more influenced by last year's publications than by last century's ones; a user on social media is more likely to answer a recent post than a years-old one; a Spotify user is more likely to listen to a band if she listened to it five minutes ago than if she listened to it five years ago. In general, interactions between entities last for a given time; eventually entities fade to a non-interacting "ground-state" as time goes forward. In Chapter II, we called this ground-state *virality*, which is the probability of an outcome in the absence of interactions.

The histograms presented in Fig.III.1 illustrate this assumption: the probability of an action on an entity (like, share, comment, etc.) varies according to the temporal distance separating any two given entities. We refer to such figures as *interaction profiles*. They represent an action's probability evolution given the time separating the interacting entities.

The study of this quantity is a novel perspective: interactions between pieces of information have been little explored in the literature, and no previous work unveils trends in the information interaction mechanisms.

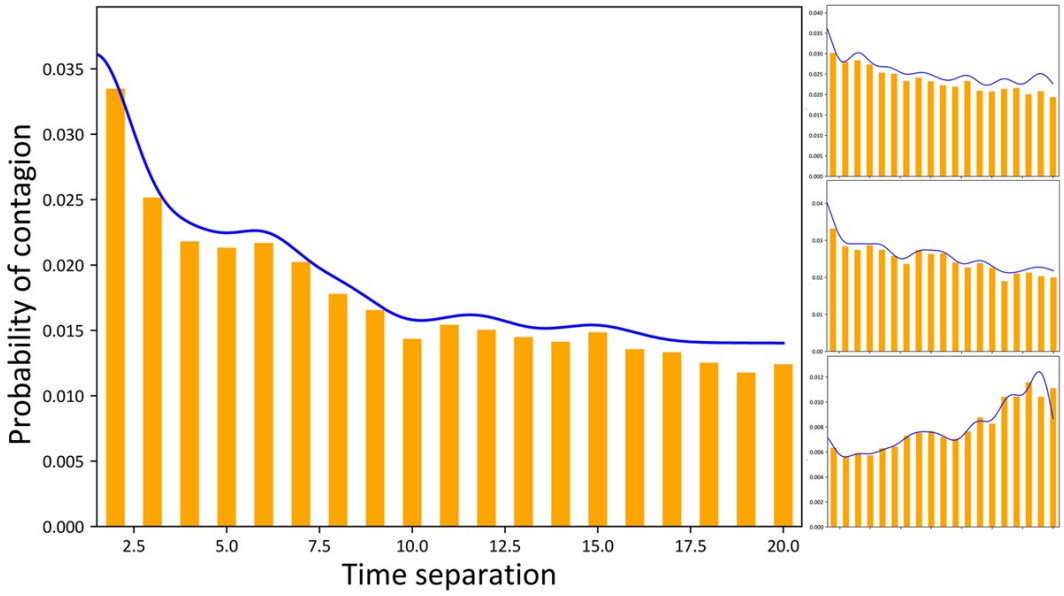

FIGURE III.1: **Interaction profiles between pairs of entities** — Examples of interaction profiles on Twitter; here is shown the effect of URL shortening services migre.me (left), bit.ly (right-top), tinyurl (right-middle) and t.co (right-bottom) on the probability of tweeting a t.co URL and its evolution over time. This interaction profile shows, for instance, that there is an increased probability of retweeting for a t.co URL when it appears shortly after a migre.me one (interaction). This increase fades when the time separation grows (no more interaction). In blue, the interaction profile inferred by our model.

The study of interactions between entities has several applications in real-world systems. We can mention the fields of recommender systems (the probability of adoption is influenced by what a user saw shortly before it), news propagation and

control (when to expose users to an entity to maximize its spreading probability) (Vosoughi, Roy, and Aral, 2018), advertising (same reasons as before) (Cao and Sun, 2019), choice behaviour (what influences a choice and how) (Cobo-López, Godoy-Lorite, and Duch, 2018).

III.1.2 Proposed approach

In this chapter, we propose to unveil the temporal mechanisms at stake within those interacting processes; we infer their interaction profiles. Imagine, for instance, that an internet user is exposed to a tweet at time t_1 and to another at time $t_2 > t_1$. We suppose that the exposure to the first one influences the user's sensitivity (likeliness of a retweet) to the second one a time $t_2 - t_1$ later. Modelling this process involves quantifying the influence a tweet exerts on the other and how this influence varies with the time separating the exposures. The representation of this probability evolution is what we call the *interaction profile* –illustrated Fig.III.1). It represents the influence the exposure to an entity exerts on an outcome (click, buy, retweet, etc.) for another exposure to an entity a given time later.

To perform this task, we introduce an efficient method to infer both the entities' interaction network and its evolution according to the temporal distance separating the interacting entities. In the proposed InterRate model, nodes are entities, and edges between them represent the intensity of their interaction; this intensity is a continuous-time function that depends on the time separating two exposures to the entities (or nodes). This intensity function is the interaction profile.

III.1.3 Workflow

First, we review the state of the art in temporal interaction between entities and underline the open challenges they rise in Section III.2. Then, we answer them with InterRate in Section III.3, a model for inferring all the interaction profiles between every node pair in a given set. This is performed in a continuous-time setup using multi-kernel inference methods (Du et al., 2012). We show in Section III.4 that the inference of the parameters boils down to a convex optimization problem for specific kernel families. Moreover, the problem can be subdivided into as many subproblems as entities, which can be solved in parallel. The convexity of the problem guarantees convergence to the likelihood's global optimum for each subproblem and, therefore, to the problem's optimal likelihood. In Section III.4.2, we use InterRate to investigate the role of interaction profiles on synthetic data and in various corpora from different fields of research: advertisement (the exposure to an ad influences the adoption of other ads (Cao and Sun, 2019)), social dilemmas (the previous actions of one influences an other's actions (Cobo-López, Godoy-Lorite, and Duch, 2018)) and information spread on Twitter (the last tweets read influence what a user retweets (Myers and Leskovec, 2012)). Finally, in Section III.5, we provide analysis leads and show that our method recovers state-of-the-art results on interaction processes on each of the three considered datasets.

III.1.4 Contributions

The main contributions developed in this chapter are the following:

- We introduce the interaction profile. It represents the evolution of interaction intensities as the time separating interacting entities increases. The interaction profile is a powerful tool to understand how interactions take place in a given

corpus (see Fig.III.4) and it has not been developed in the literature up to now. Its introduction is the main contribution of the present section.

- We design a convex non-parametric algorithm that can be solved in parallel, baptized InterRate. InterRate automatically infers the interaction profile for every node pair in a network. The collection of interaction profiles can be interpreted as a temporal interaction network.
- We show that InterRate yields better results than non-interacting or non-temporal baseline models on several real-world datasets. Furthermore, our model can recover several conclusions about the datasets from state-of-the-art works.
- We discuss the temporal aspect of entities interactions. Specifically, we show that interactions are brief in real-world diffusion processes. We also recover the interactions sparsity that has been observed in Chapter II.

III.2 State of the art on temporal interaction network inference

III.2.1 Temporal interactions in general

Previous efforts in investigating the role of interactions in information diffusion have shown their importance in the underlying spreading processes. Several works study the interaction of information with users' attention (Weng, Flammini, and al., 2012), closely linked to information overload concepts (Gomez-Rodriguez, Gummadi, and Schölkopf, 2014), but not the interaction between the pieces of information themselves. On the other hand, whereas most of the modelling of spreading processes is based on either no competition (Senanayake, O'Callaghan, and Ramos, 2016; Poux-Médard, Pastor-Satorras, and Castellano, 2020) or perfect competition (Prakash et al., 2012) assumption, intermediate competitions lead to a better description of their spread (Beutel et al., 2012) –with the example of Firefox and Chrome web browsers, whose respective popularity are correlated but not perfectly as in (Prakash et al., 2012).

III.2.2 Modelling interactions

A significant effort has been put in elaborating complex processes to *simulate* interaction on real-world networks (Prakash et al., 2012; Zhu, Gao, and Zhang, 2020). However, fewer works have been developed to tackle interaction in information spread from a machine learning point of view. The correlated cascade mode (Zarezade et al., 2017) infers an interacting spreading process's latent diffusion network. In this work, the interaction is modelled by a hyper-parameter β. It tunes the intensity of interactions according to an exponentially decaying kernel. In their conclusion, the authors formulate the open problem of learning several kernels and the interaction intensity parameter β, which we address in the present chapter.

To our knowledge, the attempt the closest to our task to model the interaction intensity parameter β is Clash of the contagions (Myers and Leskovec, 2012). It predicts retweets on Twitter based on tweets seen by a user. This model estimates the probability of retweeting a piece of information, given the last tweets a user has been exposed to, according to their relative position in the Twitter feed. The method suffers various flaws (scalability, non-convexity). It also defines interactions based on an arguable hypothesis made on the prior probability of a retweet (in the absence of

interactions) that makes its conclusions about interactions sloppy (see Section II.2.1). It is worth noting that in (Myers and Leskovec, 2012), the authors outline the problem of the inference of the interaction profile but do so without searching for continuous trends such as the one shown in Fig.III.1.

In Chapter II, we addressed the various flaws observed in (Myers and Leskovec, 2012) and suggested a more general approach to the estimation of the interaction intensity parameters. However, we neglected the temporal aspect of interactions. To take back the Twitter case study, it implies that in the case of a retweet at time t, a tweet appearing at $t_1 \ll t$ in the news feed has the same influence on the retweet as a tweet that appeared at $t_2 \approx t$; the interaction profile is constant over time.

III.2.3 Temporal network inference

For several years, temporal network inference has been a subject of interest. Significant advances have been made using survival theory modelling applied to partial observations of independent cascades of contagions (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011; Gomez-Rodriguez, Leskovec, and Schölkopf, 2013b). In this context, an infected node tries to contaminate every other node at a rate that is tuned by β. While this work is not directly linked to ours, it has been a strong influence on the interaction profile inference problem we develop here; the problems are different, but the methodology they introduce greatly helped building InterRate (development and convexity of the problem, analogy between interaction profile and hazard rate). Moreover, advances in network inference based on the same works propose a multi-kernel network inference method that we adapted to the problem we tackle here (Du et al., 2012). Inspired by these works, we develop a flexible approach that allows for the inference of the best interaction profile by using several candidate kernels.

III.3 InterRate – Interaction dynamics

This work has been published, see (Poux-Médard, Velcin, and Loudcher, 2021a)

III.3.1 Problem definition

We illustrate the process to model in Fig. III.2. It runs as follows: a user is first exposed to a piece of information at time t_0. The user then chooses whether to act on it at time $t_0 + t_s$ (an act can be a retweet, a buy, a booking, etc.); t_s can be interpreted as the "reaction time" of the user to the exposure, assumed constant. The user is then exposed to the next piece of information a time δt later, at $t_1 = t_0 + \delta t$ and decides whether to act on it a time t_s later, at $t_1 + t_s$, and so on. Here, δt is the time separating two consecutive exposures, and t_s is the reaction time, separating the exposure from the possible contagion. In the remaining of this chapter, we refer to the user's action on an exposure (tweet appearing in the feed, exposure to an ad, etc.) as a contagion (retweet or the tweet, click on an ad, etc.).

This choice of modelling comes with several hypotheses. **First**, the pieces of information a user is exposed to appear independently from each other. It is the main difference between our work and survival analysis literature: the pseudo-survival of an entity is conditioned by the random arrival of pieces of information. Therefore, users' actions cannot be modelled as a survival process. This assumption holds in our experiments on real-world datasets, where users have no influence on what information they are exposed to. **Second** hypothesis, the user is contaminated solely

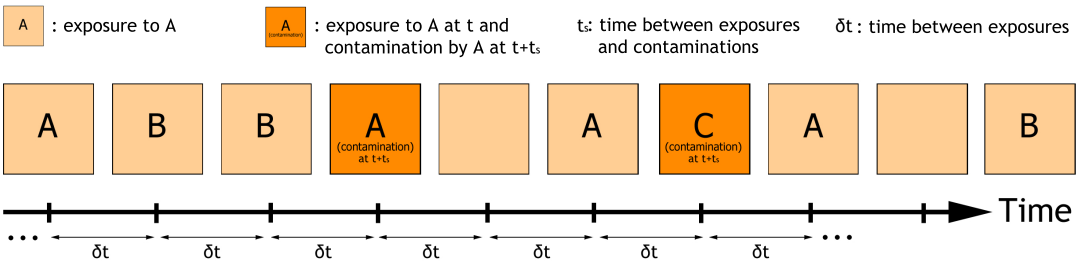

FIGURE III.2: **Illustration of the interacting process** — Light orange squares represent the exposures, dark orange squares represent the exposures that are followed by contagions and empty squares represent the exposures to the information we do not consider in the datasets (they only play a role in the distance between exposures when we consider the order of appearance as a time feature). A contagion occurs at a time t_s after the corresponding exposure. Each new exposure arrives at a time δt after the previous one. Contagion takes place with a probability conditioned by all previous exposures. In the example, the contagion by A at time $t + t_s$ depends on the effect of the exposure to A at times t and $t - 3\delta t$, and to B at times $t - \delta t$ and $t - 2\delta t$.

based on the previous exposures in the feed (Myers and Leskovec, 2012; Zarezade et al., 2017). **Third**, the reaction time separating the exposure to a piece of information from its possible contagion, t_s, is constant (i.e., the time between a read and a retweet in the case of Twitter). Importantly, this hypothesis is a deliberate simplification of the model for clarity purposes; relaxing this hypothesis is straightforward by extending the kernel family, which preserves convexity and time complexity. Note that this simplification does not always hold, as shown in recent works concluding that response time can have complex time-dependent mechanisms (Yu et al., 2017).

III.3.2 Likelihood

We now define the likelihood of the model whose process is described in Fig. III.2. Let $t_i^{(x)}$ be the exposure to x at time t_i, and $t_i^{(x)} + t_s$ the time of its possible contagion. Consider now the instantaneous probability of contagion (*hazard function*) $H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy})$, that is the probability that a user exposed to the piece of information x at time t_i is contaminated by x at $t_i + t_s$ given an exposure to y at time $t_j \leq t_i$. The matrix of parameters β_{ij} is what the model infers. β_{ij} is used to characterize the interaction profile between entities. We define the set of exposures preceding the exposure to x at time t_i (or history of $t_i^{(x)}$) as $\mathcal{H}_i^{(x)} := \{t_j^{(y)} \leq t_i^{(x)}\}_{j,y}$. Let $\mathcal{D}$ be the whole dataset such as $\mathcal{D} := \{(\mathcal{H}_i^{(x)}, t_i^{(x)}, c_{t_i}^{(x)})\}_{i,x}$. Here, c is a binary variable that account for the contagion ($c_{t_i}^{(x)} = 1$) or non-contagion ($c_{t_i}^{(x)} = 0$) of x at time $t_i + t_s$. The likelihood for one exposure in the sequence given $t_j^{(y)}$ is:

$$L(\beta_{xy}|\mathcal{D}, t_s) = P(\mathcal{D}|\beta_{xy}, t_s) =$$
$$\underbrace{H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy})^{c_{t_i}^{(x)}}}_{\text{contagion at } t_i^{(x)} + t_s \text{ due to } t_j^{(y)}} \cdot \underbrace{(1 - H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy}))^{(1 - c_{t_i}^{(x)})}}_{\text{Survival at } t_i^{(x)} + t_s \text{ due to } t_j^{(y)}}$$

The likelihood of a sequence (as defined in Fig.III.2) is then the product of the previous expression over all the exposures that happened before the contagion event $t_i^{(x)} + t_s$ e.g. for all $t_j^{(y)} \in \mathcal{H}_i^{(x)}$. Finally, the likelihood of the whole dataset $\mathcal{D}$ is the

product of $L(\beta_x|\mathcal{D}, t_s)$ over all the observed exposures $t_i^{(x)}$. Taking the logarithm of the resulting likelihood, we get the final log-likelihood to maximize:

$$\begin{aligned}\ell(\beta|\mathcal{D}, t_s) = \\ \sum_{\mathcal{D}} \sum_{t_j^{(y)} \in \mathcal{H}_i^{(x)}} c_{t_i}^{(x)} \log\left(H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy})\right) \\ + (1 - c_{t_j}^{(y)}) \log\left(1 - H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy})\right)\end{aligned} \tag{III.1}$$

III.3.3 Proof of convexity

The convexity of a problem guarantees to retrieve its optimal solution and allows using dedicated fast optimization algorithms.

Proposition 1. *The inference problem* $\min_\beta -\ell(\beta|\mathcal{D}, t_s)\ \forall \beta \geq 0$, *is convex in all of the entries of* β *for any hazard function that obeys the following conditions:*

$$\begin{cases} H'^2 \geq H'' H \\ H'^2 \geq -H''(1 - H) \\ H \in]0; 1[\end{cases} \tag{III.2}$$

where $'$ *and* $''$ *denote the first and second derivative with respect to* β, *and* H *is the shorthand notation for* $H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy})\ \forall i, j, x, y$.

Proof. The negative log-likelihood as defined in Eq.III.1 is a summation of $-\log H$ and $-\log(1 - H)$; therefore $H \in]0; 1[$. The second derivative of these expressions according to any entry β_{mn} (noted $''$) reads:

$$\begin{cases} (-\log H)'' = \left(\frac{-H'}{H}\right)' = \frac{H'^2 - H'' H}{H^2} \\ (-\log(1 - H))'' = \left(\frac{H'}{1-H}\right)' = \frac{H'^2 + H''(1-H)}{(1-H)^2} \end{cases} \tag{III.3}$$

The convexity according to a single variable holds when the second derivative is positive, which leads to Eq.III.2. The convexity of the problem then follows from the composition rules of convexity. □

Several functions obey the conditions of Eq.III.2, such as the exponential ($e^{-\beta t}$), Rayleigh ($e^{-\frac{\beta}{2} t^2}$), power-law ($e^{-\beta \log t}$) functions, and any log-linear combination of those (Du et al., 2012). These functions are standard in survival theory literature (Gomez-Rodriguez, Leskovec, and Schölkopf, 2013a).

The final convex problem can then be written $\min_{\beta \geq 0} -\ell(\beta|\mathcal{D}, t_s)$. An interesting feature of the proposed method is that the problem can be subdivided into N convex subproblems that can be solved independently (one for each piece of information). To solve the subproblem of the piece of information x, that is to find the vector β_x, we need to consider only the subset of $\mathcal{D}$ where x appears. Explicitly, each subproblem consists in maximizing Eq.III.1 over the set of observations $\mathcal{D}^{(x)} := \{(\mathcal{H}_i^{(x)}, t_i^{(x)}, c_{t_i}^{(x)})\}_i$.

III.4 Experiments

III.4.1 Experimental setup

III.4.1.a Kernel choice

Gaussian RBF kernel family (IR-RBF)

Based on (Du et al., 2012), we consider a log-linear combination of Gaussian radial basis function (RBF) kernels as hazard function. We also consider the time-independent kernel needed to infer the base probability of contagion discussed in the section "Background noise in the data" below. The resulting hazard function is then:

$$\log H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{ij}) = -\beta_{ij}^{(bg)} - \sum_{s=0}^{S} \frac{\beta_{ij}^{(s)}}{2} (t_i + t_s - t_j - s)^2$$

The parameters $\beta^{(s)}$ of Rayleigh kernels are the amplitude of a Gaussian distribution centered on time s. The parameter S represents the maximum time shift we consider. In our setup, we set S=20. We think it is reasonable to assume that an exposition does not significantly affect a possible contagion 20 steps later. The parameter $\beta_{ij}^{(bg)}$ corresponds to the time-independent kernel –base probability of contagion by i, or virality. The formulation allows the model to infer complex distributions from a reduced set of parameters whose interpretation is straightforward.

Exponentially decaying kernel (IR-EXP)

We also consider an exponentially decaying kernel. We consider the following form for the hazard function and refer to this modelling as IR-EXP:

$$\log H(t_i^{(x)} + t_s | t_j^{(y)}, \beta_{xy}) = -\beta_{ij}^{(bg)} - \beta_{ij}(t_i + t_s - t_j)$$

where $\beta_{ij}^{(bg)}$ once again accounts for the background noise in the data discussed further in this section.

III.4.1.b Parameters learning

Datasets are made of sequences of exposures and contagions, as shown in Fig.III.2. To assess the robustness of the proposed model, we apply a 5-fold cross-validation method. After shuffling the dataset, we use 80% of the sequences as a training set and the 20% left as a test set. We repeat this slicing five times, taking care that an interval cannot be part of the test set more than once. The optimization is made in parallel for each piece of information via the convex optimization module for Python CVXPY.

We also set the time separating two exposures δt as constant. It means that we consider only the order of arrival of exposures instead of their absolute arrival time. The hypothesis that the order of exposures matters more than the absolute exposure times has already been used with success in the literature (Myers and Leskovec, 2012). Besides, in some situations, the exact exposure time cannot be collected, while the exposures' order is known. For instance, in a Twitter corpus, we only know in what order a user reads her feed, unlike the exact time she read each of the posts.

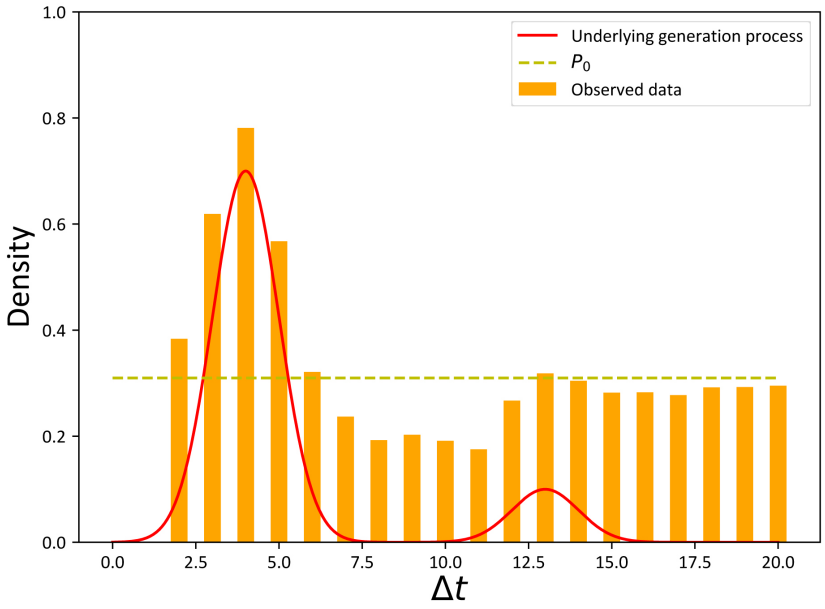

FIGURE III.3: **Underlying generation process vs observed data** — The red curve represents the underlying probability of contagion by C given an exposure observed Δt steps before C. The orange bars represent the observed probability of such events. We see that there is a noise $P_0(C)$ in the observed data. The underlying generation process can then only be observed in the dataset when its effect is larger than some threshold $P_0(C)$.

However, from its definition, our model works the same with non-integer and non-constant δt in datasets where absolute time matters more than the order of appearance.

III.4.1.c Background noise in the data

Because the dataset is built looking at all exposure-contagion correlations in a sequence, there is inherent noise in the resulting data. To illustrate this, we look at the illustrated example Fig.III.2 and consider the exposure to C leading to a contagion happening at time t. We assume that in the underlying interaction process, the contagion by C at time $t + t_s$ only took place because C appeared at time t. However, when building the dataset, the contagion by C is also attributed to A appearing at times $t - \delta t$, $t - 3\delta t$ and $t - 6\delta t$, and to B appearing at times $t - 4\delta t$ and $t - 5\delta t$. It induces noise in the data. In general, for any contagion in the dataset, several observations (pair exposure-contagion) come from the random presence of entities unrelated to this contagion.

We now illustrate how this problem introduces noise in the data. In Fig.III.3, we see that the actual underlying data generation process (probability of a contagion by C given an exposure present Δt step earlier) does not exactly fit the collected resulting data: the data gathering process induces a constant noise whose value is noted $P_0(C)$ –that is the average probability of contagion by C. Thus, the interaction effect can only be observed when its associated probability of contagion is larger than $P_0(C)$. Consequently, the performance improvement of a model that accounts for interactions may seem small compared with a baseline that only infers $P_0(C)$. That is what we observe in the experimental section. However, in this context, a small improvement in performance shows an extended comprehension of the underlying interacting processes at stake (see Fig.III.3, where the red line explains the data better

than a constant baseline). Our method efficiently infers $P_0(C)$ *via* a time-independent kernel function $\beta_{i,j}^{P_0(i)}$. Inferring $P_0(i)$ is in line with the discussion of Section II.2.1 on the necessity to infer the virality along with interaction terms.

III.4.1.d Evaluation criteria

The main difficulty in evaluating these models is that interactions might occur between a small number of entities only. It is the case here, where many pairs of entities have little to no interaction (see the Discussion section). This makes it difficult to evaluate how good a model is at capturing them. To this end, our principal metric is the residual sum of squares (**RSS**). The RSS is the sum of the squared difference between the observed and the expected frequency of an outcome. This metric is particularly relevant in our case, where interactions may occur between a small number of entities: any deviation from the observed frequency of contagion is accounted for, which is what we aim at predicting here. We also consider the Jensen-Shannon (**JS**) divergence; the JS divergence is a symmetric version of the Kullback–Leibler divergence, which makes it usable as a metric (Nielsen, 2020).

We finally consider the best-case F1-score (**BCF1**) of the models, that is, the F1-score of the best scenario of evaluation. It is not the standard F1 metric (that poorly distinguishes the models since few interactions occur), although its computation is similar. Explicitly, it generalizes F1-score for comparing probabilities instead of comparing classifications; the closer to 1, the closer the inferred and observed probabilities. It is derived from the best-case confusion matrix, whose building process is as follows: we consider the set of every information that appeared before information i at time t_i in the interval, that we denote $\mathcal{H}_i$. We then compute the contagion probability of i at time $t_i + t_s$ to every exposure event $t_j^{(y)} \in \mathcal{H}_i$. Confronting this probability with the observed frequency f of contagions of i at time $t_i + t_s$ given $t_j^{(y)}$ among N observations, we can build the best-case confusion matrix. In the best-case scenario, if out of N observations the observed frequency is f and the predicted frequency is p, the number of True Positives is $N \times \min\{p, f\}$, the number of False Positives is $N \times \min\{p - f, 0\}$, the number of True Negatives is $N \times \min\{1 - p, 1 - f\}$, the number of False Positives is $N \times \min\{f - p, 0\}$.

Finally, when synthetic data is considered, we also compute the mean squared error of the β matrix inferred according to the β matrix used to generate the observations, that we note **MSE** β.

We purposely ignore evaluation in prediction because, as we show later, interactions influence quickly fades over time: probabilities of contagion at large times are mainly governed by the background noise discussed in previous sections. Therefore, it would be irrelevant to evaluate our approach's predictive power on the whole range of times where it does not bring any improvement over a naive baseline (see Fig.III.1). A way to alleviate this problem would be to make predictions only when interaction effects are above/below a certain threshold (at short times, for instance). However, such an evaluation process would be debatable. Here, we choose to focus on the descriptive aspect of InterRate.

III.4.1.e Baselines

Naive baseline

For a given piece of information i, the contagion probability is defined as the number of times a user acts on it divided by the number of its occurrences.

TABLE III.1: **Experimental results on synthetic data** — Our model outperforms all of the baselines in almost every dataset for every evaluation metric. The standard deviations of the 5 folds cross-validation are negligible. The models presented in this chapter are underlined.

		RSS	JS div.	BCF1	MSE β
Synth-20	IR-RBF	**18.42**	**0.0023**	**0.9188**	**0.0005**
	ICIR	139.59	0.0100	0.8270	0.0159
	Naive	145.51	0.0104	0.8221	
	CoC	123.06	0.0094	0.8220	
	IMMSBM	222.06	0.0173	0.7265	
Synth-5	IR-RBF	**0.12**	**0.0002**	0.9742	**0.0053**
	ICIR	8.27	0.0081	0.8499	0.0192
	Naive	10.03	0.0100	0.8214	
	CoC	**0.12**	**0.0002**	**0.9763**	
	IMMSBM	11.69	0.0136	0.7693	

Clash of the contagions

We use the work presented in (Myers and Leskovec, 2012) as a baseline. In this work, the authors model the probability of a retweet given the presence of a tweet in a user's feed. This model does not look for trends in the way interactions take place (it does not infer an interaction profile), considers discrete time steps (while our model works in a continuous-time framework), and is optimized via a non-convex SGD algorithm (which does not guarantee convergence towards the optimal model). More details on implementation are provided in Appendix, Section II.1.

IMMSBM

The Interactive Mixed-Membership Stochastic Block Model (IMMSBM) is a model that takes interactions between pieces of information into account to compute the probability of a (non-)contagion –see Section II.2.3.a. Note that this baseline does not take the position of the interacting pieces of information into account (time-independent) and assumes that interactions are symmetric (the effect of A on B is the same as B on A).

ICIR

The Independent Cascade InterRate (ICIR) is a reduction of our main IR-RBF model to the case where interactions are not considered. We consider the same dataset, enforcing the constraint that off-diagonal terms of β are null. The (non-)contagion of a piece of information i is then determined solely by the previous exposures to i itself.

III.4.2 Results

III.4.2.a Synthetic data

Data generation

We generate synthetic data according to the process described in Fig.III.2 for a given β matrix using the RBF kernel family. First, we generate a random matrix β, whose entries are between 0 and 1. A piece of information is then drawn with uniform

probability and can result in a contagion according to β, the RBF kernel family and its history. We simulate the outcome by drawing a random number and finally increment the clock. The process then keeps on by randomly drawing a new exposure and adding it to the sequence. We set the maximum length of intervals to 50 steps and generate datasets of 20,000 sequences.

Numerical results

We present in Tab. III.1 the results of the various models with generated interactions between 20 (Synth-20) and 5 (Synth-5) entities. The interactions are generated using the RBF kernel, hence the fact we are not evaluating the IR-EXP model –its use would be irrelevant. The InterRate model outperforms the proposed baselines for every metric considered. It is worth noting that performances of non-interacting and/or non-temporal baselines are good on the JS divergence and F1-score metrics due to the constant background noise P_0. For cases where interactions do not play a significant role, IMMSBM and Naive models perform well by fitting only the background noise. By contrast, the RSS metric distinguishes very well the models that are better at modelling interactions.

Note that while the baseline (Myers and Leskovec, 2012) yields good results when few interactions are simulated (Synth-5), it performs as bad as the naive baseline when this number increases (Synth-20). This is due to the non-convexity of the proposed model, which struggles to reach a global maximum of the likelihood even after 100 runs (see Appendix, Section II.1 for implementation details).

III.4.2.b Real data

We consider 3 real-world datasets. For each dataset, we select a subset of entities that are likely to interact with each other. For instance, it has been shown that the interaction between the various URL shortening services on Twitter is non-trivial (Zarezade et al., 2017).

We provide details on the way datasets have been built from raw data. For each of the real-world datasets, we choose to consider only the order of the various entities' apparition instead of their absolute appearance times. It implies setting the time separating two successive exposures as constant, that we note δt. This choice is supported by state-of-the-art works (Myers and Leskovec, 2012), and we observed in our experiments that it is more relevant than considering absolute times. Besides, we do not consider the first 10 pieces of information of any sequence to avoid boundary effects (the first 5 steps for the PD dataset): the history of exposures is incomplete in this case and could lead to biased results. For each dataset entities list, the number before the entity name is the key used in Fig. III.4. The entities subsets have been chosen by computing the co-occurrence matrix of all the entities and then selecting the ones that are part of a cluster using a K-means algorithm. The datasets are:

Datasets

- **Twitter** dataset (Hodas and Lerman, 2014): a collection of all the tweets containing URLs that have been posted on Twitter during October 2010, with the associated followers' network. A tweet read by a user in her feed is an exposition, and its possible retweet is a contagion. We consider only the URLs associated with the following URL shortening websites, the same as in (Zarezade et al., 2017): {0: migre.me, 1: bit.ly, 2: tinyurl, 3: t.co}. The final dataset is made

of 104,349 sequences of average length 53.5 steps (1 step = t_s), for 1,276,670,965 observed interactions.

- Prisoner's dilemma dataset (**PD**): contains ordered sequences of repeated Prisoner's dilemma games between two players. From the dataset introduced in (Nay and Vorobeychik, 2016), we consider the sub-dataset noted BR-risk 0 (first entry of Tab.2 in the reference); we choose this subset to have decisions made in a homogeneous context, where players struggle in a dilemma that is hard to solve (which depends on the combination of the parameters T, R, S and P discussed further). Within each round, the two players can defect or cooperate. Each duel is made of 10 rounds. If both cooperate, the reward R is high; if both defect, the reward P is low; if one player cooperates while the other defects, this one gets a penalty S, while the other gets a reward T. To make the game a Prisoner dilemma, the variables have to obey T>R>P>S. We refer to the combination of players' actions ("the user cooperated, and the opponent defected at time t") as exposures and to the *defect* actions of the player in the following round as a contagion. We defined the action of cooperating as a non-contagion. We therefore have 4 possible situations ({0: Player cooperated, and opponent defected, 1: Both players defected, 2: Both players cooperated, 3: Player defected and opponent cooperated}) and 2 possible outcomes (Player cooperates or defects). The final dataset is made of 2,337 sequences of average length of 10.0 steps, for 189,297 observed interactions.

- Taobao dataset (**Ads**): contains all ads exposures for 1,140,000 randomly sampled users from the website of Taobao for 8 days (5/6/2017-5/13/2017) (Cao and Sun, 2019). Taobao is one of the largest e-commerce websites and is owned by Alibaba. Each exposure is associated with the corresponding timestamp and user's action (click on the ad or not). A click is considered a contagion. The subset of ads we consider is: {0: 4520, 1: 4280, 2: 1665, 3: 4282}. The resulting dataset is made of 87,500 sequences of average length of 23.9 steps, for 240,932,401 observed interactions.

Numerical results

The results on real-world datasets are presented in Tab.III.2. We see that the IMMSBM baseline performs poorly on the PD dataset: either considering the time plays a consequent role in the probability of contagion, or interactions are not symmetric. Indeed, as we saw in the previous chapter, a core hypothesis of the IMMSBM is that the effect of exposition A on B is the same as B on A, whichever is the time separation between them. In a prisoner's dilemma game setting, for instance, we expect that a player does not react in the same way to defection followed by cooperation as to cooperation followed by defection, a situation for which the IMMSBM does not account. When there are few entities, the CoC baseline performs as good as IR, but it fails when this number increases; this is mainly due to the non-convexity of the problem that does not guarantee convergence towards the optimal solution. Overall, the InterRate models yield the best results on every dataset.

TABLE III.2: **Experimental results on real-world data** — Our model outperforms all of the baselines in almost every dataset for every evaluation metric. The standard deviations of the 5 folds cross-validation are negligible. The models presented in this chapter are underlined.

		RSS	JS div.	BCF1
Twitter	IR-RBF	**0.001**	0.000 06	0.9832
	IR-EXP	**0.001**	**0.000 05**	**0.9862**
	ICIR	0.014	0.000 63	0.9614
	Naive	0.016	0.000 73	0.9379
	CoC	0.002	0.000 07	0.9572
	IMMSBM	0.015	0.000 68	0.9543
PD	IR-RBF	**1.127**	**0.007 58**	**0.9789**
	IR-EXP	1.553	0.008 67	0.9661
	ICIR	3.536	0.018 23	0.9381
	Naive	3.653	0.019 15	0.9455
	CoC	1.241	0.008 09	0.9736
	IMMSBM	20.377	0.087 01	0.7672
Ads	IR-RBF	0.004	0.000 04	0.9814
	IR-EXP	**0.003**	**0.000 03**	**0.9852**
	ICIR	0.098	0.000 85	0.9659
	Naive	0.145	0.001 26	0.9126
	CoC	0.005	0.000 05	0.9741
	IMMSBM	0.015	0.000 15	0.9543

III.5 Discussion

III.5.1 Exponential interaction profiles

In Fig.III.4, we represent the interaction intensity over time for every pair of information considered in every corpus fitted with the RBF kernel model. The intensity of the interactions is the inferred probability of contagion minus the base contagion probability in any context: $P_{ij}(t) - P_0(i)$. We recall that $P_0(i)$ is inferred along with the other parameters; it is in line with the discussion of Section II.2.1 on the necessity to infer the virality along with interaction terms. Therefore, we can determine the characteristic range of interactions, investigate recurrent patterns in interactions, whether the interaction effect is positive or negative, etc.

Overall, we understand why the EXP kernel performs as good as the RBF on the Twitter and Ads datasets: interactions tend to have an exponentially decaying influence over time. However, this is not the case in the PD dataset: the effect of a given interaction is very dependent on its position in the history (pike on influence at $\Delta t = 3$, shift from positive to negative influence, etc.).

III.5.2 Recovering state of the art conclusions

In the Twitter dataset, the most substantial positive interactions occur before Δt=3. This finding agrees with previous works, which stated that the most informative interactions within the Twitter URL dataset occur within the 3 time steps before the possible retweet (Myers and Leskovec, 2012). We also find that the vast majority of interactions are weak, matching with previous study's findings see Chapter II and (Myers and Leskovec, 2012). However, it seems that tweets still exert influence even a long time after being seen, but with lesser intensity.

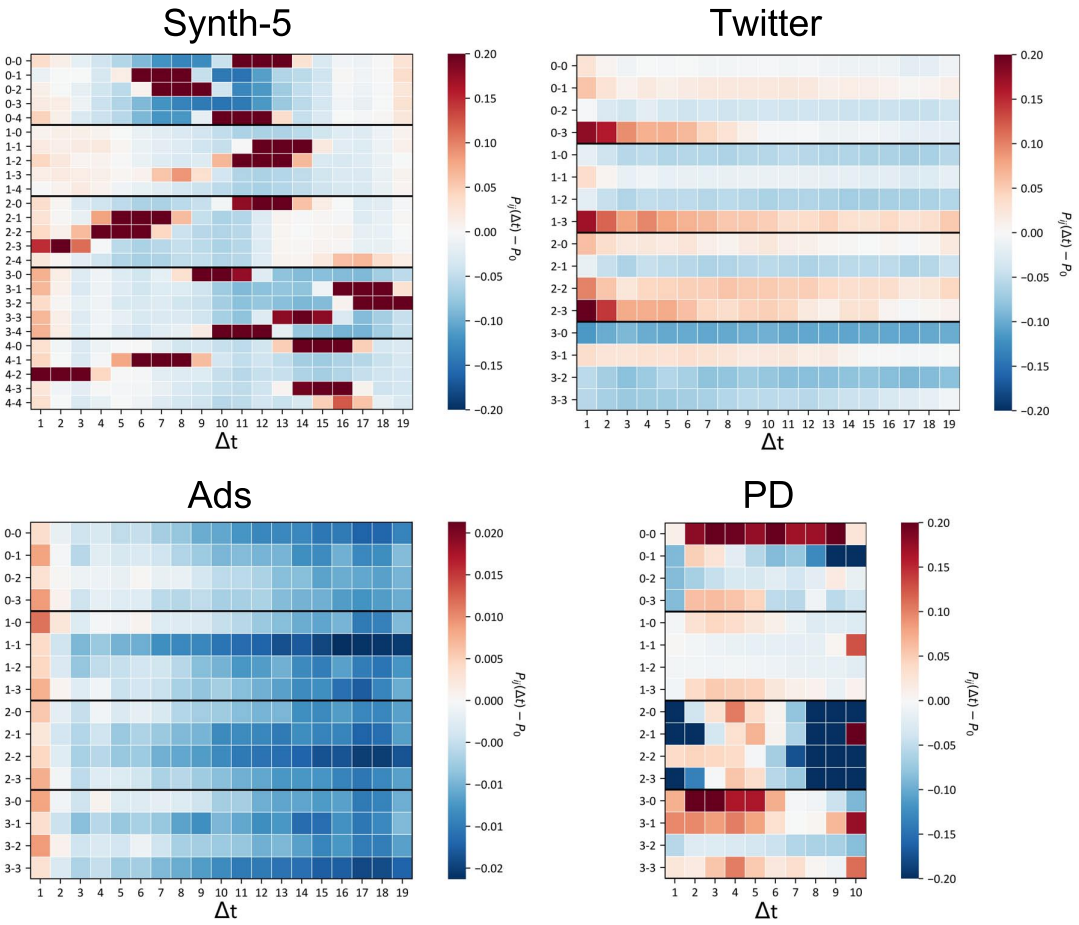

FIGURE III.4: **Visualization of the interaction profiles** — Intensity of the interactions between every pair of entities according to their time separation (one line is one pair's interaction profile, similar to Fig.III.1 seen "from the top"). A positive intensity means that the interaction helps the contagion, while a negative intensity means it blocks it. Note that we represented discontinuous times for visualization proposes, but the inferred kernel is continuous, as in Fig. III.1. The key linking numbers on the y-axis to names for each dataset is provided in the main text, Section III.4.2.b.1.

In the Prisoner's Dilemma dataset, players' behaviours are heavily influenced by the previous situations they have been exposed to. For instance, in the situation where both players cooperated in the previous round (pairs 2-x, 3^{rd} section in Fig.III.4-PD). The probability that the player defects is then significantly increased if both players cooperated or if one betrayed the other exactly two rounds before but decreased if it has been two rounds that players both cooperate.

Finally, we find that the interactions play a lesser role in the clicks on ads. We observe a slightly increased probability of clicking on every ad after direct exposure to another one. We also observe a globally decreasing probability of click when two exposures are distant in time, which agrees with previous work's findings (Cao and Sun, 2019). Finally, the interaction profile is very similar for every pair of ads; we interpret this as a similarity in users' ads perception.

We showed that for each of the considered corpus, considering the interaction profile provides an extended comprehension of choice adoption mechanisms and retrieves several state-of-the-art conclusions. The proposed graphical visualization also provides an intuitive view of how the interaction occurs between entities and the associated trends, hence supporting its relevance as a new tool for researchers in

a broad meaning.

III.6 Conclusions

Modelling the temporal aspect of interactions

We showed in previous works that interactions between entities play a substantial role in individuals' actions. However, the temporal aspect of interactions has been little explored in the literature, despite their importance in modelling real-world processes (Myers and Leskovec, 2012; Zarezade et al., 2017). In this chapter, we filled the gap and introduced an efficient convex model that investigates the temporal aspect of interactions.

Unlike previous models, our method accounts for both the interaction effects and their influence over time (the interaction profile). We showed InterRate yields better results on synthetic and real-world datasets. Therefore, taking the temporal aspect of interactions provides a finer comprehension of the processes at stake.

However, the method introduced in this section presents a major flaw. As it has been stated in the previous chapter, most entities do not interact with each other – interactions are rare. Therefore, in this chapter, we restricted to the study of small subsets of nodes which we knew were strongly interacting together. Our method shows that time is of importance in the modelling of interactions in spreading processes, but it cannot generalize these interaction patterns to a class of entities; it lacks clustering. It makes the results less interpretable, as the model does not look for a trend in individual entities' behaviour, in which we are interested.

Recovering conclusions on temporal interactions

We showed that InterRate manages to recover several state-of-the-art conclusions that have been made on the same datasets. On Twitter, most significant interactions happen at small times, and the majority of interactions are weak. On the advertisement corpus, our model highlighted the effects of marketing fatigue, which is a short increase in the probability of the click on an ad immediately after a first exposition, and a decreasing probability when expositions get more distant. On the prisoner's dilemma dataset, we showed our model recovers clues to a deterministic circular reasoning between players.

Interactions are short

The overall conclusion of this section is that **interactions in spreading processes are brief**. Typically, the intensity of an interaction on Twitter decreases exponentially with the time separating the interacting entities.

Towards proper interactions modelling

All the models proposed in both Chapter II and Chapter III make strong assumptions about the type of entities considered. Typically, an entity has been identified by a link on Twitter, independently from its *content*. However, two different links may be about the same topic and thus be regarded identically by other entities they interact with. It seems reasonable to assume that entities carrying identical semantic meanings are regarded in the same way by other entities they interact with.

As such, a more refined model should comprise three crucial elements to interaction modelling. **Entities should be given a more complete definition that includes their semantic meaning** rather than only their identifier. These **entities must be clustered together** according to their dynamics to be able to recover significant interactions. **Interactions should be dynamic**, meaning their value should depend on the time separating the interacting entities. In the next chapter, we derive such refined model that answers all these challenges.

Chapter IV

Dirichlet-Hawkes Processes - Modelling rare and brief interactions

Abstract

The conclusions drawn from Chapter II and Chapter III are that interactions are sparse (we need clusters to model them) and that interactions are brief (we need to consider time). Besides, we discussed the fact that interacting entities may have a short lifespan, and that their semantic content must be taken into account. In this last chapter, we present the steps paving our way to a single model that answers all of these challenges.

Section IV.1, we frame our approach as an in-depth modification of an existing Bayesian prior, the Dirichlet-Hawkes Process.

Section IV.2, we detail how to get this model by merging Dirichlet Processes with Hawkes processes, and highlight its main limits.

Section IV.3, as an indirect way to overcome these limits, we explore alternative forms of Dirichlet Processes and end up with an expression that alleviates a major feature of the Dirichlet Processes, their "rich-get-richer" property. We show that such a hypothesis is not always a relevant modelling choice. We propose the Powered Dirichlet Process as a way to directly control the importance of the "rich-get-richer" assumption.

Section IV.4, we then incorporate the newly proposed Powered Dirichlet Process into the standard Dirichlet-Hawkes process to create the Powered Dirichlet-Hawkes Process. We show our formulation yields significantly better results than state-of-the-art models when temporal information or textual content is weakly informative and alleviates the hypothesis that textual content and temporal dynamics are always perfectly correlated. Our approach eventually allows us to correctly model self-interactions within a cluster.

Section IV.5, we extend our approach to the multivariate case, which allows us to explore not only self-interactions, but interactions between all clusters. We present the challenges that arise from such an extension and how we overcome them.

Section IV.6, finally, we perform large-scale experiments on a real-world Reddit dataset, made of all of the headlines published on news subreddits throughout the whole year 2019. We confirm the findings of previous chapters and conclude that interactions play a minor role in this particular dataset.

Published works: (Poux-Médard, Velcin, and Loudcher, 2021c) in Section IV.4

IV.1 Introduction

IV.1.1 How to properly model interactions

In the two previous chapters, we underlined several challenges that arise when modelling interactions between entities in information spread. In Chapter II, we showed that interactions between pairs of entities are sparse –few entities interact significantly. In Chapter III, we showed that pair-interaction between entities are brief –entities have to be close in time for significant interactions to happen.

These conclusions invite us to rethink the definition of what we consider an entity. In previous chapters, an entity is identified using a unique ID. It can be a link, a word, or a song, but it take multiple forms. This is a strong assumption since several of such entities can carry an identical semantic meaning. Let link A[1] and link B[2] point to the same content. The previous definition of entities would consider them as distinct despite their semantic meaning being strictly identical. This could become problematic in contexts where such entities can express the same thing in several manners. Typically on Twitter or Reddit, opinions on recent news are expressed in various forms but may express the same thing. Another advantage of a more complete definition is the case of repeated information. Imagine a Twitter account that provides a daily weather forecast; each of the reports would be considered as a different entity whereas users are expected to react in similar ways to this or that forecast (sunny, rainy, cloudy, etc.), but not to every one of them.

Therefore, we need to account for entities' *content*. This is even more crucial in situations where pieces of information appear and disappear at a high rate, such as on online media platforms. The half-life of a Tweet is around 18 minutes, around 30 minutes for a Facebook post, 19h on Instagram, 6 days on Youtube, etc., which are all short in some perspective. Short lifespans provide less information on which to learn the interacting processes. However, aggregating these pieces of information together using both their content and dynamics would provide enough data to study interactions.

To summarize, we need to develop a model that creates clusters of entities based on both their content and temporal interactions.

IV.1.2 Objective

Our final goal is to model the temporal interaction between entities in real-world large-scale datasets. Nowadays, online information is generated at an unprecedented rate. At the time of writing, every minute, 500,000 comments are posted on Facebook, 400 hours of videos are uploaded on Youtube, and 500,000 tweets are published on Twitter.

As we stated in Chapter II, we need to cluster this mass of entities together to make sense of this mass of information. As we saw in the introduction, Section IV.1.1, this clustering must take entities' content into account –typically their textual content. Many clustering algorithms are based on text similarity, that is, how similar the words of two published documents are (Blei, Ng, and Jordan, 2003; Bahdanau, Cho, and Bengio, 2015; Rathore, Gupta, and Bhandari, 2018).

However, in Chapter III, we saw that these clusters must also include a temporal dimension to reflect the interacting processes at stake. Another variable to account for is thus the time of publication (Blei and Lafferty, 2006; Du et al., 2012).

[1] https://www.youtube.com/watch?v=dQw4w9WgXcQ
[2] https://www.youtube.com/watch?v=oHg5SJYRHA0

IV.1.3 Proposed approach

Many clustering models that claim to model dynamic clusters do not in fact explicitly account for time. At a given time t, they sample a subset of recent observations according to a temporal sampling function and then learn a static model for this time-step using the selected data only (Blei and Lafferty, 2006; Ahmed and Xing, 2008; Yin et al., 2018). However, sampling observations over time implies defining a sampling function that might not correctly model the temporal dynamics at stake. In general, time is not used by the model but instead only reduces the data provided to static models. It has been argued that such modelling is not fit to account for the arrival of documents in continuous-time settings (Du et al., 2015).

In (Du et al., 2015), the authors combine techniques of standard textual clustering with temporal point processes. The idea is to infer the time-sampling functions for data selection jointly with the clustering model with which they are associated. They derive the Dirichlet-Hawkes process (DHP) prior for clustering document streams by using jointly textual and temporal information in the cluster inference. In this model, clusters are self-stimulated. It means that each entity they comprise gets associated with a temporal function representing the probability of a new entity appearing at all times. This can be interpreted as the diagonal of the interaction matrix in Fig. II.8, but in its temporal version –with each case being associated with an interaction profile, see Section III.5.

This model (Du et al., 2015) seems to fit our task very well. However, we cannot use it as such, as it suffers from some limitations and assumptions. For instance, it has been argued that this method cannot handle limit cases where text is less informative –e.g., short texts, overlapping vocabularies (Yin et al., 2018).

IV.1.4 Workflow

In this chapter, we will first detail the Dirichlet-Hawkes Process (DHP) prior introduced in (Du et al., 2015) and point out its limits (Section IV.2). In particular, we will show that their inclusion of the temporal dimension results from an arbitrary choice, and that *tuning* the influence of time allows us to recover better results on challenging datasets.

To answer this problem, we will first question the Dirichlet Process (DP) on which DHP is built (Section IV.3). We demonstrate that alternative, more flexible variations of DP are possible and that they allow for better modelling performances; we call this alternative process the Powered Dirichlet Process (PDP).

We then develop the Powered Dirichlet-Hawkes Process (PDHP) as the reformulation of the DHP model in terms of PDP (Section IV.4). We show this novel, more flexible formulation yields significantly better results than the original DHP and alleviate several hypotheses the original model made.

Finally, in Section IV.5, we extend the PDHP to the multivariate case (MPDHP): we allow clusters to have temporal interactions with each other, and not only with themselves. This is equivalent to a temporal version of every matrix presented in Page 41-Fig. II.8 –and not only their diagonal as for DHP.

The final form of the proposed approach is able to model interactions that are:

- sparse –by clustering entities together (Chapter II)
- temporal –by associating each cluster an interaction profile, (Chapter III)
- content sensitive –by considering documents instead of entities identifiers

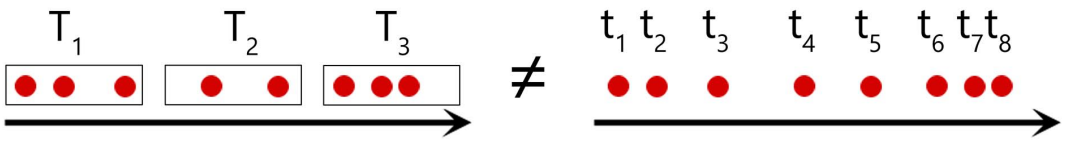

FIGURE IV.1: **Slicing the data to consider time can introduce bias** – Here, events in a same time slice can be further in time than observations in different ones. Explicitly modelling time using temporal processes helps getting rid of this bias.

We finally conduct a large-scale study of the multivariate temporal interactions between clusters of documents (entities with a content) in Section IV.6. We use 12 months of Reddit data specific to news subreddits, and conclude that the impact of interactions is small in this specific dataset. Reassuringly, we also confirm the findings of previous chapters on the sparsity and persistence of interactions in social media.

IV.2 State of the art and limits

IV.2.1 A brief overview of temporal clustering of textual documents

The use of temporal dimension in document clustering has been studied on many occasions; a notable spike of interest happened in 2006. Many authors tackled the problem of inferring time-dependent clusters from models based on LDA (Blei and Lafferty, 2006; Wang and McCallum, 2006; Iwata et al., 2009). However, most of these models are parametric, meaning the number of clusters is fixed at the beginning of the algorithm. Depending on the considered time range and the dataset, the number of clusters needs to be fine-tuned with several independent runs, making them hardly usable for many real-world applications. In all three references cited, the authors mention that a non-parametric version of the model might be derivable.

In 2008, A. Ahmed *et al.* proposed the Recurrent Chinese Restaurant Process (RCRP) as an answer to this problem (Ahmed and Xing, 2008). Instead of considering a fixed-size dataset, this model can handle a stream of documents arriving in chronological order, and the number of clusters is automatically updated. In this model, time is split into episodes to capture the temporal aspect of cluster formation; it considers an integer count of publications within a given time window. A later version of the model from 2010, the Distance-Dependent Chinese Restaurant Process (DD-CRP), tries to alleviate this approximation by replacing fixed-time episodes with a continuous-time sampling function (Blei and Frazier, 2010). However, the model still considers integer counts with only their distribution over time changing. The temporal dimension is not explicitly modelled, but instead used as a filter for the data fed to the model. Such slicing (even based on continuous sampling functions) can induce strong bias in the temporal modelling. One of these biases is illustrated in Fig. IV.1, where observations in the same time slice can be further in time than observations in different ones. Thus, the model is not designed to consider every temporal information in a continuous-time setting.

In 2015, N. Du *et al.* answered this problem by combining the Dirichlet process with the Hawkes process, used to model the appearance of events in a continuous-time setting. The key idea is to replace the counts of a Dirichlet process with the intensity function of the Hawkes process. The resulting Dirichlet-Hawkes process

(DHP) is then used as a prior for clustering documents appearing in a continuous-time stream. The inference is realized with a Sequential Monte-Carlo (SMC) algorithm. Following DHP, two articles have been published extending the idea: the Hierarchical Dirichlet-Hawkes process (Mavroforakis, Valera, and Gomez-Rodriguez, 2017) and the Indian Buffet Hawkes process (Tan, Rao, and Neville, 2018). Another work proposed an EM algorithm for the inference (Xu and Zha, 2017) (it uses a heuristic method to update the number of clusters and cannot handle a stream of documents).

IV.2.2 Dirichlet-Hawkes Process

IV.2.2.a Dirichlet Process

The Dirichlet Process is typically used in clustering models. It naturally yields a partition over a possibly infinite number of clusters. It is used as a prior on entities' membership –such as the ones introduced in Section II.2.1.b.

A well-known metaphor for the Dirichlet process is referred to as "Chinese restaurant". The corresponding process is named "Chinese Restaurant Process" (CRP). It can be illustrated as follows: when a n^{th} client arrives in a Chinese restaurant, she will sit at one of the K already occupied tables with a probability proportional to the number of persons already sitting at this table. She can also go to a new table in the restaurant and be the first client to sit there with a probability inversely proportional to the total number of clients already sitting at other tables. It can be written formally as:

$$CRP(C_i = c|\alpha, C_1, C_2, ..., C_{i-1}) = \begin{cases} \frac{N_c}{\alpha+N} \text{ if c = 1, 2, ..., K} \\ \frac{\alpha}{\alpha+N} \text{ if c = K+1} \end{cases} \quad \text{(IV.1)}$$

where c is the cluster chosen by the n^{th} customer, N_k is the population of cluster k, K is the number of already occupied tables and $\alpha \in \mathcal{R}^+$ the concentration parameter. When the number of clients goes to infinity, this process is equivalent to a draw from a Dirichlet distribution over an infinite number of clusters with a uniform concentration parameter α. The form of Eq.IV.1 is helpful to understand the underlying dynamics of the process and the contribution of seminal works we will detail now. It can be shown that the expected number of clusters after N observations evolves as $\log N$ (Arratia, Barbour, and Tavaré, 1992).

The two best-known variations of the regular Dirichlet process that address the "rich-get-richer" property control are the seminal Pitman-Yor process (Pitman and Yor, 1997) and the Uniform process (Wallach et al., 2010). Each of them can be expressed in a similar form as Eq.IV.1, and will be detailed in Section IV.3.

IV.2.2.b Hawkes Process

Hawkes processes are typically used to model sequences of events in continuous time, under the assumption that the probability that new events happen is conditioned by the realization of earlier events. Typically, it can be used to model retweet cascades, as the more retweets there are, the most likely other retweets are to appear (Chen and Tan, 2018).

A Hawkes process is defined as a self-stimulating temporal point process. Point processes are fully characterized by an intensity function $\lambda(t)$, which is related to the probability $P(t_{events} \in [t; t + \Delta t])$ of an event happening between t and $t + \Delta t$

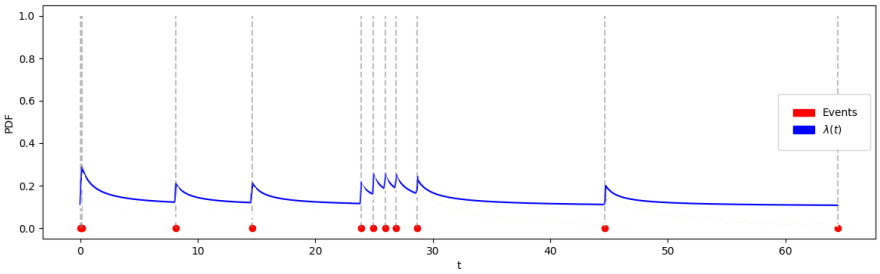

FIGURE IV.2: A realization of a Hawkes process

by $\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t_{events} \in [t; t+\Delta t])}{\Delta t}$. In the case of Hawkes processes, $\lambda(t)$ is defined conditionally on all the events that happened at times lower than t, that it the history up to t, noted $\mathcal{H}_{<t}$. We provide a synthetic realization of a Hawkes process as an illustration in Fig. IV.2. In our setup, we define one Hawkes process for each cluster, independent from the others. The intensity of the Hawkes process associated with cluster c is defined as:

$$\lambda_c(t|\mathcal{H}_{<t,c}) = \sum_{\mathcal{H}_{<t,c}} \vec{\alpha_c}^T \cdot \vec{\kappa}(t_{i,c}) \tag{IV.2}$$

where $t_{i,c}$ is the time of the i^{th} observed in cluster c, $\mathcal{H}_{<t,c} = \{t_{i,c} | t_{i,c} < t\}_{i=1,2,...}$ is the history of events in cluster c up to t, $\vec{\alpha_c}$ is a vector of coefficients, $\vec{\kappa}(t)$ is a vector of kernel functions with the same dimension as $\vec{\alpha}$ and $\cdot$ represents the dot product. The kernel functions are defined at the beginning of the algorithm and are not modified afterwards. We will later infer the weights vector $\vec{\alpha}$ to determine which entries of the kernel vector are the most relevant for a given situation. This technique has become standard in Hawkes processes modelling and used in several occasions (Du et al., 2012; Yu, Gupta, and Kolar, 2017).

The likelihood of a combination of C independent Hawkes processes can be written:

$$\begin{aligned}
\mathcal{L}(\vec{\lambda}|\mathcal{H}_{<T,c}) &= \prod_{c \in C} \mathcal{L}_c(\lambda_c|\mathcal{H}_{<T,c}) \\
&= \prod_c e^{-\int_0^T \lambda_c(t)dt} \prod_{t_{i,c}} \lambda_c(t_i|\mathcal{H}_{<t_{i,c},c}) \\
&= e^{-\sum_c \int_0^T \lambda_c(t|\mathcal{H}_{<t,c})dt} \prod_{t_{i,c'}, c'=c} \lambda_c(t_{i,c'}|\mathcal{H}_{<t_{i,c'},c})
\end{aligned} \tag{IV.3}$$

where T is the upper time of the considered observation window, going from 0 to T. From now on, the dependence of Hawkes intensity functions on the history will be implicit for clarity of presentation; we define $\lambda(t) := \lambda(t|\mathcal{H}_{<t})$.

IV.2.2.c Dirichlet-Hawkes Process – Expression

As the merging of Dirichlet processes and Hawkes processes, Dirichlet-Hawkes processes aim to model self-stimulating clusters in continuous time. In their definition of the DHP, the authors of (Du et al., 2015) substitute the counts N_k of the DP (Eq. IV.1) with the inferred Hawkes intensities (Eq. IV.2), resulting in the following

form for the Dirichlet-Hawkes prior:

$$P(C_i = c|t_i, \lambda_0, \mathcal{H}_{<t_i,c}) = \begin{cases} \frac{\lambda_c(t_i)}{\lambda_0+\sum_{c'}\lambda_{c'}(t_i)} & \text{if c}\leq\text{C} \\ \frac{\lambda_0}{\lambda_0+\sum_{c'}\lambda_{c'}(t_i)} & \text{if c=C+1} \end{cases} \tag{IV.4}$$

where t_i is the arrival time of document i. The expression in Eq. IV.4 is used as a prior that accounts for the publication dynamics in Bayesian modelling. It is used as an *a priori* for a model that explicitly consider the textual content of a document.

In Eq. IV.4, the authors consider a time-independent intensity function $\lambda(t) = \lambda_0$. This process is used as the Dirichlet-Hawkes equivalent of the concentration parameter α_0 in a Dirichlet process (see Eq. IV.26). It corresponds to a background Poisson process which determines the rate at which new Hawkes processes are launched –that is, at which new dynamic clusters are created.

Given the existence of the underlying Poisson process of parameter λ_0, the temporal likelihood of the processes associated with all clusters can be written:

$$\begin{aligned} \mathcal{L}(\vec{\lambda}|\mathcal{H}_{<T,c}) &= \mathcal{L}(\lambda_0)\prod_c \mathcal{L}_c(\lambda_c) \\ &= e^{-\int_0^T \lambda_0 dt}\prod_c e^{-\int_0^T \lambda_c(t)dt}\prod_{t_{i,c}} \lambda_c(t_i) \\ &= e^{-\lambda_0 T - \sum_c \int_0^T \lambda_c(t)dt}\prod_{t_{i,c'},c'=c} \lambda_c(t_{i,c'}) \end{aligned} \tag{IV.5}$$

Note that $\mathcal{L}(\lambda_0) = e^{-\int_0^T \lambda_0 dt}$ because no event is ever assigned to the Poisson process; the product over the history of events runs over 0 events, which equals 1 by convention. We recall that the Hawkes intensity dependence on the history of events is implicit; $\lambda(t) := \lambda(t|\mathcal{H}_{<t})$.

IV.2.2.d Textual modelling

We choose to model the textual content of documents as the result of a Dirichlet-Multinomial distribution. This model is purposely simple to ease the understanding, but can easily be replaced by a more complex one. A more complete textual modelling is out of the scope of this presentation, which focuses on the definition of the DHP prior. A document will be associated with a given cluster according to word count in every cluster and words count in the document only. The generative process is as follows:

$$\theta_i \sim Dir(\theta_0) \quad ; \quad \omega_{v,i} \sim Mult(\theta_i) \tag{IV.6}$$

where θ_i is the cluster of document i, and $\omega_{v,i}$ is the v^{th} word of document i. Let $\mathcal{L}_{txt}(\vec{C}_{<i,c}|N_{<i,c}, \theta_0)$ be the marginal joint distribution of every document's cluster allocation up to the i^{th} one. The likelihood of the i^{th} document belonging to cluster c

can then be expressed as:

$$\begin{aligned}
\mathcal{L}(C_i = c | N_{<i,c}, n_i, \theta_0) &= P(n_i | C_i = c, N_{<i,c}, \theta_0) \\
&= \frac{\mathcal{L}_{txt}(\vec{C}_{<i,c} | N_{<i,c}, \theta_0)}{\mathcal{L}_{txt}(\vec{C}_{<i-1,c} | N_{<i,c}, \theta_0)} \\
&= \frac{\frac{\cancel{\Gamma(\theta_0)}}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\cancel{\Gamma(\theta_{0,v})}}}{\frac{\cancel{\Gamma(\theta_0)}}{\Gamma(N_c + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + \theta_{0,v})}{\cancel{\Gamma(\theta_{0,v})}}} \\
&= \frac{\Gamma(N_c + \theta_0)}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\Gamma(N_{c,v} + \theta_0)}
\end{aligned} \tag{IV.7}$$

where N_c is the total number of words in cluster c from observations previous to i, n_i is the total number of words in document i, $N_{c,v}$ the count of word v in cluster c, $n_{i,v}$ the count of word v in document i and $\theta_0 = \sum_v \theta_{0,v}$.

IV.2.3 Limits

A common feature of all the models we mentioned in this section is that they use a non-parametric Dirichlet process (DP) prior (describe in Eq. IV.1) or variations built on it, such as DHP (Eq. IV.4) and HDHP. Yet, on several occasions, it has been pointed out that there are no specific reasons to use this process in particular and that alternative forms might work better depending on the dataset. In (Welling, 2006), the author relaxes several conditions associated with DP and shows that alternative priors are an equally valid choice in Bayesian modelling. In (Wallach et al., 2010), the authors derive the Uniform process (UP) and show that it performs better on a document clustering task. In Section IV.3, we generalize UP and DP within a more general framework: the Powered Dirichlet process (PDP). We show it performs better than DP on several datasets. As we show in Section IV.3 and Section IV.4, considering alternative definitions of the DP significantly leads to significantly different results.

As for DHP itself, it does not work well when the textual information within documents conveys little information, that is when the text is short (Yin et al., 2018) or when vocabularies overlap significantly. To answer this problem, the authors develop an approach based on Dirichlet process mixtures, which is not designed for continuous-time document streams – the temporal aspect comes from a sampling function as in (Ahmed and Xing, 2008; Blei and Frazier, 2010). There are other limiting cases for DHP, for instance when temporal information conveys little information (few observations, overlapping temporal intensities) or when documents within textual clusters do not follow the same temporal dynamics. To overcome those limitations, we develop the Powered Dirichlet-Hawkes process in the next section.

In addition, we uncover in Section IV.4 new limiting cases in which DHP fails, typically when publication times convey little information (overlapping Hawkes intensities, few observations). We also show there are cases where different documents generated from the same textual cluster do not follow the same temporal dynamics (they are associated with a different $\lambda(t)$), which the DHP is not designed to handle. For instance, an article published by a popular newspaper is unlikely to have the same influence on subsequent similar articles (temporal dynamics) as the same article published by a less popular newspaper. Textual content is not perfectly correlated to publication times.

IV.3 Powered Dirichlet Process – Alleviate the "rich-get-richer" assumption

IV.3.1 Introduction

The limits of DHP raised in Section IV.2.3 can be overcome by redefining the Dirichlet Process (DP) it is built on. From a broader perspective, most existing works based on Dirichlet Processes can be revisited by considering alternative forms for Dirichlet-like priors. These priors have been used extensively in Bayesian clustering over the last years. A non-exhaustive list of application includes medicine, (Guimerà and Sales-Pardo, 2013), natural language processing (Blei, Ng, and Jordan, 2003; Yin and Wang, 2014), genetics (Qin et al., 2003; Jensen and Liu, 2008; McDowell et al., 2018), recommender systems (Airoldi et al., 2008; Godoy-Lorite et al., 2016), sociology (Guimera, Llorente, and Sales-Pardo, 2012; Cobo-López, Godoy-Lorite, and Duch, 2018), etc.

The key idea of Bayesian clustering is to simulate a corpus of independent observations by drawing them from a set of latent variables (clusters). Those clusters are each associated with a probability distribution on the observations, whose parameters are drawn from a prior distribution, as we will formulate mathematically later. Now, an often desirable property of Bayesian models is to make them nonparametric. In our case, it means that both the number of clusters and their associated distributions are inferred. A very popular prior on clusters distributions that allows this is the Dirichlet process. It includes a chance for a new cluster to be created in the prior probability of a distribution (often when an observation is not likely to be explained by existing clusters). Otherwise, the observation is associated *a priori* with an existing cluster with a probability proportional to that cluster's population. Note that the model the prior is associated with might use this prior information, but might as well ignore it by design.

However, the Dirichlet process (and the related Pitman-Yor process) prior comes with a strong hypothesis on the way observations are allocated to various clusters: the *rich-get-richer* property (Ferguson, 1973). A new observation *a priori* belongs to a cluster with a probability proportional to the number of observations already present in the cluster (see Eq.IV.1); large clusters have a greater chance to get associated with new observations. This modelling implies a strong assumption on the way data is generated. It has already been pointed out (Welling, 2006) that there is a need for more flexible priors.

IV.3.2 Motivation

The need for alternative priors is particularly relevant in the case of imbalanced data and scale-dependent clustering. A cluster made of fewer entities might go unnoticed due to a rich-get-richer prior. As an example of the imbalance problem, consider a case where data is processed sequentially –which is often the case when it comes to the Dirichlet process prior. The first observation from a new cluster would then have a much larger *a priori* probability to belong to a populated but irrelevant cluster, than to open a new one (this probability decreases as $\frac{1}{N_{obs}}$). This typically happens when sampling topics from news streams (Wallach et al., 2010; Xu, Li, and Qiang, 2021). In the case of scale-dependent clustering, a similar problem arises. Consider clustering people pinpointed on a map. Tiny clusters (at the scale of cities, for instance) might go unnoticed at larger scales (countries, for instance). To spot city clusters on a

world map, the "rich-get-richer" assumption becomes irrelevant and a "rich-get-no-richer" prior would be preferred (Wallach et al., 2010); the optimal solution might as well be in-between these two priors, as in Fig.IV.6. We design a method to bridge the variety of possible priors between the Dirichlet process and the Uniform process in a continuous fashion. By generalizing existing works, our method shows there exist Dirichlet-based priors that exhibit a yet unexplored class of behaviours, such as "rich-get-less-richer", "rich-get-more-richer" and "poor-get-richer".

Little effort has been put into exploring alternative forms of priors for nonparametric Bayesian modelling. In this section, we address this problem by deriving a more general form of the Dirichlet process that explicitly controls the importance of the "rich-get-richer" assumption. Explicitly, we derive the Powered Chinese Restaurant Process (PCRP) that generalizes state-of-the-art works such as UP (Wallach et al., 2010) and DP. We show that controlling the "rich-get-richer" prior of simple models yields better results on synthetic and real-world datasets.

This work is motivated by the need to control the importance of the "rich-get-richer" assumption in Dirichlet process (DP) priors. Developing such a more permissive prior would impact many works based on the vanilla Dirichlet Process. In our case, we are particularly interested in the implication of DHP for the reasons discussed in Section IV.2.3.

The "rich-get-richer" property of the DP may not always be the most suitable prior for modelling a given dataset. The usual motivation for using a DP prior is that a new observation has a probability of being assigned to any cluster proportional to its population (or intensity function, as in (Du et al., 2015)) in the absence of external information (such as inter-points distance in case of spatial clustering, for instance). However, this assumption might be flawed in several cases.

Typically, most state-of-the-art works rely on tuning a parameter α (see Eq.IV.1) to get the "right" number of clusters (this parameter shifts the distribution of the number of clusters as $\mathbb{E}(K|N) \propto \alpha \log N$ with K the number of clusters and N the number of observations). However, we argue this is a bad practice. Imagine sampling topics in a news stream: there is no specific reason for topics to appear at a rate $\alpha \log N$ as in the regular DP. The logarithmic dependence on N cannot be tuned using α. Such prior is then unfit to describe the data correctly as the number of observations grows; worst, it can lead the model it is coupled with on the wrong track. Moreover, when considering observations streams (Wallach et al., 2010; Xu, Li, and Qiang, 2021), there is usually no specific *a priori* reason for a new observation to belong to a cluster with a probability depending linearly on the cluster's size, as in the regular Dirichlet and Pitman-Yor processes (and variants). The order of appearance then plays too important of a role. Typically, newer observations might be associated with an existing large cluster, despite being radically different from the points it comprises, due to a too influential "rich-get-richer" assumption. To alleviate those assumptions, we develop a more general form of the DP process allowing a natural control of the "rich-get-richer" property.

IV.3.3 Background

IV.3.3.a Previous works

Dirichlet process

The standard Dirichlet process has already been detailed in Section IV.2.2.a. For completeness, we recall its expression as a Chinese Restaurant Process:

$$CRP(C_i = c|\alpha, C_1, C_2, ..., C_{i-1}) = \begin{cases} \frac{N_c}{\alpha+N} \text{ if c = 1, 2, ..., K} \\ \frac{\alpha}{\alpha+N} \text{ if c = K+1} \end{cases} \tag{IV.8}$$

Uniform process

A first process that breaks the "rich-get-richer" property is the Uniform process. It has been used on some occasions (Qin et al., 2003; Jensen and Liu, 2008) without focus on the prior itself. It has later been formalized and studied in comparison with the regular Dirichlet and Pitman-Yor processes (Wallach et al., 2010). It can be written as follows:

$$UP(C_i = c|\alpha, C_1, C_2, ..., C_{i-1}) = \begin{cases} \frac{1}{\alpha+K} \text{ if c = 1, 2, ..., K} \\ \frac{\alpha}{\alpha+K} \text{ if c = K+1} \end{cases} \tag{IV.9}$$

This formulation completely gets rid of the "rich-get-richer" property. The probability of a new client joining an occupied table is a uniform distribution over the number of occupied tables; it does not depend on the tables' population. In (Wallach et al., 2010), it has been shown that the expected number of tables evolves with N as $\sqrt{N}$. Removing the "rich-get-richer" property leads to a flat prior. As we show later, our formulation allows to retrieve such flat priors and thus generalizes the Uniform Process.

Pitman-Yor process

Following the Chinese Restaurant process metaphor, the Pitman-Yor process (Pitman and Yor, 1997; Ishwaran and James, 2003) proposed to incorporate a *discount* parameter when a client opens a new table. Mathematically, the process can be formulated as:

$$PY(C_i = c|\alpha, \beta, C_1, C_2, ..., C_{i-1}) = \begin{cases} \frac{N_c-\beta}{\alpha+N} \text{ if c = 1, 2, ..., K} \\ \frac{\alpha+\beta K}{\alpha+N} \text{ if c = K+1} \end{cases} \tag{IV.10}$$

The introduction of the parameter $\beta > 0$ increases the probability of creating new clusters. A table with a small number of customers has significantly fewer chances to gain new ones, while the probability of opening a new table increases significantly. It can be shown that the number of tables evolves with the number of clients N as N^β (Sudderth and Jordan, 2009; Wallach et al., 2010; Goldwater, Griffiths, and Johnson, 2011). However, this process does not control the arguable "rich-get-richer" hypothesis (Welling, 2006), since the relation to the population of a table remains linear; it only shifts this dependence of a value β. It makes so by creating clusters based on the number of existing clusters and the total number of observations, but not according to the population of already existing clusters. Those play the same role in the Pitman-Yor process as in the DP. The Pitman-Yor process thus comes with two limitations. Firstly, since $\beta > 0$, it cannot modify the process to generate fewer

clusters. Secondly, the discount parameter does not modify the linear dependence on previous observations for cluster allocations — rich still get richer; the prior is as peaky on large clusters as before. The present section offers to address those two limitations.

Other extensions

Another similar prior, the Power-law Indian Buffet Process, has been proposed so that a realization would yield a number of clusters obeying a power-law as the number of observations increases (Teh and Gorur, 2009). This formulation can be seen as a generalization of the Pitman-Yor process; it adds an additional parameter that sums with N in the denominator of Eq. IV.10. However, the posterior probability for a new customer to belong to a cluster depends linearly on each cluster's size, and the "rich-get-richer" hypothesis is preserved.

Finally, the Generalized Gamma Process proposed a similar discount idea to increase the probability of opening new clusters in (Lijoi, Mena, and Prünster, 2007). The proposed prior ((Lijoi, Mena, and Prünster, 2007)-Eq.4) modifies a cluster's probability to get chosen by subtracting a constant term from each cluster's population. Thus, the "rich-get-richer" property is not alleviated in their approach either, since the dependence on the cluster's population is still linear. As for the PY process, this formulation only allows to increase the number of clusters and does not alleviate the "rich-get-richer" hypothesis.

IV.3.3.b Contributions

In the next section, we derive the Powered Dirichlet Process (PDP) that allows controlling the "rich-get-richer" property. The process is also referred to as the Powered Chinese Restaurant Process (PCRP) due to its formulation being close to the vanilla metaphor. This new process generalises state-of-the-art works. Such generalization allows to define unexplored classes of *a priori* hypotheses: poor-get-richer, rich-get-no-richer (Uniform process), rich-get-less-richer, rich-get-richer (DP), and rich-get-more-richer. In doing so, we define the Powered Dirichlet-Multinomial distribution. We detail some key properties of the Powered Dirichlet Process (convergence, expected number of clusters). Finally, we show that controlling the "rich-get-richer" prior of simple models yields better results on synthetic and real-world datasets.

IV.3.4 The model

IV.3.4.a The Dirichlet-Multinomial distribution

As explained earlier, the Dirichlet distribution yields a collection of positive variables whose sum equals 1. The Multinomial distribution yields a sum of counts in each of the K "boxes" (here clusters) following N draws from an identical distribution over those boxes. We recall the definition of the Dirichlet distribution and of the Multinomial distribution:

$$Dir(\vec{p}|\vec{\alpha}) = \frac{\prod_k p_k^{\alpha_k - 1}}{B(\vec{\alpha})} \qquad Mult(\vec{N}|N, \vec{p}) = \frac{\Gamma(\sum_k N_k + 1)}{\prod_k \Gamma(N_k + 1)} \prod_k p_k^{N_k} \tag{IV.11}$$

with $\vec{N} = (N_1, N_2, ..., N_K)$ where N_k is the integer number of draws assigned to cluster k, $N = \sum_k N_k$ the total number of draws, $\Gamma(x) = (x-1)!$ is the gamma function, and $B(\vec{x}) = \prod_k \Gamma(x_k) / \Gamma(\sum_k x_k)$ is the beta function.

The Dirichlet-Multinomial distribution merges both distributions: the probabilities for each "box" in the Multinomial distribution are sampled once from a Dirichlet distribution. As we will show later, the Dirichlet process can be derived from this distribution. The Dirichlet-Multinomial distribution is defined as follows:

$$\begin{aligned} p(\vec{N}|\vec{\alpha}, n) &= \int_{\vec{p}} p(\vec{N}|\vec{p}, n) p(\vec{p}|\vec{\alpha}) d\vec{p} \\ &= \frac{(n!)\Gamma(\sum_k \alpha_k)}{\Gamma(n + \sum_k \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(N_k + \alpha_k)}{(N_k!)\Gamma(\alpha_k)} \\ \text{where } \vec{p} &\sim Dir(\vec{p}|\vec{\alpha}) \,;\, \vec{N} \sim Mult(\vec{N}|n, \vec{p}) \end{aligned} \tag{IV.12}$$

In Eq.IV.12, we sample n values over a space of K distinct clusters each with probability $\vec{p} = (p_1, p_2, ..., p_K)$, using a Dirichlet prior with parameter $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)$. To derive the Dirichlet process equation, we must compute a new observation's conditional distribution to belong to any cluster given the allocation of all the previous random variables when $K \to \infty$.

IV.3.4.b Powered conditional Dirichlet prior

In the derivation of the standard Dirichlet-Multinomial posterior predictive, we consider a single draw from the Multinomial distribution (i.e., a categorical distribution) with a Dirichlet prior on the parameter $\vec{p}$. Usually, this prior is linearly dependent on previous draws from the distribution. We propose to modify this assumption by using a Dirichlet prior that depends non-linearly on the history of draws as:

$$Dir_r(\vec{p}|\vec{\alpha}, \vec{N}) = \frac{1}{B(\vec{\alpha} + \vec{N}^r)} \prod_k p_k^{\alpha_k + N_k^r - 1} \tag{IV.13}$$

In Eq.IV.13, the vector $\vec{N}^r$ shifts the parameter $\vec{\alpha}$ according to the count of draws allocated to each cluster k up to the n^{th} draw. The parameter $r \in \mathbb{R}$ controls the intensity of this shift for each entry of $\vec{N}$.

We demonstrate that the Powered Dirichlet distribution is a conjugate prior of the Multinomial distribution, by writing Eq.IV.13 as:

$$\begin{aligned} Dir_r(\vec{p}|\vec{\alpha}, \vec{N}) &= \frac{1}{B(\vec{\alpha} + \vec{N}^r)} \prod_k p_k^{\alpha_k - 1} \prod_k p_k^{N_k^r} \\ &\overset{\text{Eqs.IV.11}}{=} \frac{B(\vec{\alpha}) \prod_k N_k^r!}{B(\vec{\alpha} + \vec{N}^r)(\sum_k N_k^r)!} Dir(\vec{p}|\vec{\alpha}) Mult(\vec{N}^r | \sum_k N_k^r, \vec{p}) \\ &\propto Dir(\vec{p}|\vec{\alpha}) Mult(\vec{N}^r | \sum_k N_k^r, \vec{p}) \end{aligned} \tag{IV.14}$$

where the prior on vector $\vec{N}$ is a regular Multinomial distribution of parameter $N = \sum_k N_k^r$. Note that for certain values of r, the vector $\vec{N}^r$ might not be made of integer values; the resulting Multinomial prior on $\vec{N}^r$ must then be expressed in terms of Γ functions (see Eq.IV.11) to be valid for $\vec{N}^r \in \mathbb{R}^{|\vec{N}|}$. Distributions of non-integer counts are not new in the literature (Khurshid, Ageel, and Lodhi, 2005; McCarthy, Chen, and Smyth, 2012; Ghitza and Gelman, 2013) and are essentially allowed by the generalized definition of the factorial function in terms of the gamma function. When $r = 1$, we recover the standard Dirichlet-Multinomial prior on $\vec{p}$ for the n^{th} draw; the history of draws $\vec{N}$ can be expressed as the result of N independent draws of equal probability $\vec{p}$. When $r \neq 1$, the prior on $\vec{N}$ is sampled

from a Multinomial distribution in which the number of samples drawn depends on r as $\sum_k N_k^r$. For instance, let $\vec{N} = (1,2)$ and $r = 2$: the resulting powered conditional Dirichlet prior would then be sampled from a Multinomial distribution $Mult(\vec{N} = (1,4)|N = 5, \vec{p} = (p, 1-p))$.

IV.3.4.c Posterior predictive

We now derive the posterior distribution for the n^{th} draw to belong to a cluster c given all previous draws. We assume that $\vec{C_-}$ represents all previous realizations up to $n-1$, that is, the cluster to which each previous draw has been associated. For simplicity of notation, we define the population of a cluster k at time $n-1$ as $N_k = |\{C_i|i = k\}_{i=1,2,\dots,n-1}|$. We are now looking at the probability distribution of its n^{th} draw to belong to c. It is expressed as the probability of a draw from the categorical distribution given all previous observations (because there is only one new draw, it is the same as a Multinomial distribution with parameter $N = 1$) combined with the powered Dirichlet prior defined Eq.IV.13. Then:

$$\begin{aligned} DirCat_r(C_n = c|\vec{\alpha}, \vec{C_-}) &= \int_{\vec{p}} Cat(C_n = c|\vec{p}) \underbrace{Dir_r(\vec{p}|\vec{\alpha}, \vec{N})}_{\text{Eq.IV.13}} \\ &= \int_{\vec{p}} \frac{1}{B(\vec{\alpha} + \vec{N}^r)} \prod_k p_k^{c_k + \alpha_k + N_k^r - 1} \\ &= \frac{B(\vec{c} + \vec{\alpha} + \vec{N}^r)}{B(\vec{\alpha} + \vec{N}^r)} \end{aligned} \tag{IV.15}$$

where $\vec{c}$ is a vector of the same length as $\vec{\alpha}$ whose c^{th} entry equals to 1, and 0 anywhere else. Alternative demonstrations of this result are possible (Wilks, 1992; Sethuraman, 1994).

IV.3.4.d Powered Chinese Restaurant process

We finally derive an expression for the Powered Chinese Restaurant process from Eq.IV.15. We recall that $N_k = |\{C_{-i}|i = k\}_{i=1,2,\dots,n-1}|$. Taking back the conditional probability for the n^{th} observation to belong to cluster c (Eq.IV.15), we have:

$$\begin{aligned} p(C_n =& c|\vec{C_-}, \vec{\alpha}) = DirCat_r(C_n = c|\vec{C_-}, \vec{\alpha}) \\ =& B(\vec{c} + \vec{N}^r + \vec{\alpha}) / B(\vec{N}^r + \vec{\alpha}) \\ =& \Gamma(N_c^r + \alpha_c + 1) \frac{\prod_{k \neq c} \Gamma(N_k^r + \alpha_k)}{\Gamma(1 + \sum_k N_k^r + \alpha_k)} \frac{\Gamma(\sum_k N_k^r + \alpha_k)}{\prod_k \Gamma(N_k^r + \alpha_k)} \\ =& \frac{(N_c^r + \alpha_c)}{\sum_k N_k^r + \alpha_k} \frac{\prod_k \Gamma(N_k^r + \alpha_k)}{\Gamma(\sum_k N_k^r + \alpha_k)} \frac{\Gamma(\sum_k N_k^r + \alpha_k)}{\prod_k \Gamma(N_k^r + \alpha_k)} \\ =& \frac{N_c^r + \alpha_c}{\sum_k N_k^r + \alpha_k} \end{aligned} \tag{IV.16}$$

Every cluster with $N_c = 0$ (empty clusters) has an identical probability of getting chosen. Besides, the result is identical if either of them gets chosen. Therefore, we can express the probability of choosing any empty clusters as a function of $\alpha = \sum_k \alpha_k$.

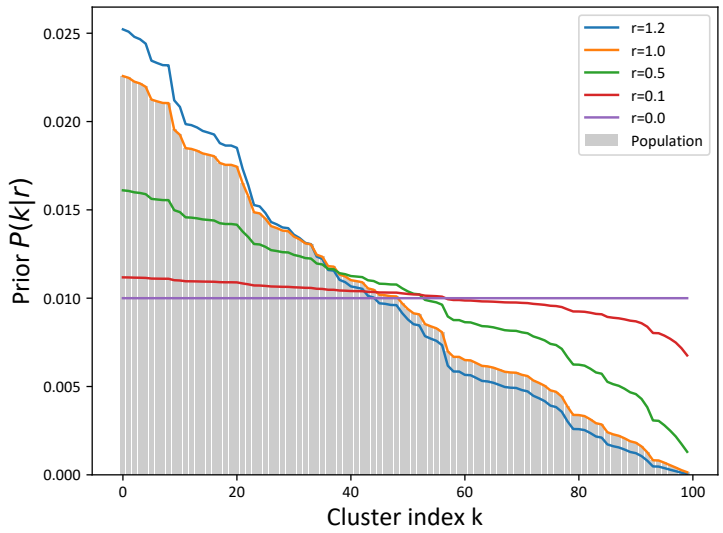

FIGURE IV.3: Effect of r on the Powered Chinese Restaurant process prior probability. The grey bars represent the population in each cluster at a given time. The solid lines represent the prior probability of the next observation to belong to each of these. For $r = 1$, we recover the Dirichlet Process prior.

Finally, taking the limit $K \to \infty$ and defining the limit value $\lim_{K\to\infty} \sum_k^K \alpha_k = \alpha$, we find the Powered Chinese Restaurant Process ($PCRP$):

$$PCRP(C_i = c|\alpha, C_1, C_2, ..., C_{i-1}) = \begin{cases} \frac{N_c^r}{\alpha + \sum_k^K N_k^r} \text{ if c = 1, 2, ..., K} \\ \frac{\alpha}{\alpha + \sum_k^K N_k^r} \text{ if c = K+1} \end{cases} \tag{IV.17}$$

The formal derivation of the Powered Chinese Restaurant process in Eq.IV.17 and the demonstration of its link to the conditional Dirichlet prior on $\vec{p}$ are the first main contribution of this section. Besides, this demonstration uncovers the link between the prior in Eq.IV.13 and an exotic formulation of the Multinomial distribution, which has never been considered before. Most importantly, it highlights that it originates from sampling every new observation from independent weighted Multinomial distributions of different parameters $N_r = \sum_k x_k^r$. As stated in the introduction, special cases of the process have already been used in some occasions (Qin et al., 2003; Jensen and Liu, 2008; Wallach et al., 2010) but never demonstrated. Furthermore, this formulation generalizes the Uniform process when $r \to 0$ (Wallach et al., 2010), the Dirichlet process when $r \to 1$ and the Pitman-Yor process when $r_k(N_k) = (log(1 - \beta/N_k) + log(N_k))/log(N_k)$ and $\alpha(K) = \alpha + \beta K$ (see Eq.IV.10, we recall that $e^{\log x} = x$). The present expression explicitly allows for controlling the importance of the "rich-get-richer" property as well as recovering state-of-the-art processes.

We illustrate the change in prior probability for an existing cluster to get chosen induced by the Powered Chinese Restaurant process in Fig.IV.3 – we do not plot the prior probability for a new cluster to be created. This figure plots the population of clusters (grey bars) and their associated prior probability of getting chosen. When $r > 1$, the most populated clusters are associated with a more significant prior probability than in the standard CRP, whereas the less populated ones have even fewer chances to get chosen; rich-get-more-richer, the prior on population is peakier on large clusters. On the other hand, when $r < 1$, most populated clusters have fewer

chances to get chosen than in CRP, whereas less populated ones have an increased chance of getting chosen; rich-get-less-richer, the prior on population is flatter across clusters of different sizes. In the limit case $r = 0$, the clusters' population does not play any role anymore; rich get-no-richer, the prior is flat over all clusters. Note that if we wanted to represent the Pitman-Yor process prior in this figure, it would correspond to the plot for $r = 1$ vertically shifted of $-\beta$ (such as defined Eq.IV.10) leading to an increased probability of creating a new cluster of βK (not represented in the plot) (Pitman and Yor, 1997). Varying the parameter α of $\Delta\alpha$ plays a similar role as β in this situation. It would uniformly shift the prior probability for each existing cluster to get chosen by $\frac{\Delta\alpha}{K}$ and increase the probability of creating a new one by $\Delta\alpha$. For Both Pitman-Yor and Dirichlet processes, the linear dependence of each cluster's population does not change.

In Fig. IV.3, we understand that the Powered Chinese Restaurant process allows for defining priors from clusters population that are not possible when tuning the Chinese Restaurant or Pitman-Yor processes. Introducing non-linearity in the dependence on previous observations allows giving more or less importance to the "rich-get-richer" property.

IV.3.5 Properties of the Powered Chinese Restaurant process

We will now investigate some key properties of the Powered Chinese Restaurant process. We recall that N_k is the population of the cluster k, and $N = \sum_k N_k$.

IV.3.5.a Convergence

Proposition 2. *For $N \to \infty$, the Powered Chinese Restaurant process converges towards a stationary distribution. When $r < 1$, it converges towards a uniform distribution over all the possible clusters, and when $r > 1$, it converges towards a Dirac distribution on a single cluster.*

Proof. We consider a simple situation where only 2 clusters are involved. The generalization to the case where K clusters are involved is straightforward. When the clusters' population is large enough, we make the following Taylor approximation:

$$(N_i + 1)^r = N_i^r (1 + \frac{1}{N_i})^r = N_i^r + r N_i^{r-1} + \mathcal{O}(N^{r-2}) \tag{IV.18}$$

Since the population of a cluster N_i is a non-decreasing function of N, we assume that first order Taylor approximation holds when $N \to \infty$. Given clusters population at the N^{th} observation, we perform a stability analysis of the gap between probabilities $\Delta p(N) = p_1(N) - p_2(N)$. We recall that the probability for cluster i to get chosen is $p_i(N) = N_i^r / (\sum_k N_k^r)$ and that either of the clusters is chosen with this probability at the next step (at step $N + 1$, $\Delta p(N + 1) = p_1(N + 1) - p_2(N)$ with probability $p_1(N)$ and $\Delta p(N + 1) = p_1(N) - p_2(N + 1)$ with probability $p_2(N)$). Explicitly the

variation of the gap between probabilities when N grows is written as:

$$\begin{aligned}
&\frac{p_1(N)(p_1(N+1)-p_2(N))+p_2(N)(p_1(N)-p_2(N+1))-\Delta p(N)}{\Delta p(N)} \\
&\overset{\text{Eq.IV.18}}{\approx} \frac{1}{p_1(N)-p_2(N)} \times \left(p_1(N)\frac{N_1^r-N_2^r+rN_1^{r-1}}{N_1^r+N_2^r+rN_1^{r-1}} + p_2(N)\frac{N_1^r-N_2^r-rN_2^{r-1}}{N_1^r+N_2^r+rN_2^{r-1}} \right) \\
&= \frac{2rN_1^rN_2^r}{(N_1^r+N_2^r+rN_1^{r-1})(N_1^r+N_2^r+rN_2^{r-1})} \left(\frac{N_1^{r-1}-N_2^{r-1}}{N_1^r-N_2^r} \right)
\end{aligned} \tag{IV.19}$$

We see in Eq.IV.19 that the sign of the variation of the gap between probabilities depend only on the term $\frac{N_1^{r-1}-N_2^{r-1}}{N_1^r-N_2^r}$. We can therefore perform a stability analysis of the Powered Chinese Restaurant process using only this expression.

When $0 < r < 1$, the sign becomes negative because the following relation holds: $N_1^{r-1} - N_2^{r-1} < 0 \Leftrightarrow N_1^r - N_2^r > 0 \; \forall N_1, N_2$; that makes right hand side of Eq.IV.19 negative. Therefore, adding a new observation statistically reduces the gap between the probabilities of the two clusters. We could forecast this prediction from Eq.IV.18 by seeing that adding a new observation to a large cluster increases its probability to get chosen lesser than for a small cluster – rich-get-less-richer. Moreover, we see from Eq.IV.18 that a crowded cluster (such as $N_1^r \gg N_2^r$) see its probability evolve as N^{r-1}. Asymptotically, the only fixed point of Eq.IV.19 when $N \to \infty$ is $N_1 \to N_2$, which implies a uniform distribution.

On the contrary, when $r > 1$ we have the following relation: $N_1^{r-1} - N_2^{r-1} > 0 \Leftrightarrow N_1^r - N_2^r > 0 \; \forall N_1, N_2$; ; that makes right hand side of Eq.IV.19 positive. Adding a new observation statistically increases the gap between probabilities. From Eq.IV.18, we see that adding an observation to a large cluster increases its probability with its population – rich-get-more-richer. In this case, Eq.IV.19 has $K+1$ fixed points, with K the number of clusters. The uniform distribution is an unstable fixed point, while K Dirac distributions (each on one cluster) are stable fixed points of the system. It means the gap converges to 1, that is a probability of 1 for one cluster and a probability of 0 for the others.

When $r = 1$, the right-hand side of Eq.IV.19 is null. It means the gap remains statistically constant $\forall N_i$, which is a classical result for the regular Dirichlet process. This convergence has already been studied on many occasions (Ferguson, 1973; Arratia, Barbour, and Tavaré, 1992).

We note that as $r \to 0$, Eq.IV.19 is not defined anymore. That is because the probability for a cluster to be chosen does not depend on its population anymore. In this case, $p_1(N) - p_2(N) \propto N_1^0 - N_2^0 = 0$: the probability for any cluster to be chosen is equal, hence the Uniform process – "rich-get-no-richer".

□

IV.3.5.b Expected number of tables

Proposition 3. *When N is large, $\sum_k N_k^r$ varies with N as $N^{\frac{r^2+1}{2}}$ when $r < 1$, and with N^r when $r \geq 1$.*

Proof. Taking back Eq.IV.17, we are interested in the variation of $p_i = \frac{N_i^r}{\sum_k N_k^r}$ according to N when N_i^r is large:

$$p_i(N+1) - p_i(N) \approx \begin{cases} \frac{rN_i^{r-1} + \mathcal{O}(N^{r-2})}{\sum_k N_k^r} & \text{if } N_i \text{ grows} \\ 0 & \text{otherwise} \end{cases} \tag{IV.20}$$

We see in Eq.IV.20 that for $r < 1$, the larger N_i the slower the variation of p_i. It means that for large N_i^r, we can write $N_i \propto Np_i$, with p_i a constant of N. Since N_i is either way a non-decreasing function of N, we reformulate the constraint N_i^r large in N^r large.

For $r > 1$, the probability p_i varies greatly with N and quickly converges to 1 for large N (see Proposition 2), and so $N_i \approx N$ for cluster i and $N_{j\neq i} \ll N_i \; \forall j$.

Since the sum $\sum_k N_k^r$ essentially varies according to large N_k, we can approximate $\sum_k N_k^r \approx N^r \sum_k p_k^r$ for large N^r.

Besides, we showed in Proposition 2 that for large N the process converges towards a uniform distribution for $r < 1$ and towards a Dirac distribution when $r > 1$. Therefore, we can express $\sum_k^K p_k^r$ as:

$$\sum_k^K p_k^r \overset{N\gg 1}{\approx} \begin{cases} K^{1-r} & \text{for } r < 1 \\ 1 & \text{for } r \geq 1 \end{cases} \tag{IV.21}$$

Based on the demonstration of Eq.4 in (Wallach et al., 2010), we suppose that K evolves with N as $N^{\frac{1-r}{2}}$ when $r < 1$. We verify that this assumption holds in the Experiment section.

Therefore, we can write:

$$\sum_k N_k^r \approx N^r \sum_k^K p_k^r \approx \begin{cases} N^r \left(N^{\frac{1-r}{2}}\right)^{1-r} = N^{\frac{1+r^2}{2}} & \text{for } r < 1 \\ N^r & \text{for } r \geq 1 \end{cases} \tag{IV.22}$$

□

Proposition 4. *The expected number of tables of the Powered Chinese Restaurant process evolves with $N \gg 1$ as $H_{\frac{r^2+1}{2}}(N)$ for $r < 1$ and as $H_r(N)$ when $r \geq 1$, where $H_m(n)$ is the generalized harmonic number.*

Proof. In general, the expected number of clusters at the N^{th} step can be written as:

$$\mathbb{E}(K|N,r) = \sum_1^N \frac{\alpha}{\sum_k N_k^r + \alpha} \overset{N^r\gg 1}{\propto} \sum_1^N \frac{1}{\sum_k N_k^r} \tag{IV.23}$$

We showed in Proposition 3 that we can rewrite $\sum_k N_k^r \propto N^{\frac{r^2+1}{2}}$ when $r < 1$ and $\sum_k N_k^r \propto N^r$ when $r \geq 1$. Injecting this result in Eq.IV.23 for r, we get:

$$\mathbb{E}(K|N,r) \overset{N^r\gg 1}{\propto} \begin{cases} \sum_1^N \frac{1}{N^{\frac{r^2+1}{2}}} = H_{\frac{r^2+1}{2}}(N) \\ \sum_1^N \frac{1}{N^r} = H_r(N) \end{cases} \tag{IV.24}$$

□

For $r = 1, \mathbb{E}(K|N, r = 1) \propto H_1(N) \approx \gamma + \log(N)$ where γ is the Euler–Mascheroni constant, which is a classical result for the regular Dirichlet process.

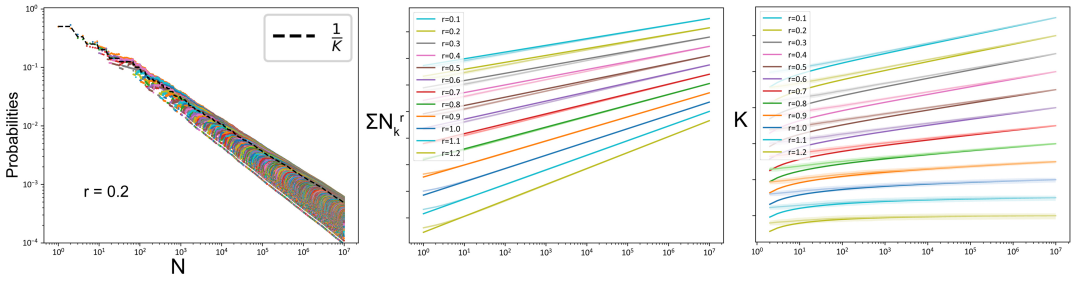

FIGURE IV.4: Numerical validation of Propositions 2 (left), 3 (middle), 4 (right). In the first plot, K is the number of non-empty clusters. In the second and third plots, the theoretical results are the solid lines, and the associated numerical results are the transparent lines of same colour. Except for small N, the difference between theory and experiments is almost indistinguishable.

When $r > 1$ and $N \to \infty$, the term $H_{\frac{r^2+1}{2}}(N)$ converges towards a finite value and the sum $\sum_k p_k^r$ goes to 1 (see Proposition 2). By definition $\mathbb{E}(K|N, r > 1) \overset{N\to\infty}{\propto} \zeta(\frac{r^2+1}{2})$, where ζ is the Riemann Zeta function.

When $r < 1$, we can approximate the harmonic number in a continuous setting. We rewrite Eq.IV.24 as:

$$\begin{aligned}\mathbb{E}(K|N, r) &\overset{N^r\gg 1}{\propto} \sum_{n=1}^{N} \frac{1}{n^{\frac{r^2+1}{2}}} \overset{N^r\gg 1}{\approx} \int_1^N n^{-\frac{r^2+1}{2}} dn \\ &= \frac{2}{1-r^2}(N^{\frac{1-r^2}{2}} - 1)\end{aligned} \tag{IV.25}$$

It can be shown that $\frac{N^{1-x}-1}{1-x} = H_x(N) + \mathcal{O}(\frac{1}{N^x})$. Therefore, the Powered Chinese Restaurant process exhibits a power-law behaviour similar to the Pitman-Yor process Eq.IV.10 for $r = \sqrt{1-2\beta}$ for $0 < r < 1$. For values of $r > 1 \Leftrightarrow \beta < 0$, the equivalent Pitman-Yor process is not defined unlike the Powered Chinese Restaurant process. Note that there is *a priori* no reason for r to be constrained in the domain of real number. Complex analysis of the process might be an interesting lead for future works.

IV.3.6 Experiments

IV.3.6.a Numerical validation of propositions

First of all, we present numerical confirmations of propositions stated above (Propositions 2, 3, 4) by simulating 100 independent Powered Chinese Restaurant processes with parameter $\alpha = 1$ for various values of r. Note that in all the theorems we verify here, α acts as a scaling factor without modifying the shape of the results discussed here. We present the results of numerical simulations in Fig.IV.4.

In the **left** part, we plot the evolution of the probability for each cluster to be chosen as N grows for $r = 0.2$ for one run. We see that the probabilities do not remain constant but instead diminish as the number of clusters grows. The figure suggests they all converge to a common value (a uniform probability) as shown in Proposition 2. The black line shows the probability of a uniform distribution. We chose not to show the results for $r > 1$; in this case, one probability goes to 1 as the other fades to 0 as N grows, as expected.

In the **middle** part of the figure, we plot the expression for $\sum_k N_k^r$ derived in Proposition 3 (solid lines) versus the value of the sum from experimental results

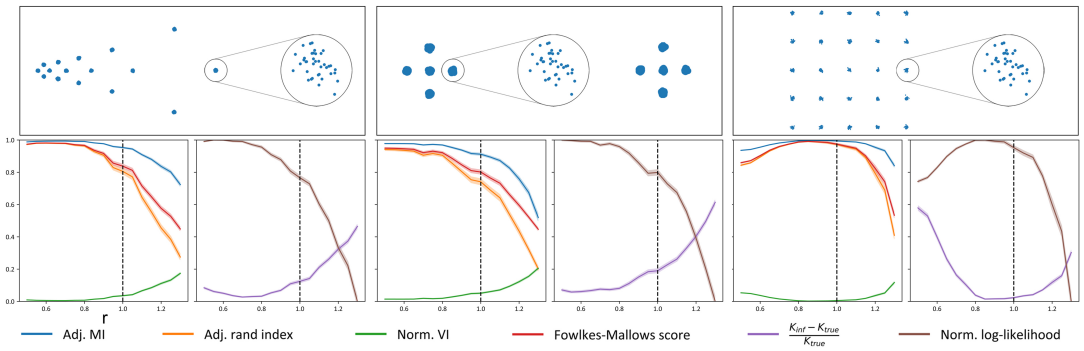

FIGURE IV.5: Application on synthetic data. (Top) Original datasets used for the experiments (Density, Diamond, and Grid). (Bottom) Results for various values of r; the x and y axes are all the same. The dashed line indicates the regular DP prior as $r = 1$. The error corresponds to the standard error of the mean over all runs.

(transparent lines), averaged over 100 runs. Note that plots are in a log-log scale and that curves have been shifted vertically for visualization purposes. As assumed in Proposition 3, the approximation holds for all values of r.

Finally, in the **right** picture, we plot the evolution of the number of clusters K versus N according to Proposition 4 (solid lines) and experiments (transparent lines). The error bars correspond to the standard deviation over the 100 runs. We see that the expression derived in Proposition 4 accounts well for the evolution of the number of clusters. Note that plots are in a log-log scale and that curves have been shifted vertically for visualization purposes. We must point out that there is a constant shift from experiments to the theory that does not appear on the plot (because of the rescaling). This shift comes from the approximation of large N^r which is not valid at the beginning of the process. However, it does not play any role in the evolution of K as N grows large enough.

IV.3.6.b Use case: infinite Gaussian mixture model

We now illustrate the usefulness of a prior that alleviates the "rich-get-richer" property on several synthetic datasets and on a real-world application. We choose to consider as an illustration its use as a prior in the infinite Gaussian mixture model. We choose this application to ease visual understanding of the implications of the PCRP, but the argument holds for other models using DP priors as well (text modelling, gene expression clustering, etc.).

We consider a classical infinite Gaussian mixture model coupled with a Powered Dirichlet process prior. We fit the data using a standard collapsed Gibbs sampling algorithm for IGMM (Rasmussen, 1999; Wallach et al., 2010; Yin and Wang, 2014), with a Normal Inverse Wishart prior on the Gaussians' parameters. The input data is shuffled at each iteration to reduce the ordering bias from the dataset. Note that we cannot completely get rid of the bias because the Powered Dirichlet Process is not exchangeable for all r. The problem has been addressed on numerous occasions (Uniform process (Wallach et al., 2010), distance-dependent CRP (Blei and Frazier, 2010; Ghosh et al., 2014), spectral CRP (Socher, Maas, and Manning, 2011)) and shown to induce negligible variations of results in the case of Gibbs sampling. We stop the sampler once the likelihood of the model reaches stability ; we repeat this procedure 100 times for each value of r. Finally, the parameter α is set to 1 in all experiments (see Section IV.3.2).

TABLE IV.1: Numerical results of the Powered Dirichlet process, Uniform process and Dirichlet process priors coupled to a standard Infinite Gaussian Mixture Model, for the 3 synthetic datasets plotted Fig. IV.5 and 2 real-world datasets (100 runs each). We see that using PDP as prior makes the model outperform the baselines consistently on every metric. The standard error is given in shorthand form – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$.

		Adj.MI	Adj.RI	Norm.VI	$\frac{K_{inf}-K_{true}}{K_{true}}$
Density	PDP (r=0.60)	**0.992(1)**	**0.980(2)**	**0.006(1)**	**0.045(5)**
	DP	0.951(4)	0.797(17)	0.037(3)	0.128(10)
	UP	0.939(2)	0.854(4)	0.050(1)	0.548(1)
Diamond	PDP (r=0.50)	**0.982(2)**	**0.956(5)**	**0.011(1)**	**0.063(7)**
	DP	0.909(7)	0.731(19)	0.053(4)	0.202(12)
	UP	0.927(2)	0.844(6)	0.051(2)	0.544(2)
Grid	PDP (r=0.85)	**0.997(1)**	**0.990(2)**	**0.003(1)**	**0.014(2)**
	DP	0.995(1)	0.977(4)	**0.004(1)**	**0.018(3)**
	UP	0.811(1)	0.517(3)	0.154(1)	2.120(1)
Iris	PDP (r=0.90)	**0.868(4)**	**0.866(7)**	**0.057(2)**	**0.000(0)**
	DP	0.843(6)	0.820(12)	0.065(2)	0.030(10)
	UP	0.544(2)	0.295(3)	0.303(2)	2.777(32)
Wines	PDP (r=0.10)	**0.712(15)**	**0.637(20)**	**0.102(5)**	**0.157(17)**
	DP	0.589(19)	0.461(16)	0.128(4)	0.327(13)
	UP	**0.713(17)**	**0.657(21)**	**0.103(5)**	**0.147(17)**
Cancer	PDP (r=0.10)	**0.254(17)**	**0.278(21)**	**0.118(1)**	**0.000(0)**
	DP	0.085(16)	0.094(19)	0.108(2)	**0.000(0)**
	UP	**0.271(17)**	**0.300(21)**	**0.118(1)**	**0.000(0)**
20-NG	PDP (r=0.80)	**0.421(4)**	**0.119(3)**	**0.477(3)**	-
	DP	0.404(4)	0.105(4)	0.491(3)	-
	UP	0.000(4)	0.000(0)	0.830(3)	-

Note that we choose not to compare to other types of clustering algorithms. In this section, we demonstrate the power of alternative forms of the Dirichlet process. The argument on a simple model (here a regular DP combined with IGMM) extends to other priors built on Dirichlet processes (Hierarchical and Nested Dirichlet processes). Besides, comparison of DP-based priors to other clustering methods (KNN, DBScan, Spectral clustering, etc.) has already been done numerous times and is out of this section's scope.

Synthetic data

Synthetic datasets are represented in Fig.IV.5-top and comprise N=1000 observations each. They have been generated by sampling from 2D Gaussian distributions whose relative parameters are explicit from Fig.IV.5.

We present the results on synthetic data in Fig.IV.5-bottom and in TableIV.1. We consider standard metrics in clustering evaluation with a non-fixed number of clusters: mutual information score and rand index both adjusted for chance (**Adj.MI** and **Adj.RI**), normalized variation of information (**Norm.VI**, lower is better), Fowlkes-Mallow score, marginal likelihood (normalized for visualization) and absolute relative variation of the inferred number of clusters according to the number used to generate the data ($\frac{\mathbf{K_{inf}}-\mathbf{K_{true}}}{\mathbf{K_{true}}}$, lower is better). Note that we purposely chose stereotypical cases to illustrate the argument better. The Density dataset on the **left** of Fig.IV.5 is informative about the change induced by r. Here, clusters are distributed

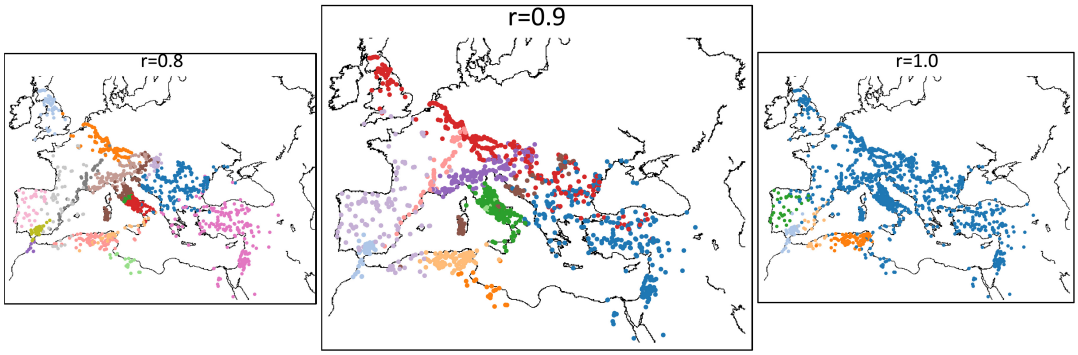

FIGURE IV.6: Application to spatial clustering on geolocated data for $r = 0.8$ (left), $r = 1$ (right) and $r = 0.9$ (middle). We see that the model using a Powered Dirichlet process prior for $r = 0.9$ and $r = 0.8$ describes the data better than the same model using a Dirichlet process prior ($r = 1$).

at various scales in the dataset; we see that the lower the value of r, the better the results. Indeed, when r is small, the model can distinguish clusters in the dense area better, whereas when r is closer to 1, the clusters in the dense area are put together in a larger cluster. The same happens with the Diamond dataset in the **middle** of Fig.IV.5, where clusters are distributed according to two different scales. Finally, on the Grid dataset on the **right** part of Fig.IV.5, we see an optimum r exists to distinguish the clusters distributed on a grid; it makes sense since only one scale in clusters distribution is involved in this dataset. In Table. IV.1, we explicitly report the values of the PDP optimal r and compare them to the values yielded by the same model using either a Dirichlet Process (**DP**) prior or a Uniform Process (**UP**) prior.

Real-world data

In Table IV.1, we report the results for the 20Newsgroup (20-NG) real-world dataset, which is a collection of 18,000 user posts published on Usenet, organized in 20 Newsgroup (which are our target thematic clusters). As a model, we consider a modified version of LDA (Blei, Ng, and Jordan, 2003) that uses a PDP prior instead of DP in the words sampling step. Note that because the number of clusters must be provided to LDA, we do not compute $\frac{\mathbf{K_{inf}} - \mathbf{K_{true}}}{\mathbf{K_{true}}}$. We also run an additional experiment on three additional well-known datasets: Iris (4 attributes, 3 classes), Wines (13 attributes, 3 classes) and Cancer (30 attributes, 2 classes). We see PDP yields improved performances on every dataset.

We now illustrate the interest of using an alternate form of prior for the Infinite Gaussian Mixture model on real-world data. We consider a dataset of 4.300 roman sepulchral inscriptions comprising the substring "Antoni" that have been dated between 150AC and 200AC and assigned with map coordinates. The dates correspond to the reign of Antoninus Pius over the Roman empire. The dataset is available on Clauss-Slaby repository Clauss et al., 2021. It was common to give children or slaves the name of the emperor; the dataset gives a global idea of the main areas of the roman empire at that time (Hanson and Lobo, 2017). The task is to discover spatial clusters of individuals named after the emperor. We expect to find geographical clusters around: Italy, Egypt, Gauls, Judea, and all along the *limes* (borders of the roman empire, which concentrate lots of sepulchral inscriptions for war-related reasons) (Hanson, 2016). We present the results in Fig.IV.6.

We see that when $r = 1$, the classical DP prior is not fit for describing this dataset, as it misses most of the clusters. On the other hand, when $r = 0.9$, the infinite

Gaussian mixture model retrieves the expected clusters. It also makes some clusters that were not expected, such as the north Italian cluster or the long cluster going through Spain and France that corresponds to the layout of the Roman roads (via Augusta and via Agrippa; it was common to bury the dead on roads edges). Finally, when $r = 0.8$, we get even more detail: some of the main clusters are broken into smaller ones (Italy breaks into Rome, North Italy, and South Italy; Britain becomes an independent cluster, etc.). In this case, changing r controls the level of details of the clustering. Note that we do not compute metrics for this experiment in the absence of "ground-truth" clustering; there is no such thing as the right clustering in this case. Applied to spatial data, the PDP prior allows to control the clustering granularity. We see how different results can be according to the extent the model relies on the "rich-get-richer" prior and how its control is needed to make modelling relevant to a given situation.

IV.3.7 Conclusion

In this section, we discussed the necessity of controlling the "rich-get-richer" property that arises from the classical Chinese Restaurant Process usual formulation. We showed cases where this modelling hypothesis must be alleviated or strengthened to describe data more accurately. To this end, we derive the Powered Chinese Restaurant Process from a powered version of the Dirichlet-Multinomial distribution. This formulation allows reducing the expected number of clusters, which is not possible in the standard Pitman-Yor processes, while generalizing the standard Dirichlet process and the Uniform process.

The main feature of this formulation is that it allows for direct control of the "rich-get-richer" priors' importance. We derive elementary results on convergence and the expected number of clusters of the new process.

Finally, we show that it yields better results on synthetic data when coupled to a standard Gaussian mixture model and illustrates a possible use case with real-world data. For future works, it might be interesting to investigate cases where r takes non-positive values (which might lead to a "poor-get-richer" kind of process), and to develop a procedure to infer it automatically for specific problems (by minimizing a dispersion criterion for instance).

The regular Chinese Restaurant process has been used for decades as a prior in many real-world applications. However, alternate forms for this prior have been little explored. We are convinced that controlling the impact of the "rich-get-richer" hypothesis will bring significant changes in many state-of-the-art models.

We now propose to illustrate this claim and to further answer our task at hand. To this end, we use PDP to redefine the Dirichlet-Hawkes Process in terms of the Powered Dirichlet-Hawkes Process (PDHP) and investigate the new possibilities brought by this reformulation.

IV.4 Powered Dirichlet-Hawkes Process – Modelling self interacting clusters

This work has been published, see (Poux-Médard, Velcin, and Loudcher, 2021c)

IV.4.1 Introduction

IV.4.1.a PDHP as an answer to DHP's limits

In this section, we develop the Powered Dirichlet-Hawkes process (PDHP) as an answer to the limits of DHP raised in Section IV.2. It makes use of the redefinition of Dirichlet Processes detailed in Section IV.3.

We highlighted earlier that DHP has several limits. It fails to model data correctly when textual information is scarce (e.g., short texts such as tweets, or when topics' vocabulary overlap) (Yin et al., 2018) and when temporal information is scarce (overlapping dynamics, few observations). In addition, the usual assumption is that publication dynamics and textual content are perfectly correlated, which cannot be true in real-world applications. For instance, two identical textual contents will trigger different publication dynamics depending on who published them, or when they have been published.

IV.4.1.b Contributions

We overcome all these limitations by developing the Powered Dirichlet-Hawkes Process (PDHP), which yields better results than DHP on every dataset considered (up to +0.3 NMI). In particular, when textual or temporal information is scarce, PDHP allows to control the importance given to one or the other and thus retrieves better clusters. PDHP is the DHP equivalent of controlling the importance of the "rich-get-richer" hypothesis in DP using PDP, only using temporal information as a prior. PDHP also allows to distinguish *textual clusters* from *temporal clusters* (documents that follow the same dynamic independently from their content).

Our contributions are listed below:

- We highlight and explain the limitations of the DHP prior: it does not handle weakly informative temporal and textual information and it is not designed to consider different dynamics between text and time.
- We derive the Powered Dirichlet-Hawkes process (PDHP) as a new prior in Bayesian non-parametric for the temporal clustering of a stream of textual documents, which is a generalization of the Dirichlet-Hawkes process (DHP) and of the Uniform process (UP).
- We show how the PDHP prior performs better than DHP and UP priors through a thorough evaluation and comparison on several synthetic datasets and real-world datasets from Reddit.
- We show that PDHP prior allows choosing whether clusters are based more on textual or temporal information, or a mixture of both. We can favour their generation more or less according to documents' textual content or their publication dynamics.
- We perform a detailed analysis of PDHP's results on real-world datasets from Reddit. We illustrate our approach with examples of real-world topics uncovered by our method. We systematically analyse the influence of the hyperparameter r introduced in PDHP. Our method allows us to recover more or less bursty events from data streams depending on r.

IV.4.2 Model and algorithm

IV.4.2.a Dirichlet prior and alternatives

We briefly recall the definition of a Dirichlet prior (see Section IV.3 for an extensive discussion on DP). A Dirichlet prior for clustering implements the assumption that the more a cluster is populated, the more chances a new observation belongs to it ("rich-get-richer" property). Besides, there is still a chance that a new observation gets assigned to a newly created cluster. It is often expressed using a metaphor, the Chinese Restaurant Process (CRP), and it goes as follows: if an i^{th} client arrives in a Chinese restaurant, they will sit at one of the K already occupied tables with a probability proportional to the number of persons already sat at this table. They can also sit alone at a new table $K+1$ with a probability inversely proportional to the total number of clients in the restaurant. When their choice is made, the next client arrives, and the process is repeated. Let c be the cluster chosen by the i^{th} customer, $\vec{C^-}$ the table assignment of previous customers up to $i-1$, N_c the population of table c, C the number of already occupied tables and $\alpha_0 \in \mathbb{R}^+$ the concentration parameter. The process can be written formally as:

$$\text{CRP}(C_i = c|\vec{C^-}, \alpha_0) = \begin{cases} \frac{N_c}{\alpha_0+N} \text{ if c = 1, 2, ..., C} \\ \frac{\alpha_0}{\alpha_0+N} \text{ if c = C+1} \end{cases} \tag{IV.26}$$

The Uniform process (Wallach et al., 2010) has been proposed as an alternative to the DP prior. In this context, a new customer entering the restaurant has an identical chance to sit at either of the occupied tables, and a chance to sit at an empty table inversely proportional to the number of occupied tables. Formally:

$$\text{U-CRP}(C_i = c|\vec{C^-}, \alpha_0) = \begin{cases} \frac{1}{\alpha_0+C} \text{ if c = 1, 2, ..., C} \\ \frac{\alpha_0}{\alpha_0+C} \text{ if c = C+1} \end{cases} \tag{IV.27}$$

Finally, the Powered Dirichlet process that we detailed Section IV.3 generalizes the two processes above, stating that the probability for a new client to sit at a new table depends arbitrarily on the number of customers already sat at this table:

$$\text{P-CRP}(C_i = c|r, \vec{C^-}, \alpha_0) = \begin{cases} \frac{N_c^r}{\alpha_0+\sum_{c'} N_{c'}^r} \text{ if c = 1, 2, ..., C} \\ \frac{\alpha_0}{\alpha_0+\sum_{c'} N_{c'}^r} \text{ if c = C+1} \end{cases} \tag{IV.28}$$

where $r \in \mathbb{R}^+$ is a hyper-parameter. Varying r allows to give more or less importance to the "rich-get-richer" hypothesis of DP. Note that $P-CRP(r=0, \vec{C^-}, \alpha_0) = U-CRP(\vec{C^-}, \alpha_0)$ and that $P-CRP(r=1, \vec{C^-}, \alpha_0) = CRP(\vec{C^-}, \alpha_0)$. We will use this more general form in the rest of this section and make r vary to compare those priors in the experimental section.

IV.4.2.b Hawkes processes

The Hawkes process has already been introduced and discussed in Section IV.2.2.b. We recall that Hawkes processes are defined as self-stimulating temporal point processes. They are fully characterized by their intensity function $\lambda(t)$, which is related to the probability $P(t_{events} \in [t; t+\Delta t])$ of an event happening between t and $t+\Delta t$ by $\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t_{events} \in [t; t+\Delta t])}{\Delta t}$.

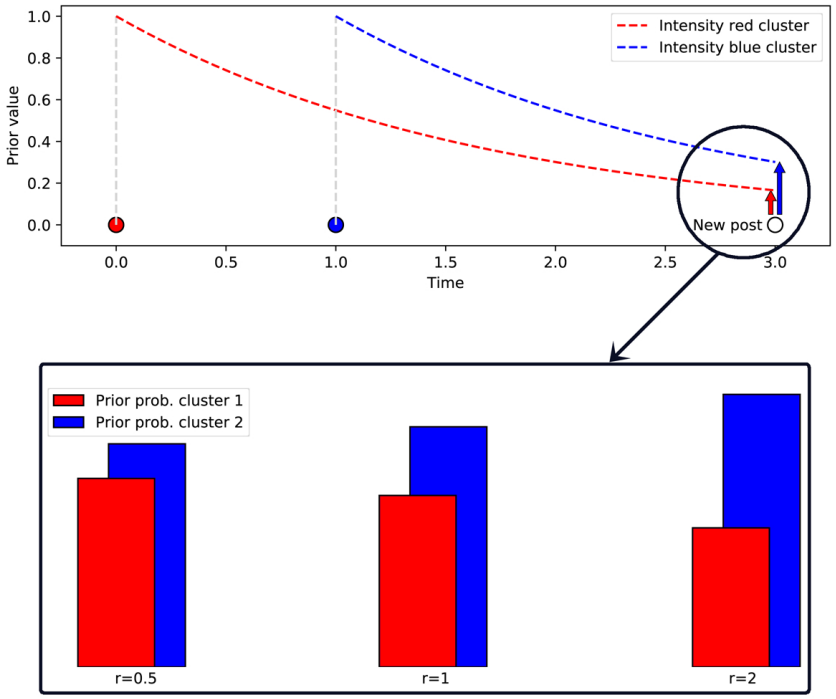

FIGURE IV.7: **Temporal prior on a new observation cluster — (Top)** Prior probability for two clusters to get chosen at all times according to DHP (dotted lines) **(Bottom)** Same prior probability for each cluster as a function of r in the PDHP at a given time.

Same as in (Du et al., 2015), the intensity of the Hawkes process associated with cluster c is defined as:

$$\lambda_c(t|\mathcal{H}_{<t,c}) = \sum_{\mathcal{H}_{<t,c}} \vec{\alpha_c}^T \cdot \vec{\kappa}(t_{i,c}) \tag{IV.29}$$

We recall that $\vec{\alpha_c}$ is a vector of weights and $\vec{\kappa}(t)$ is a vector of user-defined kernel functions with the same dimension as $\vec{\alpha}$. The kernel functions are defined at the beginning of the algorithm and are not modified afterwards. We will want to infer the weights vector $\vec{\alpha}$ to determine which entries of the kernel vector are the most relevant for a given situation.

IV.4.2.c Powered Dirichlet-Hawkes process

In (Du et al., 2015), the authors create DHP by substituting the counts N_k in the DP with the intensities of the Hawkes processes associated with each cluster. Instead, we simply substitute the counts in PDP with the intensities of the Hawkes processes associated with each cluster, forming the Powered Dirichlet-Hawkes Process (PDHP):

$$P(C_i = c|t_i, r, \lambda_0, \mathcal{H}_{<t_i,c}) = \begin{cases} \frac{\lambda_c^r(t_i)}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c}\leq\text{C} \\ \frac{\lambda_0}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c=C+1} \end{cases} \tag{IV.30}$$

where t_i is the arrival time of document i. We reformulate the Dirichlet-Hawkes process to allow nonlinear dependence (r) on the non-integer counts ($\vec{\lambda}$). We illustrate how r influences the prior probability of cluster selection in Fig. IV.7.

IV.4.2.d Textual modelling

To compare to DHP on solid ground, we consider the same textual model the authors used in the original paper, the Dirichlet-Multinomial model. This model is detailed in Section IV.2.2.d and it considers the textual content as a bag of words.

The likelihood of the i^{th} document belonging to cluster c reads:

$$\begin{aligned}\mathcal{L}(C_i = c|N_{<i,c}, n_i, \theta_0) &= P(n_i|C_i = c, N_{<i,c}, \theta_0) \\ &= \frac{\Gamma(N_c + \theta_0)}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\Gamma(N_{c,v} + \theta_0)}\end{aligned} \tag{IV.31}$$

where N_c is the total number of words in cluster c from observations previous to i, n_i is the total number of words in document i, $N_{c,v}$ the count of word v in cluster c, $n_{i,v}$ the count of word v in document i and $\theta_0 = \sum_v \theta_{0,v}$.

IV.4.2.e Posterior distribution

The resulting posterior distribution of the i^{th} document over clusters is calculated using Bayes theorem. It is proportional to the product of the textual likelihood (Eq.IV.31) and the temporal Powered Dirichlet-Hawkes prior (Eq.IV.30):

$$\begin{aligned}&P(C_i = c|r, n_i, t_i, N_c, \mathcal{H}_{<t,c}) \\ &\propto \underbrace{P(n_i|C_i = c, N_{<i,c}, \theta_0)}_{\text{Textual likelihood}} \underbrace{P(C_i = c|t_i, r, \lambda_0, \mathcal{H}_{<t_i,c})}_{\text{Temporal prior}} \\ &= \frac{\Gamma(N_c + \theta_0)}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\Gamma(N_{c,v} + \theta_0)} \\ &\quad \times \begin{cases} \frac{\lambda_c^r(t_i)}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c = 1, ..., C} \\ \frac{\lambda_0}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c = C+1} \end{cases}\end{aligned} \tag{IV.32}$$

We recall that $\lambda_c(t)$ is defined Eq. IV.29. The textual likelihood to open a new cluster $C + 1$ is computed by setting $N_{C+1,v} = 0$ – because it is empty before being opened.

IV.4.2.f Algorithm for parameters inference

We use a similar algorithm to the one in (Du et al., 2015). Briefly, the algorithm is a sequential Monte-Carlo (SMC) that takes one document at a time in their order of arrival. The algorithm starts with a number N_{part} of particles whose weights are $\omega_p = \frac{1}{N_{part}}$, each of which will keep track of a hypothesis on documents clusters. After a few iterations, particles that contained unlikely allocation hypotheses are discarded and replaced by more likely ones. The likeliness of a hypothesis is encoded in the weights of each particle ω_p. Such genetic algorithms are favoured for optimizing DHP-based models. These models are not convex, and genetic algorithms allow to tackle the problem sequentially.

For each particle, when a new document arrives, (1) the cluster of the document is sampled according to a Categorical distribution over all clusters, whose weights are determined by Eq. IV.32. After the cluster of the new document has been sampled, (2) the kernel weights $\vec{\alpha}$ from Eq. IV.29 are updated using Eq. IV.5. For efficiency purpose, we sample $\vec{\alpha}$ from a set of N_s pre-computed $\vec{\alpha}$ vectors. We finally (3) update the weights ω_p of each particle according to the posterior Eq. IV.32 such as $\omega_p^{(n+1)} = \omega_p^{(n)} \times$ Eq. IV.32. If the weight of a particle falls below a value ω_{thres}, the

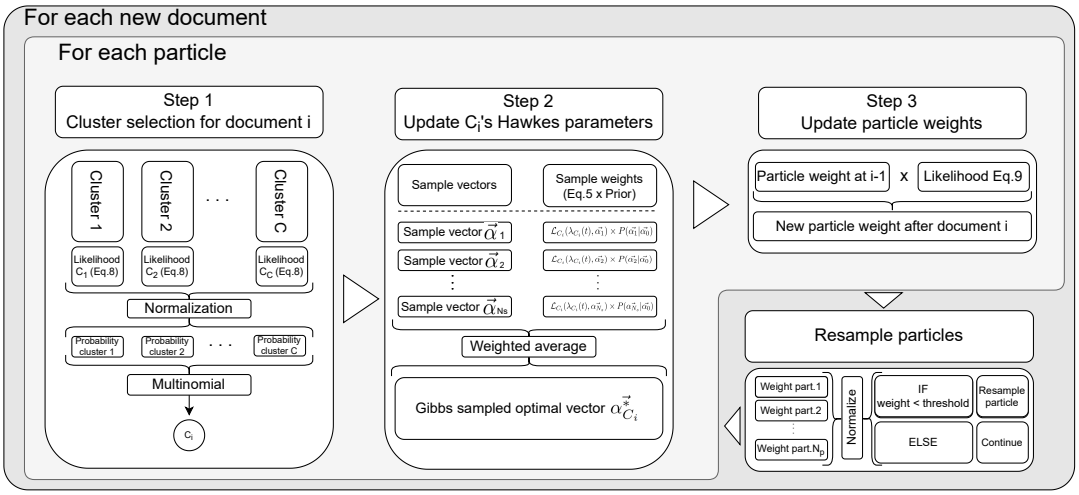

FIGURE IV.8: **Schematic workflow of the SMC algorithm** — For each new observation from a stream of document, we run steps 1 (sample document's cluster), 2 (update sampled cluster's internal dynamics) and 3 (update particle hypothesis' likeliness) for each particle, and then discard particles containing the less likely hypothesis on cluster allocation.

particle is discarded and replaced by another existing one with sufficient weight. The full process is illustrated in Fig. IV.8. By updating incrementally the likelihood associated with each of the pre-computed $\vec{\alpha}$ sample vectors, the algorithm processes each new observation in constant time $\mathcal{O}(1)$.

The task of updating kernel coefficients (2) is the same as in any Hawkes process, and the task of updating particle weights and resampling them (3) is common to any SMC algorithm. The change induced by the PDHP compared to the DHP happens in step (1). First of all, we note that for $r = 1$ the PDHP prior is identical to the DHP prior. From Section IV.3, lowering the value of r reduces the "rich-get-richer" aspect of the PDP ("rich-get-less-richer"), whereas increasing it leads to a "rich-get-more-richer" effect. These metaphors can be translated as follows in our temporal context: for lower values of r, the relative difference between cluster's temporal intensities plays a less significant role in cluster selection, whereas higher values of r tend to exacerbate these differences and make the temporal aspect of the greatest consequence on the choice of a cluster. In other words, tuning the value of r allows giving more or less importance to the temporal aspect of the clustering. This is illustrated in Fig. IV.9.

In Fig. IV.7, we plot the situation when a new observation gets assigned a cluster. The associated Hawkes intensities are the base to compute the prior probability for either cluster. This quantity is then modulated using r to give more or less importance to intensity differences between clusters. In Fig. IV.9, we plot the probability for various clusters to be chosen (which is directly proportional to the posterior distribution, see Eq. IV.32) according to r when their textual likelihood and Hawkes process intensity are known. Note that for $r = 0$, the probability for any cluster to get chosen is linearly proportional to its textual likelihood (Dirichlet-Uniform process), whereas when r increases, the probability of getting chosen gets closer to a selection based on the temporal aspect only.

This makes the main interest of the PDHP model. Tuning the parameter r allows choosing whether inferred clusters are based on textual or temporal considerations. It generalizes several state-of-the-art works, which are special cases of the PDHP for different values of r. The DHP (Du et al., 2015) is equivalent to PDHP for $r = 1$;

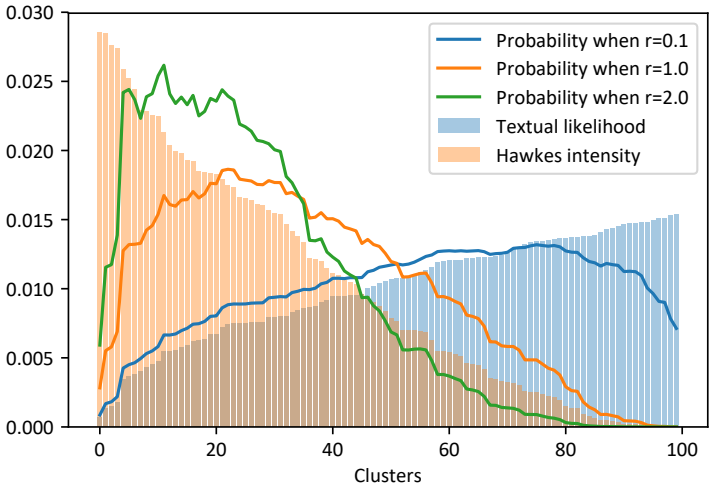

FIGURE IV.9: **Effect of r on cluster selection probabilities** — The probability for each cluster to get chosen (solid lines) for several values of r and fixed individual textual likelihood (blue bars) and Hawkes intensity (orange bars).

the UP (Wallach et al., 2010) is equivalent to PDHP when $r = 0$. In the following sections, we show how fine-tuning r systematically yields significantly better results than setting it to $r = 0$ or $r = 1$ (up to a gain of 0.3 on our experiments' normalized mutual information metric). We also show how varying it allows to recover one kind of clustering or the other (textual or temporal) with high accuracy and see how it affects clustering results on several real-world datasets.

IV.4.3 Experiments

IV.4.3.a Synthetic data

Synthetic data generation

We simulate a case where only two clusters are considered. Each cluster has its own vocabulary distribution over 1,000 words and its own kernel weights $\vec{\alpha}$, with Gaussian Hawkes kernel functions $\kappa(\vec{t})$ of parameters (μ, σ)=(3, 0.5), (7, 0.5) and (11, 0.5) (see Eq. IV.29). Finally, we set $\lambda_0 = 0.05$. We first simulate one independent Hawkes process per cluster using the Tick Python library (Bacry et al., 2017). The processes are stopped at time $t = 1500$, which makes a rough average of 7,000 events per run. Then we associate each simulated observation with a sample of 20 words drawn from the corresponding cluster's word distribution. The inference has been performed using an 8 core processor (i7-7700HQ) with 8GB of RAM on a laptop, which underlines how scalable the algorithm is. As stated before, the algorithm processes each new document in constant time $\mathcal{O}(1)$, which ranged from 0.05s on synthetic data to maximum 1s on real-world data. Note that this number is directly proportional to the number of active inferred clusters, and thus depends strongly on the dataset.

We generate ten such datasets for every considered value of vocabulary overlap and Hawkes intensities overlap, which leave us with 200 datasets (5 values of textual overlap x 4 values of temporal overlap x 10 datasets). Overlap is defined as the shared area of two distributions, normalized by the total area under the distributions. For instance, if the vocabulary of one cluster ranges from words "1" to "100"

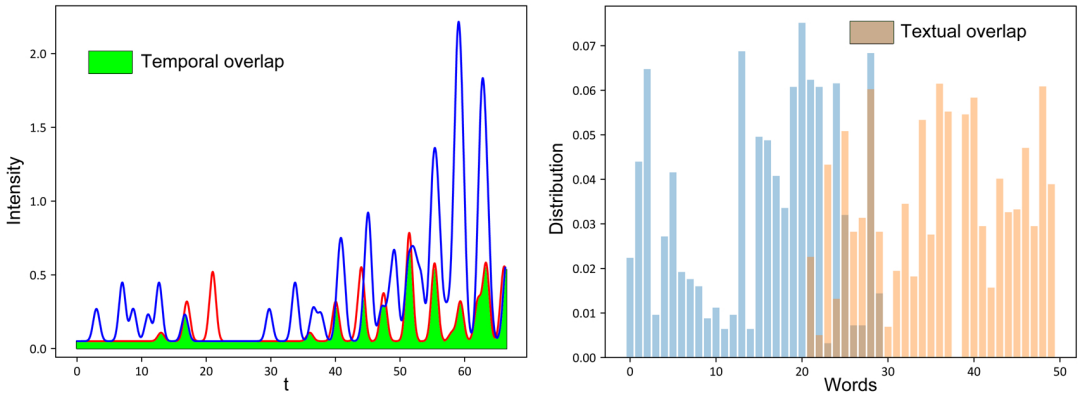

FIGURE IV.10: **Overlaps** — (Left) Temporal overlap is defined as the ratio between the area common to two Hawkes intensities and the total area under the intensity functions. (Right) Textual overlap is defined as the proportion of vocabulary that is common to two clusters, weighted by the probability of words within their respective cluster.

with uniform distribution, and the vocabulary of another cluster from words "50" to "150" with uniform distribution, the overlap equals 50%. We define the overlap of Hawkes process intensity in the same way. If the triggering Hawkes kernel of one cluster is a Gaussian function with $(\mu, \sigma) = (3, 1)$ and one associated observation at $t = 0$, and the triggering kernel of the other is also a Gaussian function but with $(\mu, \sigma) = (5, 1)$ also with an associated observation at $t = 0$, the overlap equals 32% (see Fig. IV.10). When computing the Hawkes intensity overlap, every observation within a cluster and its associated timestamp are considered. The definition of overlaps is illustrated in Fig. IV.10. To enforce a given vocabulary overlap (Fig. IV.10-right), we shift the word distributions of the clusters from which events' vocabulary is sampled. To enforce a given Hawkes intensities overlap (Fig. IV.10-left), we shift the event times of every event in one of the clusters until we get the correct overlap ($\pm 5\%$).

Note that we consider ten different datasets instead of considering ten runs per dataset for two reasons. Firstly, the generation of Hawkes processes is highly stochastic, so a model might perform significantly better on a single dataset only by chance. Secondly, given the way the SMC algorithm works, the standard deviation between runs is small: at each iteration, N_{part} clustering hypotheses are tested, which is equivalent to running N_{part} times a single clustering algorithm. We heuristically set $N_{part} = 8$, as we observe no significant improvement using more particles.

The other parameters we use for clustering synthetic data are: $\alpha_0 = 0.1$, $\theta_0 = 1$, $\vec{\kappa(t)} = [\mathcal{G}(t; 3, 0.5), \mathcal{G}(t; 7, 0.5), \mathcal{G}(t; 11, 0.5)]$ with $\mathcal{G}(t; \mu, \sigma)$ the Gaussian function, $N_{samples} = 2.000$ and $\omega_{thres} = \frac{1}{2N_{part}}$.

We are interested in varying both vocabulary and intensities overlap to exhibit the limits of DHP and how PDHP overcomes them. Note that in the synthetic data experiments in (Du et al., 2015) (Figs.3a and 3b), the intensities overlap is almost null, which makes the task easier for the Hawkes part of the algorithm. The primary metric we use throughout the experimental section is the normalized mutual information (NMI). During the experiments, we also considered the Adjusted normalized rand index and the V-measure, which are adapted to evaluate clustering results when the number of inferred clusters is different from the true number of clusters. The observed trends in results from these other metrics are identical to the ones observed for NMI. Therefore, we choose to report only the results of the latter for clarity.

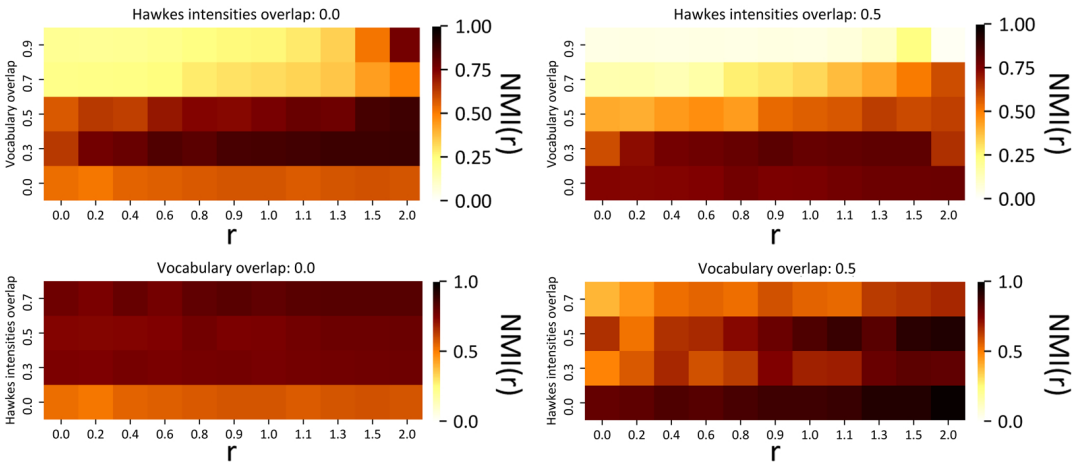

FIGURE IV.11: **PDHP yields good NMI values** — Normalized mutual information (NMI) for various values of r, intensities overlap and vocabulary overlap, for one dataset per combination. The results for $r = 0$ are the output of the Uniform process, the results for $r = 1$ are the output of the DHP (Du et al., 2015), and the other values of r correspond to other particular cases of PDHP. The darker the better. Overall, PDHP yields good NMI values (the maximum being 1).

The NMI metric is standard when evaluating non-parametric clustering models. It compares two cluster partitions (i.e., the inferred and the ground truth ones); it is bounded between 0 (each true cluster is represented to the same extent in each of the inferred ones) and 1 (each inferred partition comprises 100% of one true cluster).

PDHP yields better results as vocabulary overlap increases

We report our results when the intensities overlap is null, with varying r and the vocabulary overlap in Fig. IV.12a. Because we consider ten different datasets for each set of overlap parameters, it makes no sense to report the absolute average NMI since it can vary greatly from one dataset to the other. Instead, we plot the relative NMI difference between PDHP and DHP ($r = 1$), which we expect to be less dependent on the datasets we consider. However, to give an idea of the typical performance for some parameters, we also provide raw results for one run in Fig. IV.11.

There is a clear correlation between efficiency, vocabulary overlap and r, with a gain on NMI up to $+30\%$ of its maximal value over DHP. As stated at the end of the "Model" section, this result was expected: the more vocabulary overlap grows, the less textual content carries valuable information to cluster the documents. This observation supports the concerns raised in (Yin et al., 2018) about the efficiency of DHP for clustering short text documents. However, Hawkes intensities overlap being null, the arrival time of events carries highly valuable information when textual content does not allow to distinguish clusters well. Therefore, PDHP provides a way to tackle the problem raised in (Yin et al., 2018) without the need to sample observations.

Conversely, when vocabulary overlap is null, the textual content provides enough information to distinguish clusters correctly. The temporal dimension only allows refining the results with no significant improvement for all values of r.

Finally, we can see how the Dirichlet-Uniform process (DUP, $r = 0$) consistently yields worse performances under these settings. Once again this is expected, since in this synthetic experiment intensities overlap carries valuable information about

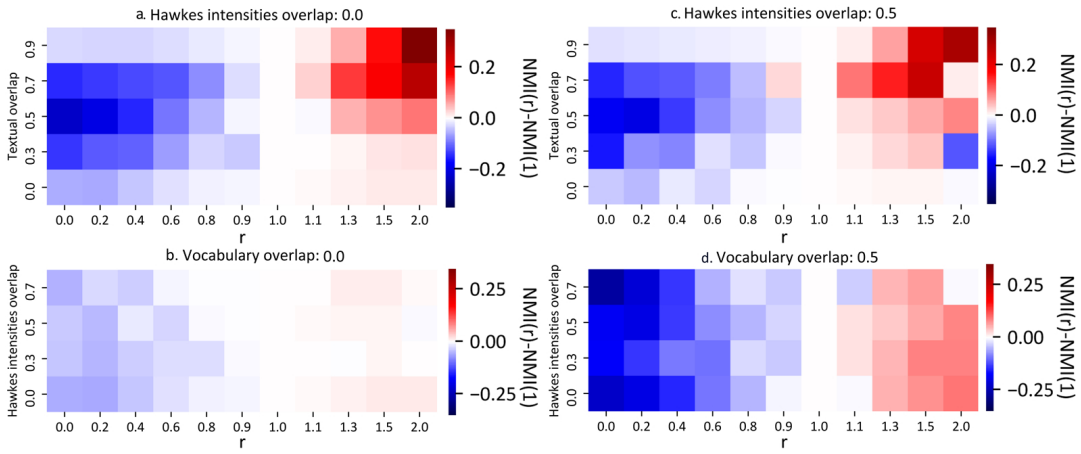

FIGURE IV.12: **PDHP performs better than DHP** — Difference between the normalized mutual information (NMI) of PDHP and DHP model (Du et al., 2015) for various values of r, intensities overlap and vocabulary overlap, averaged over all the datasets. Red means PDHP performed better, blue means PDHP performed less well. Because PDHP($r = 1$)=DHP, the column $r = 1$ show no difference. PDHP allows to increase results on NMI by as much as 0.3 over DHP.

events clustering; DUP only considers textual information and therefore misses valuable clues.

PDHP yields similar results for null vocabulary overlap

We report comparable results in Fig. IV.12b. Here, we consider a null vocabulary overlap for various values of r and of Hawkes intensities overlap. The situation is now the opposite: the textual content always carries valuable information about clusters, whereas the temporal aspect does not. We observe the same trend as in Fig. IV.12a –note that the colour scale is the same. Varying the value of r does not significantly change the performance of clustering, meaning the textual content always carries enough information. This plot shows that PDHP can handle greater intensities overlap without collapsing into unrealistic clustering. Since in most real-case applications, many clusters with various dynamics may coexist simultaneously, it is comforting that the PDHP can also handle this case.

PDHP yields better results in more realistic situations

We finally report the results for intermediate values of intensities and vocabulary overlaps in Fig. IV.12c,d. In real-world applications, it seldom happens that topics' vocabularies do not overlap at all. For instance, a quick analysis of *The Gutenberg Webster's Unabridged Dictionary by Project Gutenberg* shows that 22% of English words are associated with more than one definition. A more detailed analysis would need to consider the usage frequency of words to get correct statistics. Still, this number provides an estimate of the effective vocabulary overlap in real-world situations.

In Fig. IV.12c, we present the results for a fixed intensities overlap of 0.5 versus various values of r and vocabulary overlaps, and in Fig. IV.12d for a fixed vocabulary overlap of 0.5 versus various values of r and intensities overlaps. Once again, we see that, on average, using PDHP can increase the NMI over DHP up to +20% of the maximum possible value.

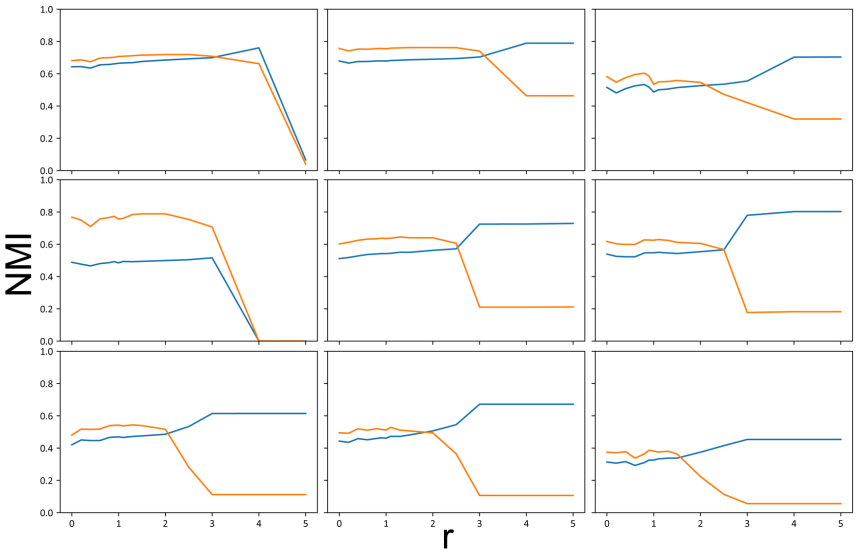

FIGURE IV.13: **Textual (orange) and temporal (blue) NMI vs r when textual and temporal clusters are decorrelated** — From top-left to bottom-right, there are 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90% of generated events that have been randomly re-assigned a textual cluster. The orange curves are the textual NMI vs r, which evaluate how well events whose vocabulary has been sampled from the same distribution are clustered together; the blue curves are the temporal NMI vs r, which evaluate how well events following the same temporal dynamic are correctly clustered together. The values presented are for one dataset. We see that varying r allows retrieving the right temporal (r large) or textual clusters (r small).

PDHP finds textual or temporal clusters depending on r

We now slightly modify our experimental setup. Instead of considering that textual clusters and Hawkes intensities are perfectly correlated, we consider a decorrelated case. A document whose vocabulary is drawn from cluster C_1 can now follow the same temporal dynamics as cluster C_2. If we imagine a dataset of news articles published online, it is clear why this might happen frequently. If popular newspapers such as New York Times or Reuters publish an article on a topic A at a time t, it is likely to trigger snowball publications of related articles from less popular journals. "Popularity" is chosen as an indicator in this example, but it may be any other external parameter (centrality in news networks, support of publications, etc.). In this case, the article's textual content allows to uncover a "story of publication", that is, how the article has been spread, when publication spikes are, etc. However, the temporal information would help understand the dynamics of publications interaction: which reduced set of articles triggered the publication of subsequent ones.

In (Du et al., 2015), it is assumed that every document within clusters follows a unique dynamic. We relax this hypothesis in our datasets as follows. For null textual and temporal overlaps, after a dataset has been generated, we resample the textual clusters of a fraction of randomly selected events, as well as the words associated with the event. In doing so, we decorrelate temporal and textual clusters. Therefore, an event is now described by two cluster indicators: its temporal cluster (which Hawkes intensity made the event appear where it is) and a textual cluster (which vocabulary has been used to sample the words the event contains).

For completeness, we also show the results for various decorrelations for one run in Fig. IV.13. To better understand the tendency of NMIs with respect to r, we plot the average difference between the NMI of textual clustering and the NMI temporal

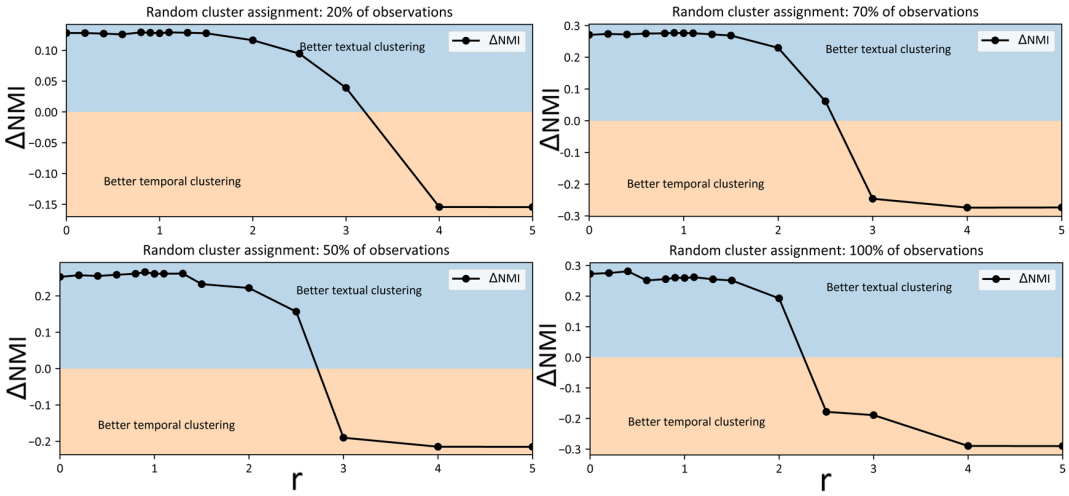

FIGURE IV.14: **Varying r allows to choose between textual or temporal clustering** — The black line plots the difference between the NMI of textual and temporal clustering. For small r, textual clustering is far better than temporal clustering, and for large r, the situation is reversed. This is because r determines the importance given to the temporal dimension and therefore allows choosing between retrieving temporal or textual clusters.

clustering over all the datasets. Explicitly: $\Delta NMI = NMI_{text} - NMI_{temp}$. The results are reported in Fig. IV.14.

As supposed at the end of the "Model" section, varying r allows retrieving one clustering or the other. Note that the value r of transition from text to time clustering depends directly on the dataset considered: number of words sampled, vocabulary size, overlaps, etc.

PDHP efficiently infers the temporal dynamics of each cluster

Finally, we show that PDHP correctly infers kernels' parameters in every situation where events are correctly assigned to their temporal cluster. The results are reported in Fig. IV.15. We looked at the mean absolute error (MAE) and the mean Jensen-Shannon divergence (MJS) between the vector $\vec{\alpha}$ used to generate the dataset and the inferred one. We note in Fig. IV.15 that when the textual overlap is small, the inferred kernel is close to the real one and r has a minor impact on the result. This is because the inferred kernels mostly depend on the correctness of inferred clusters: when observations are allocated to the right clusters, the Hawkes process inference considers relevant information when inferring these clusters' dynamics. However, when observations are misallocated, the Hawkes process infers dynamics also based on times that are not supposed to contribute to this cluster's dynamic. When the clustering task is simple and yields good results (that is, when the textual overlap is small, see Fig. IV.11), the PDHP infers correct temporal dynamics ($\sim$ 5% MAE); this shows our method correctly accounts for clusters dynamics given the available information.

When vocabulary overlap is large, the value of r significantly influences the kernel inference performances. However, when r is chosen so that clusters are correctly inferred, the kernel inference retrieves well the expected kernels ($\sim$ 5% MAE). Finally, the temporal kernel inference is expected to be less precise when temporal overlap increases, which is what happens in Fig. IV.15-bottom-right. In this case, the model does not retrieve well the synthetic kernels even for the optimal r. Besides,

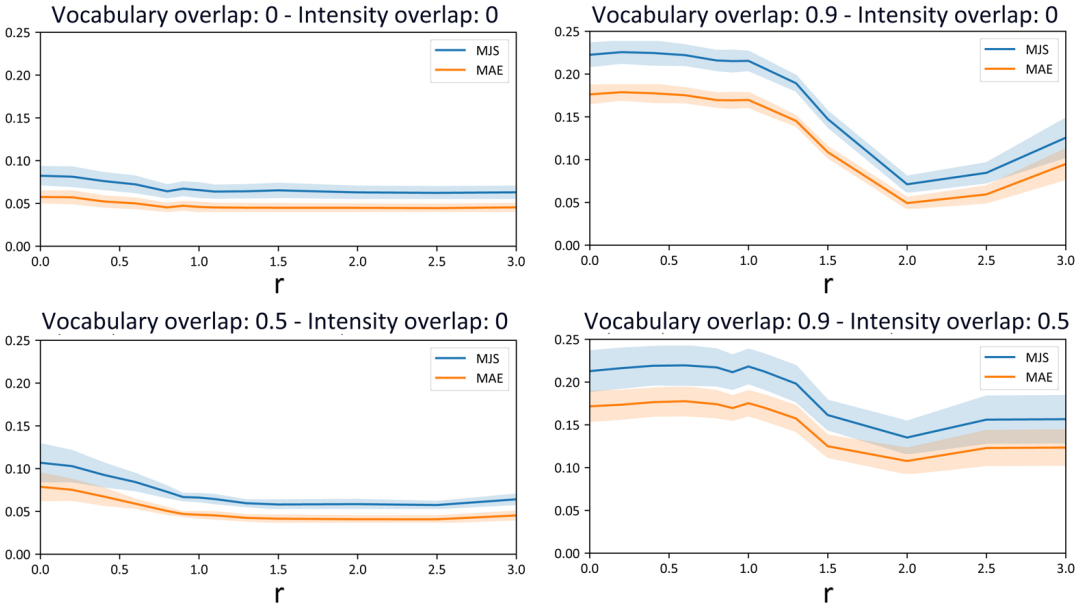

FIGURE IV.15: **Varying r allows to better capture the dynamics at stake** — We plot the mean average error and the mean Jensen-Shannon divergence of the inferred kernel function α with respect to the kernel used to generate the data, for various values of temporal and textual overlaps. The standard error bars are computed over 10 independent runs. The higher the temporal overlap, the larger the error bars; the larger the textual overlap the more influence has r.

the error bars get wider as a consequence of the cluster allocation being more challenging. Overall, provided the right clusters, we conclude that our method correctly retrieves the inferred temporal kernels.

IV.4.3.b Real-world application on Reddit

We use the PDHP prior to model real streams of textual documents. We consider three Reddit datasets Baumgartner et al., 2020 about different topics. The **News dataset** is made of 73.000 titles extracted from the subreddits inthenews, neutralnews, news, nottheonion, offbeat, open_news, qualitynews, truenews and worldnews, from April 2019. We chose this month because of the wide variety of events that happened then (for instance, Sri Lanka Easter bombings, Julian Assange arrest, first direct picture of a black hole, Notre-Dame cathedral fire). We also consider 15.000 post titles of the subreddit TodayILearned (**TIL dataset**) and 13.000 post titles of the subreddit AskScience (**AskScience dataset**) on January 2019. We extracted the nouns, verbs, adjectives, and symbols from the textual data. We run the experiments using the following parameters: $\alpha_0 = 0.5$, $\theta_0 = 0.01$, $N_{samples} = 2.000$, $N_{part} = 8$ and $\omega_{thres} = \frac{1}{2N_{part}}$. The kernel vector $\vec{\kappa}$ is chosen as in (Du et al., 2015). It is made of Gaussian functions, with means located at 0.5, 1, 4, 8, 12, 24, 48, 72, 96, 120, 144 and 168 hours. The variance of each is set to 1, 1, 3, 8, 12, 12, 24, 24, 24, 24, 24 and 24 hours. The shape of the kernel is chosen so that it can account for a dynamic that can occur at different timescales. The algorithm will then infer the weights $\vec{\alpha}$ associated with each entry of the kernel vector $\vec{\kappa}$ for each cluster.

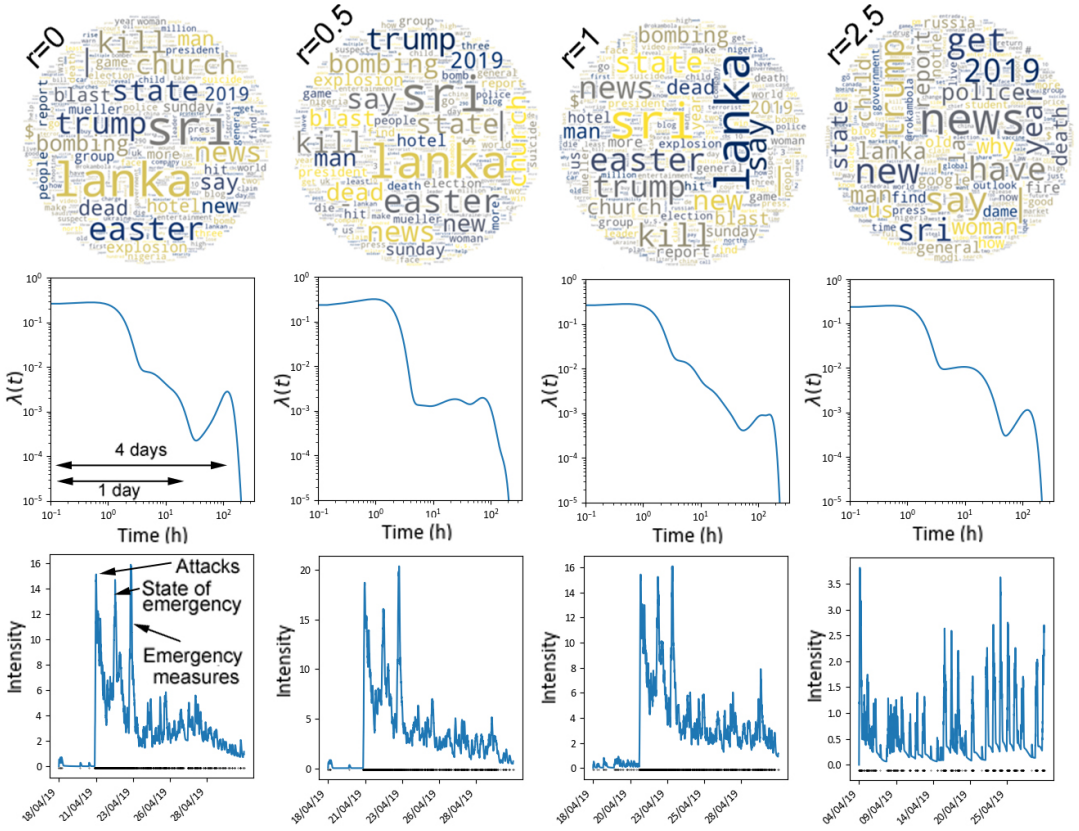

FIGURE IV.16: Wordclouds, triggering kernels and intensities for clusters the most closely related to Sri Lanka 2019 bombings for various values of r. The points at the bottom of the intensity plots are individual publication events. Note that triggering kernels are plotted on a log-log scale for visualisation purposes because most of the intensity is focused on small times: dynamics are bursty.

PDHP recovers meaningful stories

As an illustrative example, we consider the inferred clusters the most closely related to the Sri Lanka Easter bombings of April 2019 in Fig. IV.16. The main bursts in the news related to this event happened on the 21^{st}, 22^{nd} and 23^{rd} of January, and respectively correspond to the bombings themselves, the declaration of the state of emergency, and finally their application on the 23^{rd}. We plot the temporal kernels associated with this event on a log-log scale because most of the intensity is focused on small times: dynamics of information spread are bursty (Karsai, Jo, and Kaski, 2018). We see that inferred dynamics change with r as well as the cluster's vocabulary, which is expected since clusters do not contain the same documents. For $r = 0$, the Uniform process infers clusters based on textual information only; the triggering kernel is inferred afterwards. For $r = 2.5$ on the contrary, clusters are formed based on the triggering kernel, and textual information follows; we see from the right-plot that this cluster captures publications exhibiting a daily intensity cycle; this is visible both in the intensity plot (the bump around 2.10^2h which is not present on other temporal kernels) and in the real-time axis where publications seem to be packed around specific points in time roughly corresponding to a daily cycle. Given the intensity spikes on 21^{st}, 22^{nd}, and 23^{rd}, it is not surprising that articles about the Sri Lanka bombings are also part of this cluster. Note that the more r increases, the more intense the triggering kernel is around 24h. We see from Fig. IV.16 that DHP is a specific case of our modelling, and that tweaking the r parameter allows us to retrieve completely different results.

PDHP favours temporal or textual clustering depending on r

We report the values of log-likelihoods for every dataset and various values of r in Fig. IV.17. The textual likelihood is defined in Eq.IV.7, and the likelihood of a Hawkes process is defined in Eq.IV.5. Note that r does not appear in either Eq.IV.7 or Eq.IV.5; the plot in Fig. IV.17 thus only reflects the relevancy of the proposed textual modelling or temporal modelling independently from PDHP. Those likelihoods evaluate how well the textual or temporal aspect of the dataset is modelled with no consideration of the model being used. As expected from the synthetic experiments, varying r makes the model more sensitive to either textual or temporal data –note the similarity to Fig. IV.13. A low r favours the textual information clustering and is thus better at modelling documents' textual content, whereas a high r favours temporal information which makes PDHP better at capturing the publication dynamics.

PDHP infers sharper textual clusters for low r

We evaluate how meaningful textual clusters are using an entropy measure. We assume that a cluster is meaningful when it contains a reduced set of words; a cluster talking about one topic only is more likely to have a smaller vocabulary than a cluster about two or more topics. A way to measure this is to see how spread the vocabulary of a cluster is using Shannon entropy. Let $N_{c,v}$ be the count of word v in cluster c. The normalized Shannon entropy of a cluster c is defined as:

$$S(\vec{N}_c) = \frac{1}{-\log(V)} \sum_{v}^{V} \log\left(\frac{N_{c,v}}{\sum_{v}' N_{c,v'}}\right) \frac{N_{c,v}}{\sum_{v}' N_{c,v'}} \tag{IV.33}$$

An entropy of 0 means the vocabulary of the cluster is concentrated on a single word among the V possible words in the vocabulary; an entropy of 1 means that

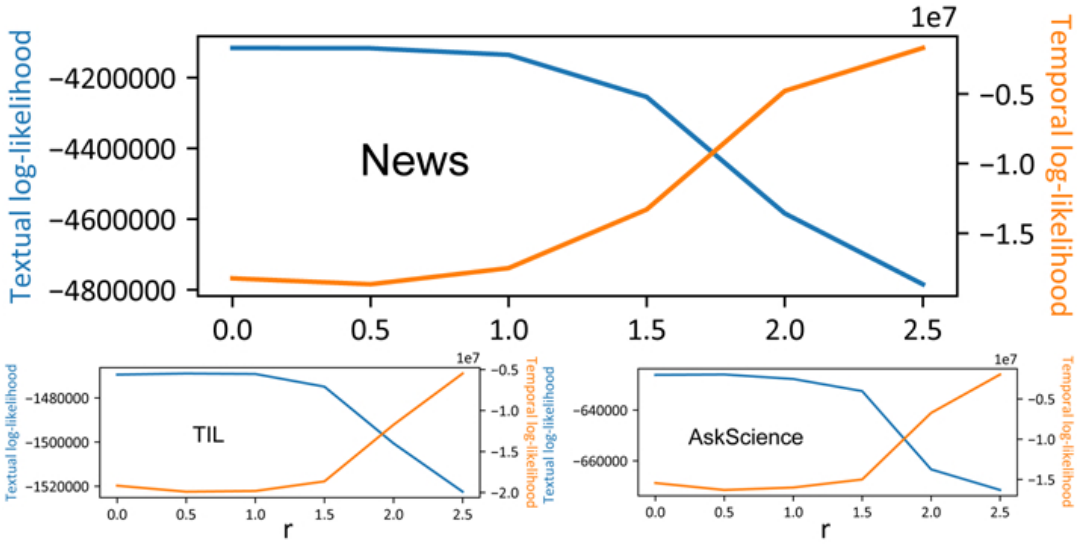

FIGURE IV.17: ***r* allows to favour text-based of time-based clustering on real world datasets** — Textual likelihood and Hawkes process likelihood for various values of r. The lower r the higher textual likelihood is, and the higher r the higher Hawkes process likelihood is.

every of the V words is present to the same extent with probability $\frac{1}{V}$. In Fig. IV.18, we plot the mean entropy for various values of r for all the datasets, along with the standard error over the clusters. The results show that on average vocabulary is more concentrated within clusters for low values of r. The inflection point of the curves corresponds to what has been previously observed with likelihoods in Fig. IV.17 and Fig. IV.14. On the contrary, higher values of r lead to clusters that comprise a less concentrated vocabulary. This is expected because as r increases, the textual information is no longer the most relevant data for cluster formation.

PDHP controls the burstiness

In Fig. IV.19, we plot the intensity function associated with the News dataset on the real-time axis for several clusters for $r = 0.5$ and $r = 1.5$. Note that not each of the ∼300 inferred clusters are represented, but instead we consider only the ones whose intensity went above 10 at least once, the rest being considered as noise. First of all, both values of r allow to recover the major events of April 2019 (in order of appearance): the first direct picture of a black hole (10/04), the arrest of Julian Assange (11/04), the fire of Notre-Dame de Paris Cathedral (15/04), the release of the Mueller report on Donald Trump (18/04) and the Sri Lanka Easter bombings (21/04). The top 5 words of every cluster are reported in the legend.

When r increases, PDHP retrieves new clusters associated with shorter bursty events. For instance, the cluster associated with the release of a new episode of Equestria Girls that went unnoticed for $r = 0.5$ has been detected with $r = 1.5$. This happens because the episode has not been discussed over a long period, and associated articles have a vocabulary significantly overlapping with other clusters' one. A model relying mostly on textual information might not detect specific words (twilight, Equestria, sparkle, etc.). If detected and a new cluster is created, it might then fail to associate subsequent events to this new cluster if temporal information plays a lesser role. On the other side, a model favouring temporal information is

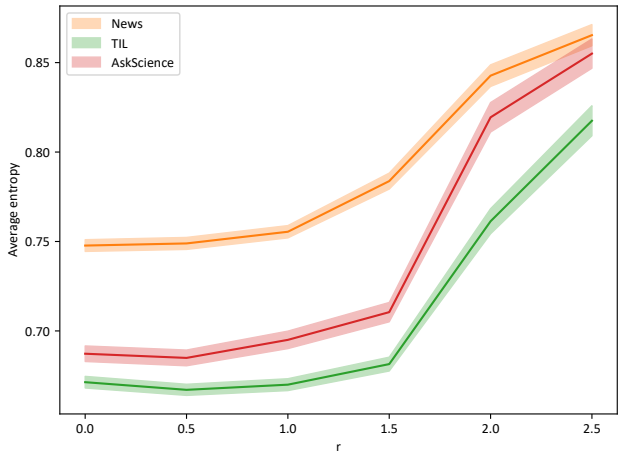

FIGURE IV.18: **Textual clusters are more informative for low values of** r — Weighted average entropy of words distribution for every dataset. Weights corresponds to the number of words within clusters. The error bar represents the standard error over all the clusters.

much more likely to associate subsequent events to a new cluster despite textual information fitting well older and more populated clusters.

This can be seen in Fig. IV.16, where the intensity of a kernel peaks at short times. This results in encouraging the burstiness. When r is large, a given event is likely to be associated with subsequent ones even when the associated vocabularies are only vaguely similar. On the other side, when r is small, older events with closer vocabularies have more chances of getting associated with it despite their intensity not peaking at the new event time.

Recovering publication cycles

The limit case of encouraging events burstiness is the deterministic allocation of documents to a cluster based only on their relative positions on the time axis. This is achieved when r is large. In this case, textual information does not matter and only regularities in the time distributions are detected. We illustrate such a case on the News dataset in Fig. IV.20.

In Fig. IV.20, we plot on the left the intensity associated with the events for each cluster on the real-time axis for $r = 2$. We see that the two most populated clusters follow precise dynamics. We added on the right side of the plot the temporal kernel corresponding to each of these clusters. On the right plot, we retrieve the cause of the daily and weekly cycles observed for the largest cluster on the left plot. The second most populated cluster follows similar dynamics, except that it seems to be shifted by half a day on the real-time axis; the peaks are in phase opposition with the largest cluster. It is worth noting that the Notre-Dame fire cluster is still detected; this is due to its vocabulary being different enough from the existing cluster's ones to trigger its own cluster, and the associated number of documents being consequent in a short time window. Interestingly, immediately after this cluster emerged, the dynamics on the real-time axis also follow a decaying circadian circle over three days.

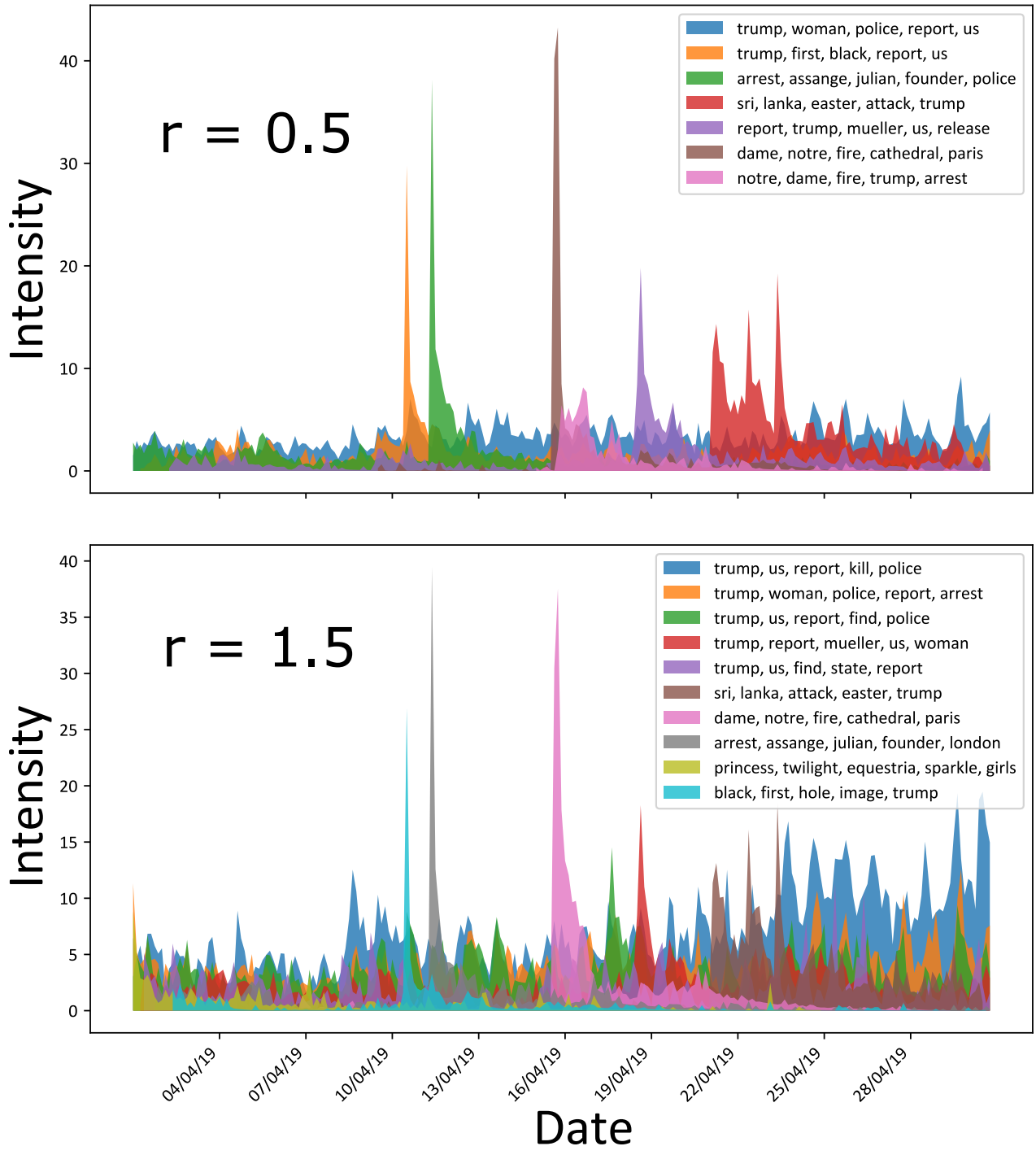

FIGURE IV.19: **PDHP allows for modelling bursty events** — PDHP's intensity for the News dataset for two values of r. A lower r finds globally relevant clusters, whereas a higher r allows to recover shorter bursty events.

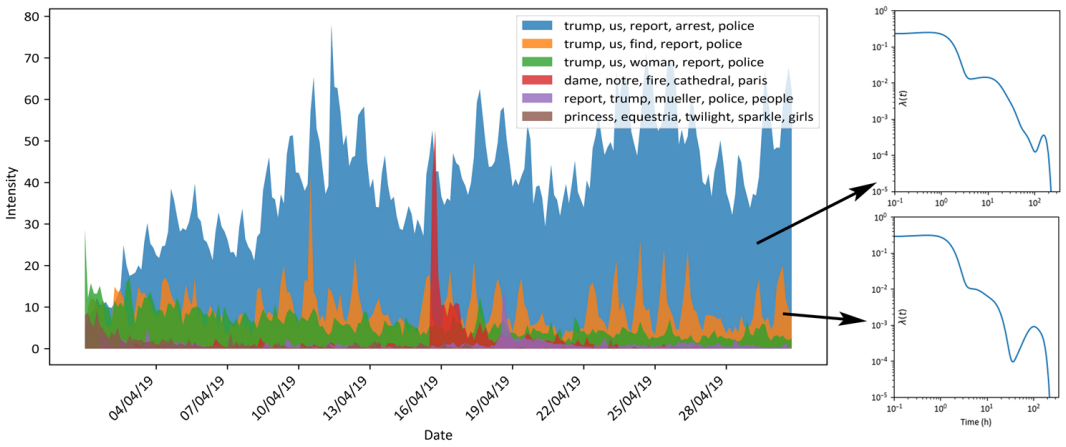

FIGURE IV.20: **Limit case of encouraging bursty events clustering** — We plot PDHP's intensity for the News dataset over the observation period for a large value of r. In this case, textual information plays a marginal role, and clusters are inferred based on the events publication dates only.

Heuristics

Choosing r — We saw that in all the previous experiments, the optimal r got determined from a grid-search-like strategy. We did not come up with a way to automatically infer the optimal r without trying several values.

However, we provide some heuristics regarding the tuning of r.

- As r increases, we usually get a smaller number of inferred clusters. This is because considering the temporal dimension adds consistency to the language model; the temporal intensity prior for a new observation is likely to be non-null, which increases the probability of *not* opening a new cluster with respect to a model that does not consider time.

- As r goes to infinity, we only infer one large cluster that comprises all the observations. This is because even the slightest difference in the prior intensities leads to a deterministic cluster choice.

Our leads to automatically determine r involve computing an ad-hoc objective metric to optimize jointly with the likelihood. Given there is no gold standard for clustering in real-world processes, the choice of r, and therefore the choice of such metric, is left to the user. As we showed in Fig. IV.14 and later in Fig. IV.17, the choice of r simply tunes the information on which clusters are based. The clustering objective is to be defined for each situation, which is made possible by manual tuning of r. Such an objective could consist in minimizing the clusters' word distribution entropy, the standard deviation of the effective triggering kernel, or the average distance between events within a cluster. A possible procedure for such optimization would involve a multi-arms bandit to deal with this trade-off.

Number of clusters — In previous experiments, we compared our clustering results to the ground truth using the NMI score. We chose this metric because it allows us to compare a different number of clusters together. Indeed, it is seldom the case that PDHP infers exactly the right number of clusters.

Typically, in our synthetic experiments with 2 ground-truth clusters, the number of clusters could differ significantly at the beginning of the algorithm (up to 10 clusters at once for small values of r). However, as the number of observations increases,

smaller clusters are discarded as the algorithm converges toward the 2 correct clusters.

In real-world data, the number of clusters can grow very high –even more for small values of r. However, the number of observations each of these clusters comprise seems to follow a power-law distribution. Many of the clusters contain very few observations (5 documents or less); they are leftovers from the process as it converges towards more robust statistics. This is why in Fig. IV.16 and Fig. IV.19, we restrict ourselves to the study of the largest clusters only.

IV.4.4 Conclusion

In this section, we built the Powered Dirichlet-Hawkes process as a generalization of the Dirichlet-Hawkes process and Uniform process. We demonstrated how it improves performance on various datasets. When textual information conveys little information, or when temporal information conveys little information, and when both do, our model can correctly retrieve the original clusters used in the generation process with high accuracy.

A central consideration in document clustering is that there are no "right" clusters. For instance, we illustrate how textual content and temporal dynamics can be decorrelated in real-life applications. PDHP is flexible enough to allow to choose the weight they wish to give to temporal or textual information depending on the situation; when textual and temporal clusters are decorrelated, the model allows to choose which of those to infer.

Many future extensions are possible for PDHP. For instance, it would be interesting to develop its hierarchical version (PHDHP) as it has already been done with HDHP for DHP (Mavroforakis, Valera, and Gomez-Rodriguez, 2017). Given several recent works have been based on the regular Dirichlet-Hawkes process, it would be insightful to study how their results vary when using the Powered Hawkes-Dirichlet process instead. A study of the influence of the language model used along with PDHP would also be interesting since the text model we used here was simple on purpose (our focus being on the PDHP prior and not on the model it gets associated with). Typically, we would expect online Bayesian models that account for mutations of textual clusters over time (e.g., shifts in vocabulary, mutating words, etc.) to bring a significant improvement in modelling real-world systems (Blei and Lafferty, 2006; AlSumait, Barbará, and Domeniconi, 2008; Bassiou and Kotropoulos, 2014; Yin and Wang, 2014)

Regarding interaction modelling, we are now close to our objective. With PDHP, we can:

- Consider entities' content. An entity is no more described as a unique identifier, but instead by its semantic content. Two entities that convey the same information are now considered as such and clustered together as a more global entity (i.e., a topic here).
- Model sparse interactions. Entities are now clustered together into temporal clusters. It makes it feasible to spot interaction terms between sets of entities. The lifespan of entities is no more a problem since clusters can comprise entities spanning over extended periods, which also increases the data available for each cluster.
- Model dynamic interactions. Each cluster is associated with its own intensity function, which determines its effect on ulterior observations. Eventually, entities' influence fades away as time goes by.

In addition, we can tune the importance given to the temporal and textual dimensions when modelling interactions. By varying r, a cluster can self-stimulate essentially according to its entities publication times, or essentially according to their content.

A major shortcoming of the proposed method is that interactions can only take place within a cluster; they are self-interactions. As stated in the introduction, this can be interpreted as the diagonal of the interaction matrix in Fig. II.8 in its temporal version. Ultimately, we are also interested in studying how different clusters of entities influence each other. Therefore, we propose to extend both (Du et al., 2015) and the PDHP (Section IV.4) to the multivariate case in the next section.

IV.5 Multivariate Powered Dirichlet-Hawkes Process – Final model

IV.5.1 Introduction

IV.5.1.a Multivariate extension of PDHP

In this section, we extend the (univariate) Powered Dirichlet-Hawkes process introduced in Section IV.4 to the multivariate case as the Multivariate Powered Dirichlet-Hawkes Process (MPDHP). The various publications in a document's stream will now be able to influence each other. We detail and overcome several technical challenges that arise from considering interacting topics, and we conduct a systematic evaluation of MPDHP on a range of synthetic datasets. As a first step, evaluation is conducted on synthetic datasets only, so that we can discuss the performances and limitations of MPDHP in a completely controlled environment. By the end of this section, we want to determine whether it is possible to use MPDHP in a real-world setting

In previous works (Blei and Lafferty, 2006; Gomez-Rodriguez, Leskovec, and Schölkopf, 2013a; Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017), the understanding of large flows of data boils down to summarizing them into a composition of independent groups –clusters. However, as discussed throughout this manuscript, the processes at stake are more complex than that, given these clusters are not independent of each other; they interact.

Such interaction is illustrated in Fig. IV.21. This figure has to be considered in regard to Fig. IV.7, where each group of documents was restricted to self-interactions only; a given topic is assumed to only trigger observations from the same topic. In Fig. IV.21, it would mean that the red cluster can only trigger observations from the red cluster, as in Fig. IV.7. Here instead, we consider that the red cluster also influences the probability of triggering an observation from the blue cluster, and conversely.

IV.5.1.b Workflow

In this section, we extend the models discussed in Section IV.2 and Section IV.4 to account for cluster interaction mechanisms. **Firstly**, we detail the challenges that arise when developing the Multivariate Powered Dirichlet-Hawkes Process (MPDHP). We show that alleviating them makes it possible to achieve a linear time complexity $\mathcal{O}(N)$ (as in the original (Du et al., 2015) and Section IV.4) along with getting good clustering results. In doing so, we also relax the near-critical Hawkes process hypothesis made in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez,

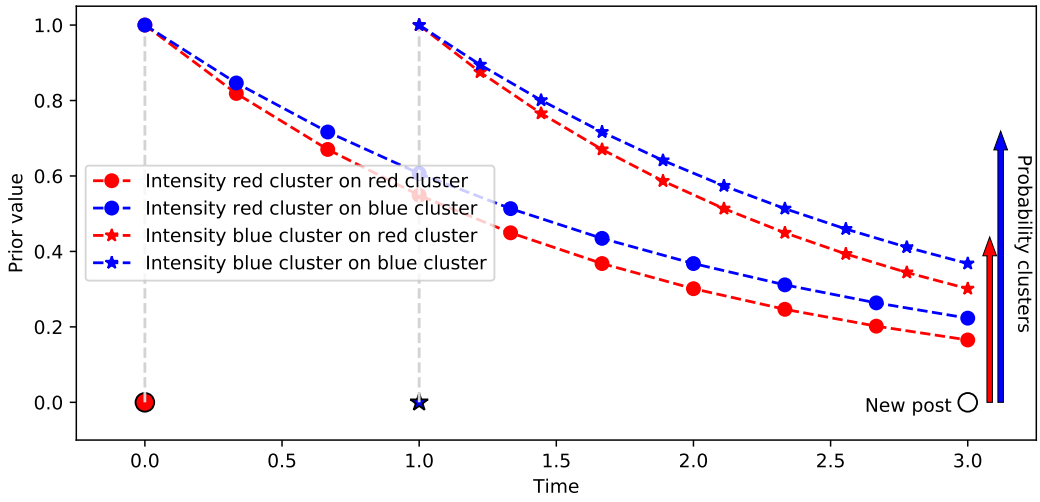

FIGURE IV.21: **Illustration of the Multivariate Powered Dirichlet-Hawkes Process prior** — A new event appears at time $t = 3$ from a cluster which is yet to be determined. The *a priori* probability that this event belongs to a given cluster c_{red} depends on the sum of the red dotted intensity functions at time $t = 3$. Similarly, the *a priori* probability that this event belongs to a cluster c_{blue} depends on the sum of the blue dotted lines at time $t = 3$. In previous existing models, this prior probability depends on each cluster self-stimulation only.

2017), and correct a major flow regarding the kernel choice in previous works. **Secondly**, we conduct a systematic evaluation of the MPDHP in a variety of synthetic situations. Our goal is to clearly identify the limits of MPDHP regarding textual and temporal overlaps, computation time, the amount of available data, the number of co-existing clusters, etc. This section is intended as a technical report. By the end of it, a potential user should know in which situations MPDHP can be useful, and in which ones alternative modelling choices are required. .

IV.5.2 The Multivariate Powered Dirichlet-Hawkes process

IV.5.2.a Multivariate Hawkes process

As discussed earlier, a Hawkes process is a temporal point process where the appearance of new events is conditional on the realization of previous events. It is fully characterized by an intensity function, noted $\lambda(t)$ that depends on the history events generated *by this process* up to t, $\mathcal{H}_{<t}$. We recall that the term $\mathcal{H}_{<t}$ is implicit anytime the intensity function λ is mentioned. A multivariate Hawkes process is an extension of the Hawkes process, where the intensity function $\lambda(t)$ depends on the history events generated *by other Hawkes processes*. It means that in the definition of $\lambda(t)$, we cannot only consider the events that happened in the same cluster, as in Eq. IV.2 and Eq. IV.29.

Similarly to Section IV.4, we associate one single Hawkes process to each cluster. However, in (P)DHP, each of them is associated with a **univariate** Hawkes process, which depends only on the history of events comprised in this cluster. In our case, instead, we associate each cluster to a **multivariate** Hawkes process that depends on all the observations previous to the time being. Let t_i^c be the time of realization of the i^{th} event belonging to cluster c. We write the intensity function for cluster c at all

times as:

$$\lambda_c(t) = \sum_{t_i^{c'} < t} \vec{\alpha}_{c,c'}^T \cdot \vec{\kappa}(t - t_i^{c'}) \tag{IV.34}$$

In Eq. IV.34, $\vec{\alpha}_{c,c'}$ is a vector of L parameters to infer, and $\vec{\kappa}(t - t_i^{c'})$ is a vector of L temporal kernel functions depending only on the time difference between two events. As we will see later, $\alpha_{c,c',l}$ is readily interpreted as the influence of c' on c according to the l^{th} entry of the temporal kernel. Once again, we consider a Gaussian RBF kernel, which allows us to model a range of different intensity functions:

$$\kappa_l(\Delta t) = \frac{1}{\sqrt{2\pi\sigma_l^2}} e^{-\frac{(\Delta t - \mu_l)^2}{2\sigma_l^2}} \quad \forall l \in L \tag{IV.35}$$

The log-likelihood of a multivariate Hawkes process for all observations up to a time T is identical to the univariate case:

$$\log \mathcal{L}(\alpha|\mathcal{H}) = \sum_c \int_0^T \lambda_c(t)dt + \sum_{t_i^c} \lambda_c(t_i^c) \tag{IV.36}$$

IV.5.2.b Multivariate Powered Dirichlet-Hawkes Process

The Multivariate Powered Dirichlet-Hawkes Process (MPDHP) arises from the merging of the Powered Dirichlet Process (Section IV.3) and the Multivariate Hawkes Process (MHP), described in the previous paragraph. As in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017) and Section IV.4, the counts in PDP are substituted with the intensity functions of a temporal point-process, here MHP. The *a priori* probability that a new event is associated with a given cluster no longer depends on the population of this cluster, but on its temporal intensity at the time the new observation appears. This is illustrated in Fig. IV.21, where two events from two different clusters c_{red} and c_{blue} have already happened at times $t_0 = 0$ and $t_1 = 1$. A new event appears at time $t = 3$. The *a priori* probability that this event belongs to the cluster c_{red} depends on the sum of the intensity functions of observations at t_0 and t_1 on cluster c_{red} at time $t = 3$ –sum of the red dotted lines. Similarly, the *a priori* probability that this event belongs to the cluster c_{blue} depends on the sum of the blue dotted lines at time $t = 3$.

Let t_i be the time at which the i^{th} event appears. The resulting expression reads:

$$P(C_i = c|t_i, r, \lambda_0, \mathcal{H}) = \begin{cases} \frac{\lambda_c^r(t_i)}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c}\leq\text{K} \\ \frac{\lambda_0}{\lambda_0 + \sum_{c'} \lambda_{c'}^r(t_i)} & \text{if c=K+1} \end{cases} \tag{IV.37}$$

In Eq. IV.37, λ_c in defined as in Eq. IV.34, and the parameter λ_0 is the equivalent of the concentration parameter described in Eq. IV.17. Taking back the illustration in Fig. IV.21, this parameter corresponds to a time-independent intensity function. It has a chance to get chosen typically when the other intensity functions are below it (meaning they do not manage to explain the dynamic aspect of a new event). In this case, a new topic is opened, and gets associated with its own intensity function.

IV.5.2.c Language model

Similarly to what has been done in the previous sections and in (Du et al., 2015), the MPDHP must be associated with a Bayesian model given it is a prior on sequential

data. Since we study applications on textual data, we choose to side the MPDHP prior with the same Dirichlet-Multinomial language model as in previous sections. We recall the likelihood of the i^{th} document belonging to cluster c reads (see Section. IV.2.2.d):

$$\begin{aligned}\mathcal{L}(C_i = c|N_{<i,c}, n_i, \theta_0) &= P(n_i|C_i = c, N_{<i,c}, \theta_0) \\ &= \frac{\Gamma(N_c + \theta_0)}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\Gamma(N_{c,v} + \theta_0)}\end{aligned} \tag{IV.38}$$

where N_c is the total number of words in cluster c from observations previous to i, n_i is the total number of words in document i, $N_{c,v}$ the count of word v in cluster c, $n_{i,v}$ the count of word v in document i and $\theta_0 = \sum_v \theta_{0,v}$.

IV.5.3 Implementation

IV.5.3.a Base algorithm

The Sequential Monte Carlo (SMC) algorithm used for the optimization has already been described in Section IV.4.2.f and in Fig. IV.8. We briefly review it in this section, before discussing the challenges that arise when using it in the multivariate case.

SMC algorithm

The goal of the SMC algorithm is to jointly infer textual documents' clusters and the dynamics associated with them. It runs as follows. First, the algorithm computes each cluster's posterior probability for a new observation by multiplying the temporal prior on cluster allocation (see Eq. IV.37, illustrated Fig. IV.21) with the textual likelihood (see Eq. IV.38). It results in an array of $K+1$ probabilities, where K is the number of non-empty clusters. A cluster label is then sampled from this probabilities vector. If the empty $(K+1)^{th}$ cluster is chosen, the new observation is added to this cluster, and its dynamics are randomly initialized (i.e., a $(K+1)^{th}$ row and a $(K+1)^{th}$ column are added to the parameters matrix α). If a non-empty cluster is chosen, its dynamics are updated by maximizing the new likelihood Eq. IV.36. The process then goes on to the next observation.

This routine is repeated N_{part} times in parallel. Each parallel run is referred to as a *particle*. Each particle keeps track of a series of cluster allocation hypotheses. After an observation has been processed, we compute the particles likelihood given their respective cluster allocations hypotheses. Particles that have a likelihood relative to the other particles' one below a given threshold ω_{thres} are discarded and replaced by a more plausible existing particle.

Sampling the temporal kernel

The parameters α are inferred using a sampling procedure. A number N_{sample} of pre-computed vectors are drawn from a Dirichlet distribution with probability $P(\alpha|\alpha_0)$, with α_0 a concentration parameter. As the SMC algorithm runs, within each existing cluster, each of these candidate vectors is associated with a likelihood computed from Eq. IV.36, noted $P(\mathcal{H}|\alpha)$, where $\mathcal{H}$ represents the data. The sampling procedure returns the average of each of the N_{sample} pre-computed α, weighted by the posterior distribution associated with them $P(\alpha|\mathcal{H}) \propto P(\mathcal{H}|\alpha)P(\alpha|\alpha_0)$. The so-returned matrix is guaranteed to be a good statistical approximation of the optimal matrix, provided the number of sample matrices N_{sample} is large enough.

Limits

This algorithm described here works well for the univariate case but fails for the multivariate case. In particular, updating the multivariate intensity function of each cluster requires knowing the number of already existing clusters, which vary over time. Therefore, we cannot pre-compute the sample matrices in advance –they must be updated as the algorithm runs to account for the right number of non-empty clusters. Moreover, the number of parameters to estimate evolves linearly with the number of active clusters K, instead of remaining constant as in DHP and variants (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017). Because the number of parameters is not constant anymore, their candidate values cannot be sampled from a Dirichlet distribution anymore. In the following, we review these challenges and propose our solutions to overcome them. Eventually, we manage to preserve a constant time complexity for each observation.

IV.5.3.b Optimization challenges

Updating the triggering kernels

In the univariate case (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017) and Section IV.4, the coefficients $\alpha_c \in \mathbb{R}^L$ are sampled from a collection of existing sample vectors computed at the beginning of the algorithm (where L is the size of the kernel vector). However, we must now infer a matrix instead. We recall that matrix α_c represents the weights given to the temporal kernel vector of every cluster influence on c –see Eq. IV.34. The likelihood Eq. IV.36 can be updated incrementally for each sample matrix. A given cluster c has a likelihood value associated with each of those N_{sample} sample matrices, which represents how fit one sample matrix is to explain one cluster's dynamics. The final value of the parameters matrix is then computed by sampling, simply averaging the sample matrices weighted by their likelihood for a given cluster times the prior probability of these vectors being drawn in the first place.

However, such sampling was possible in the univariate case, where each sample matrix was in fact a vector of fixed length. In our case, because Hawkes processes are multivariate, each entry $\alpha_c \in \mathbb{R}^{K \times L}$ is now a matrix. Moreover, the number of existing clusters K increases over time and can grow large. Each time a new cluster is added to the computation, a row is appended to the α_c matrix –it accounts for the influence of this new cluster regarding c.

Our solution is that some events can be discarded from the computation so that some old clusters can also be discarded. Clusters whose last observation has exceeded a certain age has a near-zero chance to get sampled once again. It means these clusters' contribution to the likelihood Eq. IV.36 is fixed. Therefore, they do not have a role in the computation of the parameters matrix α_c. The row corresponding to each of these clusters in the parameters matrix can then be discarded in every sample matrix. Put differently, the last sampled value for their influence on c will remain unchanged for the remaining of the algorithm. The dimension of α_c only depends on the number of *active* clusters, whose intensity function has not faded to zero. For a given dataset, the number of active clusters typically fluctuates around a constant value, making one iteration running in constant time $\mathcal{O}(1)$.

A beta prior on parameters

Another problem inherent to the proposed multivariate modelling is the prior assumption on sample vectors. In (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017), each sample vector is sampled from a Dirichlet distribution. This choice is to infer Hawkes processes that are nearly unstable: the spectral radius of their temporal kernel function $\lambda_c(t)$ is close to 1. However in our case, such an assumption is not possible because the size of each sample matrix can vary as the number of active clusters evolve. Drawing one Dirichlet vector of size L for each entry $\alpha_{c,c'}$ would force the spectral radius of α_c to equal K, which transcribes a highly unstable Hawkes process. Our solution is to consider the parameters as completely independent of each other. Each entry of the matrix α is drawn from an independent β distribution of parameter β_0. In this way, we make no assumption on the spectral radius of the Hawkes process, and sample rows/columns corresponding to new clusters can be generated one after the other.

On the temporal concentration parameter λ_0

While not specifically related to the implementation of the multivariate case, we discuss in this paragraph an important consideration when designing DHP-based models. In most recently published works on the topic (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017) and in the experiments conducted in Section IV.4, inference on real-world processes is done using an RBF temporal kernel. It means that time is paved with Gaussian functions centered at various points in time; the parameter α in DHP-based models accounts for the weights given to each of these Gaussian functions.

In these works, the kernel is chosen so that it accounts for different time scales by centering Gaussian functions on unevenly spaced points in time. The standard deviation of each of these entries varies to account for larger time ranges. However, all values of a Gaussian function are small when their standard deviation is large, for normalization reasons –the maximum value of a Gaussian function whose standard deviation is σ is $\frac{1}{\sqrt{2\pi\sigma^2}}$.

In the SMC algorithm, this RBF kernel is evaluated at a single point in time and confronted with the temporal concentration parameter λ_0 (see Eq.IV.37) to determine whether to open a new cluster. In (Du et al., 2015), such values are compared to λ_0 constant in time. It means that, mechanically, these methods cannot detect observations triggered by such Gaussian functions as their value is systematically lower than λ_0 –typically at long time ranges in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017), which can be seen from these articles' kernel plots that fade as time goes. We explicitly illustrate how this leads to a bias in the modelling in Fig. IV.22.

Consider for instance the RBF kernel used in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017), plotted in Fig. IV.22, with the Gaussian means equal to 0.5, 1, 8, 12, 24, 48, 72, 96, 120, 144 and 168 hours, and the corresponding deviations equal to 1, 1, 8, 12, 12, 24, 24, 24, 24, 24, and 24 hours. The authors used $\lambda_0 = 0.01$. For the last entry of their RBF kernel, the maximum value of the Gaussian function $\mathcal{G}(\mu = 168; \sigma = 24)$ is about 3.10^{-4}, which is much smaller than $\lambda_0 = 1.10^{-2}$. It means that even for a cluster whose intensity function only acts at long-ranges, the chances of spotting events triggered by such clusters are about 3%. This makes the models presented in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017) unfit to spot long-range interactions.

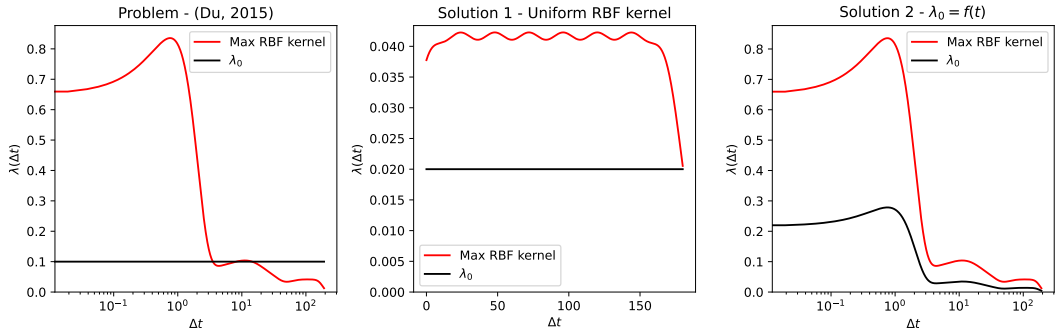

FIGURE IV.22: **Choosing the right temporal concentration parameter** λ_0 — The choice of the temporal concentration parameter λ_0 can lead to bias. **(Left)** The problem with its choice in (Du et al., 2015) is that events happening at large time ranges are likely to go undetected, as the Hawkes intensity at these ranges cannot be larger than λ_0. **(Middle)** A first solution consists in paving the space with evenly spaced Gaussian functions that all share the same standard deviation. **(Right)** A second solution is to make λ_0 a function of time so that its ratio with the temporal kernel remains constant.

There are two ways to overcome this problem (Fig. IV.22-Left) so that λ_0 can be consistently confronted to the clusters' temporal kernels (see Eq. IV.34):

- Fig. IV.22-Middle – To consider an RBF kernel whose Gaussian functions all share the same deviation, while keeping λ_0 constant. We choose this solution in the follow-up experimental section.
- Fig. IV.22-Right – To consider a λ_0 that can vary in time according to the maximum value of the RBF kernel at different time points –which depends on their standard deviation.

IV.5.4 Experiments

IV.5.4.a Setup

We now design a series of experiments to explore the possible use domain for the Multivariate Powered Dirichlet-Hawkes Process. We list the parameters we consider in our experiments. When a parameter does not explicitly vary, it takes a default value given between parentheses. These parameters are: the textual overlap (0) and the temporal overlap (0) discussed further in the text, the temporal concentration parameter λ_0 (0.01), the strength of temporal dependence r (1), the number of synthetically generated clusters K (2), the number of words associated with each document n_{words} (20), the number of particles N_{part} (10) and the number of sample matrices used for sampling α, noted N_{sample} (2,000). For the detail of these parameters, please refer to Eq. IV.37.

Note that the overlap $o(f_1, ..., f_N)$ between N functions is defined as the sum over each function f_i of its intersecting area with the largest of the $N-1$ other functions, divided by the sum of each function's total area. This value is bounded between 0 (perfectly separated functions) and 1 (identical functions). Mathematically:

$$o(f_1, ..., f_N) = \frac{1}{\sum_i \int_{\mathbb{R}} f_i(x)dx} \sum_i \int_{\mathbb{R}} min(f_i(x), max(\{f_j(x)\}_{j \neq i}))dx \tag{IV.39}$$

For each combination of parameters considered, we generate 10 different datasets. In all datasets, we consider a fixed size vocabulary $V = 1000$ for each cluster. All

datasets are made of 5,000 observations. Observations for each cluster c are generated using an RBF temporal kernel $\vec{\kappa}(t)$ weighted by a parameter matrix α_c. Explicitly, we set $\vec{\kappa}(t) = [\mathcal{G}(3;0.5);\mathcal{G}(7;0.5);\mathcal{G}(11;0.5)]$ where $\mathcal{G}(\mu;\sigma)$ is a Gaussian function of mean μ and deviation σ –following the discussion raised in Section IV.5.3.b.3. We note $L = 3$ the number of entries of $\vec{\kappa}$. The inferred entries of α determine the amplitude (or weight) of each Gaussian kernel function.

The generation process is as follows. First, we draw a random matrix $\alpha \in \mathbb{R}^{K\times L}$ and normalize it so that its spectral radius equals 1 –near unstable Hawkes process. We repeat this process until we obtain the wanted temporal overlap.[3] Then, we simulate the multivariate Hawkes process using the triggering kernels $\vec{\alpha} \cdot \vec{\kappa}(t)$, where $\vec{\kappa}(t)$ is the RBF kernel as defined earlier. Given the Hawkes process is multivariate, each event is associated with its class it has been generated from among K possible classes. For each so generated event, we draw n_{words} words from a vocabulary of size V. The vocabularies are drawn from a multinomial distribution and shifted over this distribution so that they overlap to a given extent (see Eq.IV.39).

IV.5.4.b Baselines

We compare our approach to 3 closely related baselines.

- **Dirichlet-Hawkes process (DHP)** (Du et al., 2015) – In this model, clusters can only self-replicate. It means that the intensity function of a cluster c Eq. IV.34 only considers past events that happened in the same cluster c.
- **Dirichlet process (DP)** – This prior is standard in clustering problems. It corresponds to a special case of Eq. IV.17 where $r = 1$. It assumes that the prior probability for an observation to belong to a cluster depends linearly on the population of this cluster.
- **Uniform process (UP)** (Wallach et al., 2010) – This prior corresponds to a special case of Eq. IV.17 where $r = 0$. It assumes that the prior probability for an observation to belong to a cluster does not depend on any information about this cluster (neither population nor dynamics).

As in previous experiments, we evaluate our results in terms of normalized mutual information (NMI) score. We recall that this metric is standard when evaluating non-parametric clustering models. It compares two cluster partitions (i.e., the inferred and the ground truth ones); it is bounded between 0 (each true cluster is represented to the same extent in each of the inferred ones) and 1 (each inferred partition comprises 100% of one true cluster).

IV.5.4.c Results

MPDHP outperforms state-of-the-art

In Fig. IV.23, we plot our results for datasets that has been generated using a Multivariate Hawkes process (clusters have an influence on each other) and a Univariate Hawkes process (clusters can only influence themselves). We compare MPDHP to

[3] Note that the overlap as defined here is different from the one used in Section. IV.4. In the latter, we considered the overlap of the intensity function plot on the real-time axis. Here instead we consider the overlap between the kernel intensity functions. This is because the temporal overlap as defined in Section IV.4 is always close to 1 in the multivariate case, because different clusters' events strongly interact with each other.

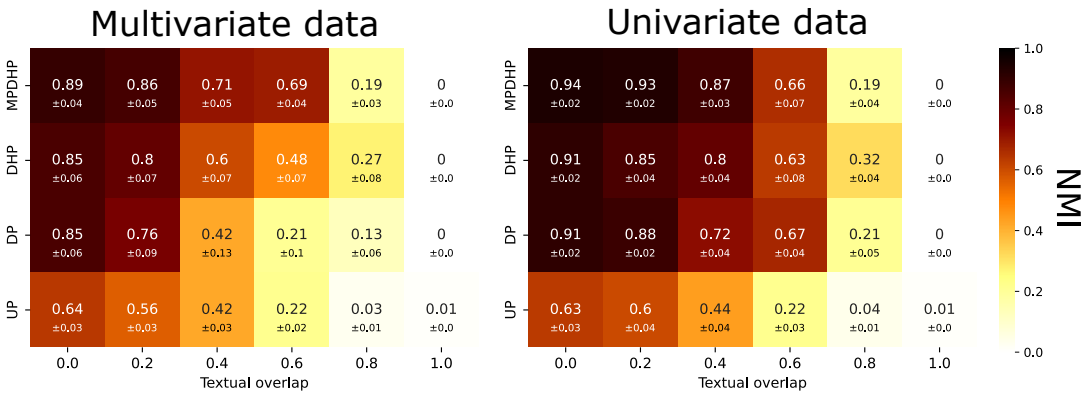

FIGURE IV.23: **Experimental results on synthetic data** — MPDHP consistently outperforms other baselines by considering both textual information and temporal information.

the proposed baselines for various values of textual overlap. We draw the following conclusions:

- **MPDHP** systematically outperforms the proposed baselines on multivariate data –when clusters interact with each other. Considering that clusters interact with each other improves our description of the datasets.
- **MPDHP** performs at least as good as PDHP on univariate data –when clusters can only self-stimulate. The complexity induced by MPDHP does not make it unfit for simpler tasks.
- **PDP** performs well for small textual overlaps, but rapidly fails when the textual overlap increases. This is expected since only the textual information is considered by the PDP. It highlights the importance of also considering the temporal information.
- **PDHP** performs better than MPDHP when the textual overlap is large (textual overlap of 0.8). This is due to the increased complexity of MPDHP over PDHP. In challenging situations such as this one, a simpler model takes fewer efforts to overcome initialization mistakes, as there are fewer parameters to put back on the right track.
- **All models** fail when the textual overlap is complete; clusters cannot be inferred from temporal information only.

Uninformative textual content and entangled dynamics

In Fig. IV.24, we plot the results of MPDHP for different values of textual and temporal overlap. Textual overlap is defined as in Eq. IV.39. According to Eq. IV.34, the influence kernel of cluster c' on cluster c can be written $\vec{\alpha}_{c,c'} \cdot \vec{\kappa}(t)$. For each cluster c, we generate values of $\alpha_{c,c'}$ so that the overlap between all the functions in the set $\{\vec{\alpha}_{c,c'} \cdot \vec{\kappa}(t)\}_{c'}$ equals a given value. The idea is to evaluate whether MPDHP is robust when clusters have similar dynamics.

Overall, we see that when the textual overlap is small, MPDHP yields good results independently from the temporal overlap. It means that in this case, the textual content is enough to differentiate clusters despite their dynamics being similar. However, as textual content gets less informative (textual overlap ≥ 0.6), results are better when the temporal overlap is low. In these cases, textual information is not

enough and MPDHP relies more on temporal data. Overall, MPDHP handles challenging cases provided either textual or temporal information is informative enough – for instance temporal overlap of 0 and textual overlap of 0.7, or temporal overlap of 1 and textual overlap of 0.4. It fails when both are uninformative – for instance, temporal and textual overlaps of 1.

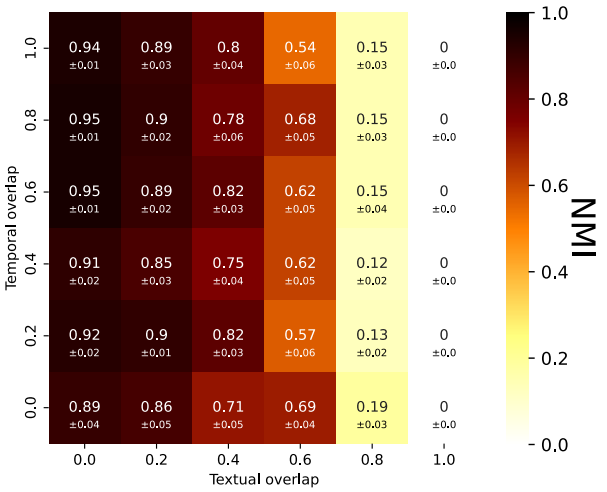

FIGURE IV.24: **MPDHP handles scarce textual or temporal information** — MPDHP handles challenging cases provided either textual of temporal information is informative enough (temporal overlap of 0 and textual overlap of 0.7; temporal overlap of 1 and textual overlap of 0.4) and fails when both are uninformative (overlaps of 1).

Highly interacting processes

Next, we assess whether MPDHP works when a large number of clusters coexist simultaneously. The rate at which new clusters get opened is mainly controlled by the λ_0 hyperparameter (see Eq. IV.37), which we vary to see whether MPDHP is robust against it. In Fig. IV.25, we plot the performances of MPDHP according to these two parameters. We can draw two conclusions:

- MPDHP can handle a large number of coexisting clusters and still correctly identify to which one each document belongs.
- MPDHP is robust versus variations of λ_0. In this case, results are similar for λ_0 varying over 5 orders of magnitude. It means MPDHP does not have to be fine-tuned according to the number of expected clusters in cases where this number is not known in advance.

Handling scarce textual information

In this paragraph, we determine how much data should be provided to MPDHP to get satisfying results. In Fig. IV.26, we plot the performances of MPDHP with respect to the number of words generated by each observation and to the clusters' vocabulary overlap. MPDHP needs a fairly small number of words to yield good results over 5,000 observations. For reference, the overlap between topics can be estimated at around 0.25 ((Posadas Duran et al., 2019), in Spanish). Similarly, we can estimate an average of ∼10-20 named entities per Twitter post (240 characters). These results support the application of MPDHP to model real-world situations.

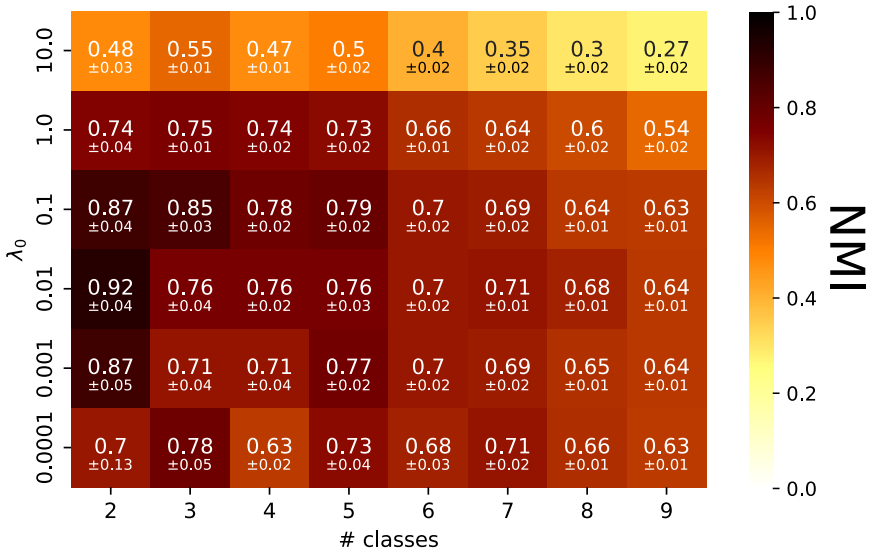

FIGURE IV.25: **MPDHP can handle a large number of coexisting clusters** — MPDHP yields good results when a large number of clusters coexist simultaneously. It is also robust against variations of λ_0 over 5 orders of magnitude.

Computational needs

Finally, we investigate how much computational resources should be allocated to MPDHP's sequential Monte-Carlo (SMC) inference algorithm to obtain good results. In Fig. IV.27, we plot the model's performance against the two main optimization parameters –the number of sample matrices and the number of particles. We recall that the sample matrices are used to infer each value of the kernel weights matrices for cluster c, noted α_c; the more sample matrices, the better the estimation. The number of particles represents the number of different cluster allocations hypotheses explored by the SMC algorithm at each step; the more particles, the more hypotheses are tested simultaneously. Overall, we see that MPDHP works well with few resources. In our experiments, results do not seem to improve significantly when using more than 20 particles, and when using more than 1000 sample vectors.

IV.5.5 Conclusion

In this section, we extended existing priors (the Dirichlet-Hawkes process and the Powered Dirichlet-Hawkes process) so that they can consider multivariate Hawkes processes, resulting in the resulting Multivariate Powered Dirichlet-Hawkes process (MPDHP). This new process is used as a Bayesian prior coupled to a textual model to infer clusters temporal interaction network from textual data flow. Along with its derivation came several optimization challenges, that we overcome to preserve a computational time that scales linearly with the size of the dataset $\mathcal{O}(N)$.

Through systematic experiments, we tested our approach against state-of-the-art models and explored its limits by varying the parameters used for synthetic data generation. We show that MPDHP outperforms existing baselines when clusters are allowed to interact with each other, and performs at least as well as the PDHP baseline when clusters are only allowed to self-interact (which PDHP is designed to model). We show that MPDHP can handle cases where textual content is uninformative better than other baselines. Besides, it handles cases where temporal dynamics are similar across clusters. We also show that MPDHP is robust against tuning of the

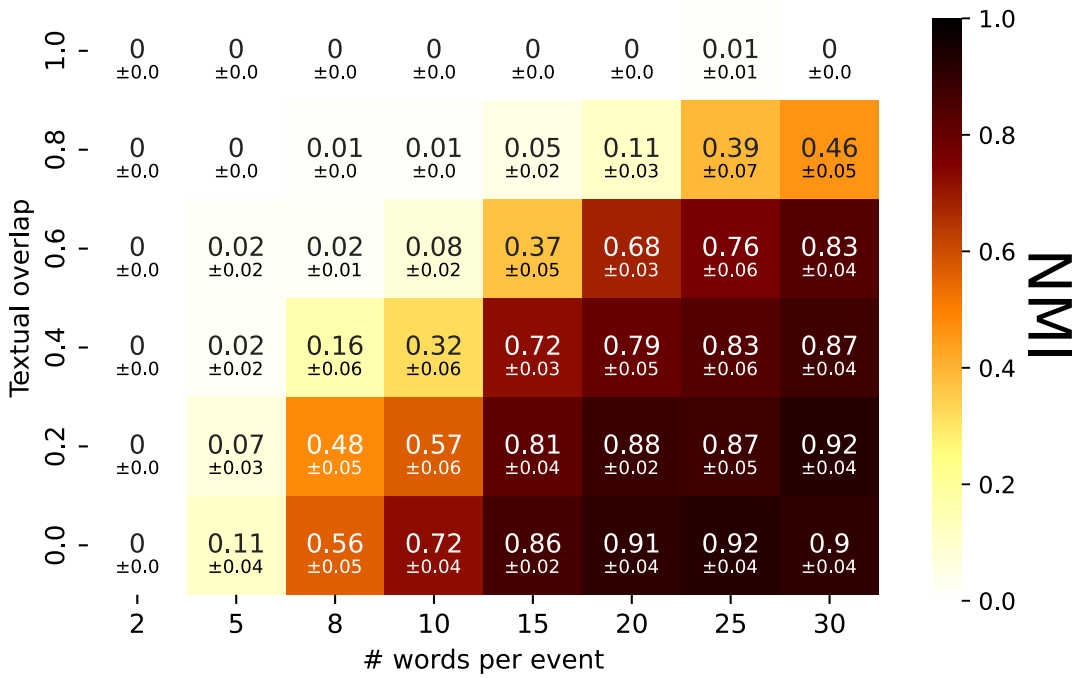

FIGURE IV.26: **How much data to use MPDHP** — MPDHP performances according to the number of words associated with each event and the overlap between clusters' vocabulary. Overall, MPDHP needs little data. This plot provides a map of how much data must be provided to MPDHP to make it work. For reference, topic overlaps in the real world can be estimated at around 0.25 and the number of named entities per Twitter post around 20.

temporal concentration parameter λ_0, which allows it to handle highly intricate processes where a lot of clusters coexist simultaneously. We evaluated the robustness of MPDHP against the number of words observed for each event and the overlap between various clusters and showed that it performs well with few textual data when vocabularies overlap is not total. Finally, we discussed the computational needs of PDHP, and show that it works correctly with minimal computational resources.

The present section was intended as a report on what can and what cannot be achieved using MPDHP. Our various results suggest that this prior can be applied in a robust way to a broad range of problems for a minimal computational cost. In particular, the results from these extensive experiments support the possibility of applying MPDHP to real-world situations.

Regarding the task at hand, it appears that MPDHP provides a robust way to model interactions in information spread. The PDHP allowed to model sparse and dynamic self-interactions between semantic entities; these interactions can now take place between different entity clusters.

IV.6 Case study on a real-world dataset – Reddit news

IV.6.1 Introduction

Throughout the previous sections, we developed a plethora of models to investigate interactions in information spread. A first approach underlined the necessity of considering clustering and a second one the necessity of considering the temporal dimension. To answer these conclusions, we extended a promising class of models, the Dirichlet-Hawkes process, so that it becomes possible to spot temporal interactions

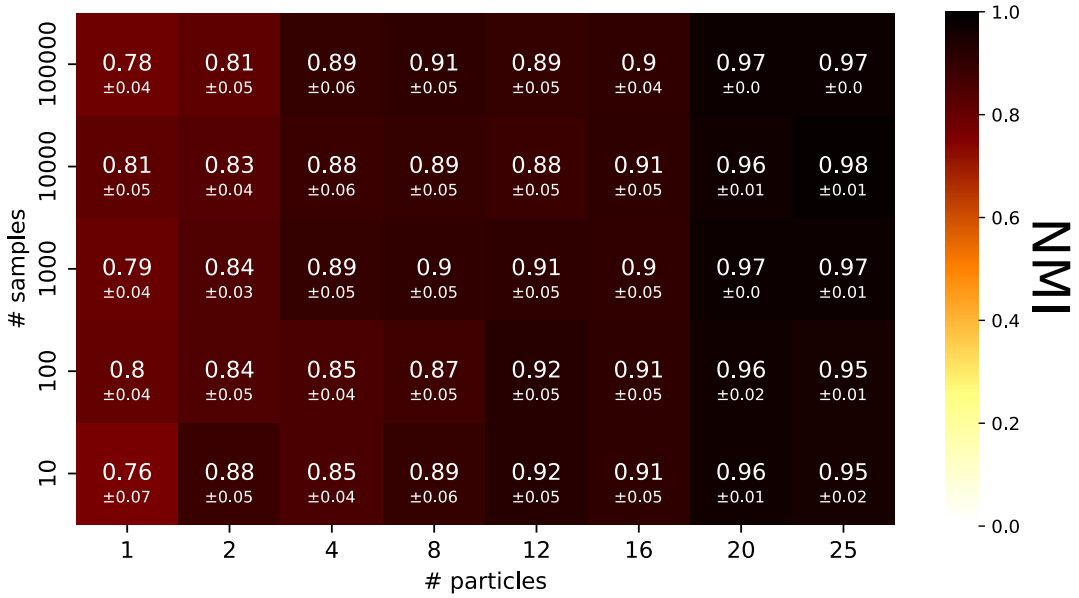

FIGURE IV.27: **How complex should the algorithm be** — Performance of MPDHP using different versions of the Sequential Monte-Carlo algorithm. Here, we plot the model's performance with respect to the number of sample matrices used to estimate the kernel's weights α_c (see Eq. IV.34) and the number of particles N_{part} used for the inference. Overall, MPDHP functions well with few computational resources.

between clusters of textual documents in large-scale settings. The resulting Multivariate Powered Dirichlet-Hawkes Process (MPDHP) answers all these constraints and yields interpretable results on interacting processes.

As a closure of our work on interactions modelling in information spread, we propose to conduct a large-scale experiment on a real-world dataset using MPDHP. In this section, we first describe and justify the choice of a large-scale real-world dataset from Reddit. In a second time, we conduct an in-depth analysis of the results yielded by MPDHP. We finally conclude on the role of interactions in the spread of news headlines on Reddit.

IV.6.2 Dataset

IV.6.2.a Origin and raw data

Why Reddit?

Reddit is a social network platform that gathers 48M of monthly users, as of today. Reddit is composed of a galaxy of forums (or subreddits) that are often specific to a topic or an organisation. Within each forum, users can open a new post made of a title and a content (typically textual or visual). Users can then comment on the post and get answered to. The contents published on the platform are therefore user-generated and highly dynamic, with an estimated 1M of new publications per day. Each post can be upvoted (score +1) or downvoted (score -1); the popularity of the post is defined as the difference between upvotes and downvotes.

For these reasons, this corpus fits our demonstration: information emerges from a large user-generated data flow, its contents are formatted, textual, timestamped and user-generated, and the number of daily publications makes it likely that some of them interact with each other.

Data

We collected our dataset from the Pushshift Reddit repository (Baumgartner et al., 2020). As its authors describe it, *"Pushshift is a social media data collection, analysis, and archiving platform that since 2015 has collected Reddit data and made it available to researchers. Pushshift's Reddit dataset is updated in real-time and includes historical data back to Reddit's inception. [...] The Pushshift Reddit dataset makes it possible for social media researchers to reduce time spent in the data collection, cleaning, and storage phases of their projects."*

In practice, the dataset comprises the entirety of the content posted on Reddit up to June 2021. In particular, for each Reddit post, we can retrieve the subreddit it came from, the title of the publication, its publication date and its score.

IV.6.2.b Preprocessing

Selecting the news subreddits

For the need of our study, we restrict ourselves to consider only popular English news subreddits. Namely, we select only posts from the following subreddits: inthenews, neutralnews, news, nottheonion, offbeat, open news, qualitynews, truenews, worldnews. This first routine leaves us with 867,328 headlines, which makes a total of 1,111,955 words drawn from a vocabulary of size 36,284.

Cleaning the textual data

As it is common in natural language processing, we must clean the raw text extracted from the Reddit posts so it becomes usable. To do so, we conduct the following routine:

1. Remove the web addresses
2. Put the text in lowercase
3. Remove punctuation signs
4. Remove extra white spaces
5. Remove all English stopwords (imported from nltk)
6. Remove all words whose length is lesser than 4
7. Remove all words that appear less than 3 times in the original dataset

Removing uninformative documents

Next, we remove publications that carry lesser textual or temporal information.

Firstly, we choose not to consider the publications that have a popularity lesser than 20 – meaning that they received less than 20 positive votes more than negative votes. We make this choice so that we consider publications that are visible enough to have any influence on the data generation process.

Secondly, we remove the publications that comprise less than 3 words. The semantic information carried is expected to be poor and is not considered in our analysis.

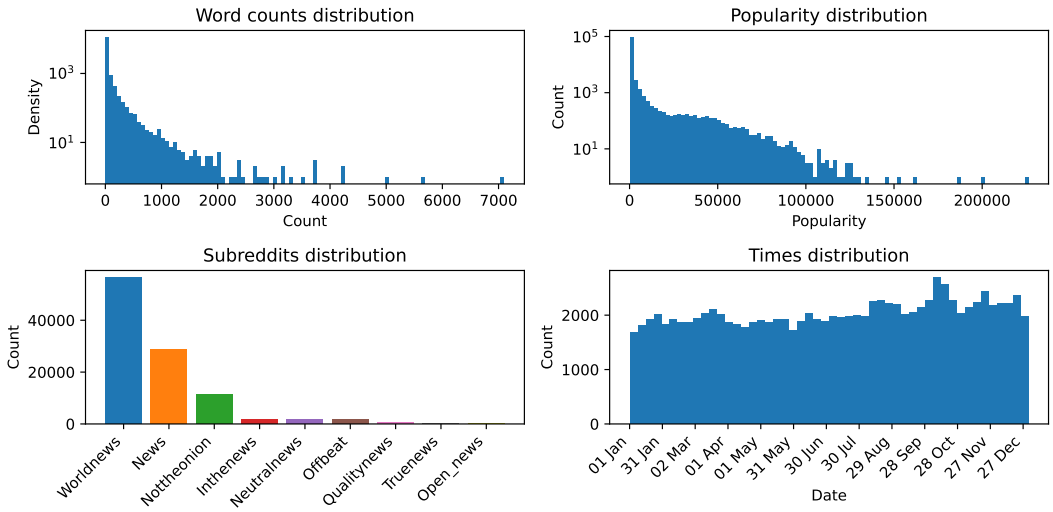

FIGURE IV.28: **Characteristics of the News dataset** — For ∼100,000 headlines and ∼13,000 different words (Top-Left) Distribution of the words count (Top-Right) Distribution of headlines popularity (Bottom-Left) How many headlines per subreddit (Bottom-Right) How publications spread over time

Final dataset

After curating the dataset in the way described above, we are left with 102,045 news headlines (one-eighth of the original data), which makes a total of 875,334 tokens (named entities, verbs, numbers, etc.) drawn from a vocabulary of size 13,241 (one-third of the original vocabulary). The characteristics of this dataset are shown in Fig. IV.28.

IV.6.3 Experimental setup

As announced earlier, we will apply the MPDHP of Section IV.5 to the News dataset detailed above. First, we must determine which hyper-parameters to use.

Temporal kernel

We run our experiments using three different RBF kernels, which should account for publication dynamics at three different timescales: minute, hour, and day. The "**Minute**" RBF kernel is made of Gaussian functions centered at the following times: $[0, 10, 20, 30, 40, 05, 60, 70, 80]$ minutes; each entry shares a same standard deviation σ of 5 minutes, and $\lambda_0 = 0.01$. The "**Hour**" RBF kernel has Gaussians centered around $[0, 2, 4, 6, 8]$ hours, with a standard deviation σ of 1 hour, and $\lambda_0 = 0.001$. The "**Day**" RBF kernel is centered around $[0, 1, 2, 3, 4, 5, 6]$ days, with a standard deviation σ of 0.5 days, and $\lambda_0 = 0.0001$.

For each of these kernels, we set the concentration parameter λ_0 so that it reaches roughly the value of one Gaussian function evaluated at 2σ. It means that an event which is 2σ away from the center of the Gaussian kernel of a single observation has 50% chances of getting associated with this Gaussian kernel entry, and 50% chances of opening a new cluster. It also translates that the chances to open a new cluster given an observation right in the center of a single RBF kernel entry are roughly $\frac{1}{8}$.

This allows us to spot interactions that might occur at different timescales. However, one of these timescales is likely to explain the data much better than the others.

If that is the case, it would mean that the dynamics at stake are not scale-free –that interactions do not have noteworthy influence at every temporal scale.

Language model

We use the same Dirichlet-Multinomial language model as in (Du et al., 2015) and in Section IV.4 and Section IV.5. We expect the vocabulary to show enough variations across the range of topics discussed in the News dataset. We vary the concentration parameter of this model between $\theta_0 = 0.001$ and $\theta_0 = 0.01$. The choice of these values is supported by other works using a similar model (Du et al., 2015) and by our own observations in Section IV.4. A larger value of θ_0 makes the inferred clusters cover a broader range of document types, whereas a small value makes the inferred clusters more specific to a topic.

SMC algorithm

Finally, we detail and justify the parameters used for running the SMC algorithm. The procedure to update the optimal parameters matrix α relies on the average of a set of sample vectors weighted by these vectors' likelihood. The number of active clusters can grow large, and the number of parameters scales as the square of this number. We therefore need to have enough sample matrices to guarantee a good approximation of the optimal α –the value recommended for 2 clusters in Fig. IV.27 is likely to be too small for the task.

We set $N_{samples} = 100000$. From our observations, the number of coexisting clusters can go as large as 80 coexisting ones (roughly 40,000 parameters to estimate), that is 2.5 sample values per parameter. In practice however, the number of coexisting clusters remains fairly low, around 10 coexisting clusters (roughly 1,000 parameters), allowing sampling each parameter from 100 candidate values.

Each sample parameter is drawn from an identical Beta distribution of concentration parameter $\alpha_0 = 2$. We set this value so that extreme values (0 and 1) for these parameters are rare, and so that the sampled parameters matrices can show great variations from one another.

Finally, we consider 8 particles for the SMC algorithm, similarly to what is done in (Du et al., 2015) and in Section IV.4, and recommended in Fig. IV.27.

IV.6.4 Results

IV.6.4.a Overview of the experiments

In our experiments, we used three temporal kernels that account for dynamics at different time scales, and tested diverse values for the language model concentration parameter θ_0 and for the exponent r. In Table IV.2, we represent the main characteristics for each of these runs in terms of number of inferred clusters K, the average cluster population $< N >$ (where $< \cdot >$ denotes the average), the average normalized entropy of the vocabulary of the top 20 clusters $S^{(20)}_{text}$, the average normalized entropy of the subreddits partition of the top 20 clusters $S^{(20)}_{sub}$.

We recall that the normalized entropy is defined as:

$$S(\vec{x}) = -\frac{1}{\ln |\vec{x}|} \sum_{i}^{|\vec{x}|} x_i \ln x_i \tag{IV.40}$$

where $\vec{x}$ is a vector that sums to 1 and $|\vec{x}|$ its cardinal (length). Each entry x_i represents the probability of i. When considering counts, $\vec{x}$ can be set equal to the frequency of each observation. The entropy is normalized between 0 (minimal spread, $\vec{x}_i = \delta_{ij}\,\forall i$) and 1 (maximal spread, $\vec{x}_i = \frac{1}{|\vec{x}|}\,\forall i$). In our case, a low entropy $S_{text}^{(20)}$ (resp. $S_{sub}^{(20)}$) means that clusters contain documents that are concentrated around a reduced set of words (resp. of subreddits); conversely, a large entropy means that clusters do not account for documents concentrated around a specific vocabulary (resp. set of subreddits).

$\vec{\kappa}(t)$	θ_0	r	K	$< N >$	$S_{text}^{(20)}$	$S_{sub}^{(20)}$
Minute	**0.01**	0.0	16150	6	0.744(68)	0.400(36)
		0.5	8498	12	0.796(55)	0.441(24)
		1.0	5069	20	0.790(49)	0.476(53)
		1.5	2730	37	0.808(43)	0.490(52)
	0.001	0.0	46304	2	0.475(48)	0.239(56)
		0.5	37277	3	0.485(40)	0.256(60)
		1.0	26858	4	0.501(37)	0.266(72)
		1.5	19275	5	0.506(39)	0.280(58)
Hour	**0.01**	0.0	3792	27	0.798(52)	0.474(106)
		0.5	1735	59	0.791(45)	0.469(77)
		1.0	825	124	0.803(47)	0.484(75)
		1.5	426	240	0.795(37)	0.489(69)
	0.001	0.0	18012	6	0.760(86)	0.397(63)
		0.5	11923	9	0.784(72)	0.425(38)
		1.0	4837	21	0.821(50)	0.497(30)
		1.5	2368	43	0.814(42)	0.481(91)
Day	**0.01**	0.0	609	168	0.713(34)	0.413(103)
		0.5	326	313	0.728(33)	0.429(98)
		1.0	172	593	0.743(31)	0.461(94)
		1.5	96	1063	0.755(36)	0.464(86)
	0.001	0.0	4349	23	0.705(49)	0.396(103)
		0.5	2654	38	0.721(59)	0.404(104)
		1.0	1399	73	0.734(57)	0.431(106)
		1.5	764	134	0.741(54)	0.442(98)

TABLE IV.2: **Results for each experiment** — We tried various combinations of parameters $\vec{\kappa}(t)$, θ_0 and r and observe how they result in a variety of outputs. We characterize these outputs in terms of clusters (number, size, textual entropy, subreddits entropy). The standard deviation on the last digits is given in standard notation – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$.

We can make several observations from Table IV.2:

- We recover the fact that textual clusters have a lower entropy for small values of r; this is because their creation is based more on textual coherence than on temporal coherence.
- The subreddit entropy *seems* to grow with r, but no clear trend is visible due to large error bars. A possible interpretation is that favouring the temporal information for cluster creation results in larger clusters. They would be too large to account for subreddit-specific dynamics. However, the entropy remains fairly low. The entropy of the distribution Fig. IV.28-top-left is equal to 0.51.

- The number of inferred clusters decreases with r, and their average population increases.
- The number of clusters grows large for the "Minute" kernel. This is because the short time range considered does not allow for clusters to last in time. A cluster that does not replicate within 1h30 is forgotten.

Choosing a timescale

From Table IV.2, we see there can be a large variety of outputs to analyse, depending on the modelling choices. If we are interested in the micro-interactions that happen at tiny time scales, the "Minute" RBF kernel should be considered. However, the short time range it allows for interactions makes the number of clusters large, and the average cluster population small. On the other hand, the "Day" RBF kernel spans over larger periods, which prevents discarding clusters too soon. For instance, it will not discard clusters that follow a circadian publication dynamic, unlike the "Minute" kernel that cannot account for such time ranges.

Choosing θ_0

In a Dirichlet-Multinomial textual model, such as written in Eq. IV.7, the hyper-parameter θ_0 controls the concentration of topics' vocabulary. A small value of θ_0 makes it so that a new document should have an almost identical word distribution as a given cluster to enter it; there are more chances that small discrepancies lead to opening a new cluster. Conversely, larger values of θ_0 allow documents to belong to clusters even if their word distribution does not fit exactly the cluster's content. We tested a small value $\theta_0 = 0.001$ and a large value $\theta_0 = 0.01$, which are standard in usual text modelling (Blei, Ng, and Jordan, 2003; Blei and Lafferty, 2006; Du et al., 2015). The choice of this parameter controls the level of specificity wanted for the textual clusters.

Choosing r

We saw the experiments of Section IV.4 that the choice of r allows us to nudge the clustering towards textual-based clustering of temporal-based clustering. Smaller values of r favour the textual content as the main information in the creation of the clusters, whereas larger values of r make their composition rely more on the inferred temporal dynamics.

In our experiments, we see that smaller values of r tend to increase the number of inferred clusters. This can be explained in the following way: the temporal concentration parameter λ_0 has been fixed so that an observation 2σ away from the center of one RBF kernel entry (that is $\sim$95% chances not to be triggered by it) has $\sim$50% chances to open a new cluster. Mechanically, it is likely that λ_0 is lesser than the value of the RBF kernel most of the time. However, for smaller values of r, the gap between λ_0 and the temporal intensity $\lambda(t)$ fades to 0 (see Fig. IV.7), making it more likely to open new clusters from the temporal perspective. In the limit $r \to 0$, we recover a Uniform Prior (Wallach et al., 2010), making the opening of new clusters entirely governed by the parameter θ_0 –clusters are created and filled based on textual content only. On the contrary, when r is large, the gap between λ_0 and $\lambda(t)$ is greater, making it less likely to open new clusters from a temporal perspective. In the limit $r \to \infty$, the opening and filling of clusters is deterministically governed

FIGURE IV.29: **Timeline of the top inferred clusters from news headlines from 01/2019 to 12/2019** — Each line is normalized with respect to its maximum value. Each bin accounts for half a day. The darker, the more observations in the cluster at a given time.

by the temporal information, as the smallest gap in the intensities leads to a Dirac distribution on the cluster with the largest temporal intensity.

IV.6.4.b Visualizing topics over time

In Fig. IV.29, we plot the timeline of the inferred clusters on a real-time axis for one of our experiments (kernel "Hour", $\theta_0 = 0.01$, $r = 1$) for illustration purpose. Each bin represents a half-day period. We can make two interesting observations from this figure.

Firstly, two clusters seem to be always present. Their intensity does not follow any visible bursty dynamics. When we look at their composition, we notice that the first cluster is made of 75% of articles from the subreddit r/worldnews, which is +20% from what one would expect from chance (55% of the corpus is from r/worldnews, see Fig. IV.28). Similarly, the second cluster comprises 46% of r/news articles, which is also roughly +20% from expected at random (28% of the corpus is from r/news, see Fig. IV.28). These two clusters therefore significantly account for publications from either of these subreddits, independently from the textual content. Our first intuition is that there are strong interactions between these subreddits. Both are general news forums with a large audience; an article that gets posted in one of them is highly likely to be copy-pasted on the other.

Secondly, topics that are not part of these two clusters appear and fade quickly in time. This is in line with the expected behaviour of news on the internet, which typically bursts for a few hours/days before being replaced by the latest news. We see however from this plot that only a small fraction of the inferred clusters coexists simultaneously in the dataset. The chance of spotting an interaction is therefore weak, as noted in Chapter II.

Thirdly, it seems that clusters expand on larger periods at the beginning of the algorithm. This is due to the cold start of MPDHP. It needs some time before statistically distinguishing clusters and explores several directions at once. The early clusters are artefacts of such a cold-start effect.

IV.6.4.c Quantifying interactions

Effective interaction

We introduce the parameters we are going to use in follow-up analyses. The output of MPDHP consists of a list of clusters comprising timestamped bags of words –news headlines. Between each pair of clusters, MPDHP inferred a temporal influence function $\lambda(t)$, that represents the probability for one cluster to trigger publications from another. Therefore, our model yields an adjacency matrix $A \in \mathbb{R}^{K\times K\times L}$, where K is the number of clusters and L the size of the RBF kernel $\vec{\kappa}(t)$. One entry $a_{i,j,l}$ represents the strength of the influence of j in i due to the l^{th} entry of $\vec{\kappa}(t)$.

However, we must consider the effective number of interactions to get relevant metrics. A given triggering function $\lambda(t)$ could be inferred from the observation of very few observations only; we must weigh these interactions. To do so, we simply consider a normalized weight matrix $W \in \mathbb{R}^{K\times K\times L}$, whose entries $w_{i,j,l}$ are the average of the intensity of i above λ_0 due to j from the kernel entry l for all observations. Explicitly:

$$w_{i,j,l} = \frac{1}{|\mathcal{H}_i|} \sum_{t_i \in \mathcal{H}_i} \sum_{t_j < t_i} max(a_{i,j,l}\kappa_l(t_i - t_j) - \lambda_0, 0) \tag{IV.41}$$

where $t_x \in \mathcal{H}_x$ is the publication time of an observation from cluster x and λ_0 the temporal concentration parameter. Note that we retract λ_0 from the intensity term, because it is considered as a background probability for a publication to happen – similarly to the virality in Chapter II and to the background noise Chapter III. Note that W can also be interpreted as the instantaneous increase in probability due to interactions.

Note that in all the following computations, we do not consider clusters that comprise less than 10 documents. Such clusters are considered leftovers from the algorithm.

Interactions strength

In Table IV.3, we investigate the effective impact of interactions in the dataset. We consider the following metrics:

- $< A >$: the average value of the whole adjacency matrix. It tells us to which extent pieces of information are connected to each other according to MPDHP.
- $< W >$: the average value of the effective interactions. It tells us the extent to which the interactions (encoded in A) effectively happen in the dataset.
- $< A >_W$: the average of the inferred interaction matrix A weighted by the effective interactions W. In this case, W can be interpreted as our confidence in the corresponding entries of A given the data with which they were inferred. This value quantifies the overall role of interactions in the dataset.
- $\frac{<W^{intra}>}{<W^{extra}>}$: ratio of the intra-cluster effective interactions with the extra-cluster effective interactions. This tells us how much different clusters influence each other versus how much they influence themselves.

When computing the means, we discard the entries of A and W equal to 0. This is because all clusters do not exist simultaneously, and thus should not be considered. An interaction strictly equal to 0 means that clusters simply did not exist at the same time.

The main conclusion of the results Table IV.3 is that most interactions are weak. The average value of A tells us that the average value of the inferred parameters is around 0.05, which is few given the value is bounded between 0 and 1. The metric $< W >$ tells us that on all events, the interaction between clusters rose the probability of publication by 0.1%-1% on average. We can also note that the values of $< W >$ are of the same order of magnitude as λ_0 (0.01 for the "Minute" kernel, 0.001 for "Hour", and 0.0001 for "Day"). We can interpret this as the probability for a new document belonging to a cluster or being from a new cluster is roughly the same from a temporal perspective. The metric $< A >_W$ tells us that when weighting the average of A with the effective interaction, the values of A are slightly higher than 0.05; we can now be confident in this value, given it has been inferred on a statistically significant number of observations. However, it still tells us that only some interactions are significant, which correlates with the findings of Chapter II. Finally, the last metric $\frac{<W^{intra}>}{<W^{extra}>}$ finds that most effective interactions take place more often within the same cluster, meaning that clusters tend to self-replicate. Only for the "Hour" kernel and $\theta_0 = 0.01$ this value is lesser than one. It is because in this case, MPDHP infers two large clusters that exist for the entire year (see the 2 first rows of Fig. IV.29) and side topic-specific clusters. These clusters strongly influence each

$\vec{\kappa}(t)$	θ_0	r	$< A > (10^{-3})$	$< W > (10^{-5})$	$< A >_W (10^{-3})$	$\frac{<W^{intra}>}{<W^{extra}>}$
Minute	**0.01**	0.5	49(21)	342(889)	66(17)	1.8(62)
		1.0	48(20)	478(1124)	60(17)	1.4(43)
		1.5	48(20)	746(1901)	60(17)	1.0(33)
	0.001	0.5	50(22)	316(882)	66(17)	3.1(138)
		1.0	50(21)	279(752)	67(16)	2.6(105)
		1.5	50(22)	268(665)	67(16)	2.3(84)
Hour	**0.01**	0.5	49(18)	389(843)	56(17)	0.5(13)
		1.0	49(18)	478(1187)	56(17)	0.6(15)
		1.5	48(17)	471(789)	52(15)	0.7(13)
	0.001	0.5	50(21)	110(398)	61(17)	1.7(67)
		1.0	50(18)	133(506)	57(17)	1.4(60)
		1.5	49(17)	183(554)	55(17)	1.1(37)
Day	**0.01**	0.5	49(18)	41(97)	55(17)	1.2(34)
		1.0	49(19)	63(131)	54(17)	1.2(31)
		1.5	49(19)	91(187)	53(18)	1.2(31)
	0.001	0.5	50(20)	18(90)	60(19)	1.1(59)
		1.0	50(19)	23(101)	58(17)	1.0(50)
		1.5	50(19)	37(111)	56(18)	1.0(36)

TABLE IV.3: **Interaction strength** — Overall, interaction between clusters is weak. The large standard deviations suggest that there is a large variety of interacting behaviours. Interactions tend to happen within a cluster (self-interactions).

other, and all the topic-specific clusters can interact with them. In fact, the same effect explains the decrease of $\frac{<W^{intra}>}{<W^{extra}>}$ as r grows: fewer clusters are inferred, and the probability of having large clusters that last for the whole period increases.

Another major observation from Table IV.3 is that standard deviations of effective interactions are large. It means that despite most interactions being weak, some of them play a significant role in the dataset. In Fig. IV.30, we plot the distribution of effective interactions for one specific run ("Hour" kernel, $\theta_0 = 0.01$, $r = 1$). Note that we recover the same trend in all other experiments. The results of this figure are similar to the ones in Chapter II (Fig. II.7): most interactions are weak, and only a few of them are significant.

Interactions range

Finally, in Table IV.4, we investigate the range of effective interactions in every experiment. The interaction range studies the persistence of interactions. To do so, we compute the effective interaction for each kernel entry individually (and not the aggregated interaction as in Table IV.3) and average it over all existing clusters.

Importantly, the raw values of effective interaction corresponding to the first entry of the triggering kernel κ_1 are consistently smaller than subsequent values. This is induced by our kernel choice, because κ_1 is always centered around $t = 0$, which makes half of the associated Gaussian function account for (impossible) backwards influence. Therefore, where other kernels can contribute on both sides of their means, κ_1 cannot. In Table IV.4, we extrapolate their value as twice the computed one.

We see in Table IV.3 that influence tends to decrease over time for all the kernels considered, after reaching a first peak. Overall, the interaction between documents

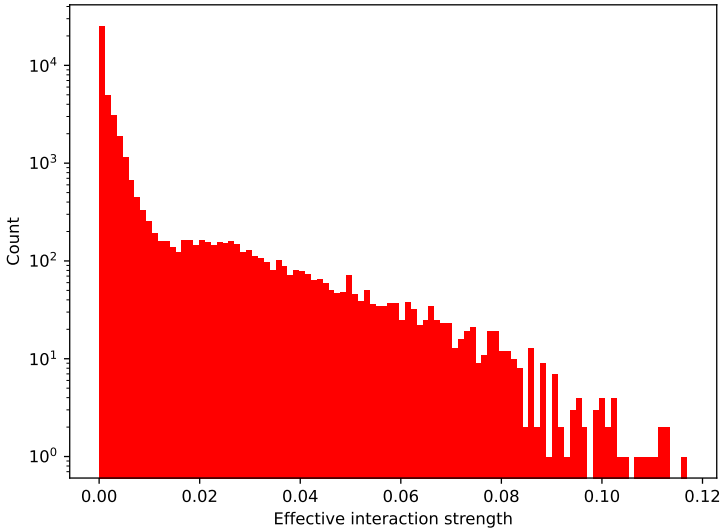

FIGURE IV.30: **Distribution of interaction strength** – Most interactions are weak.

seems to play a marginal role still. We did not plot the standard deviation for visualization purposes, but they are as large as in Table IV.3. Therefore, *most* interactions do not play a significant role in the publication of subsequent documents over time, but it greatly helps identify the right cluster for some of them. Overall, the increase in probability for a new document to belong to a cluster due to interactions is within 0.1%-1% (we recall that $\lambda(t) = \lim_{\Delta t \to \infty} \frac{P(\text{event in } \Delta t)}{\Delta t}$).

IV.6.4.d Visualizing topical interactions

In this section, we plot the temporal interaction network between the inferred clusters both on the global and the monthly scale. The clusters' composition is given explicitly. Each edge between a pair of clusters represents two metrics: the inferred interaction strength A, and the effective interaction W. The inferred interaction strength A is plotted using a colour code (darker is stronger). The effective interaction is represented using transparency (the less transparent the stronger the effective interaction). Therefore, a barely visible dark edge means that MPDHP inferred a strong interaction, but that few of them were effectively observed.

Experiment considered for subsequent analyses

In the following paragraphs, we consider the "Hour" kernel, with $\theta_0 = 0.01$ and $r = 1$. We justify this choice for easing the interpretation of our results. This way, we restrict our analysis to clusters that do not contain fragments of a whole. For instance, we prefer to have only one cluster about the Notre-Dame cathedral fire and related news instead of three clusters containing fragments of the news, such as the initial fire, the political reactions, funds raising, etc. Therefore, we choose to consider $\theta_0 = 0.01$, which avoids clusters to be overly specific. We choose the "Hour" kernel, which spans over long enough periods so that news fragments about a similar topic are considered as possibly related. Besides, from direct observation, it seldom happens for news to stick around for more than a few days, which the "Hours" kernel is fit to capture. The choice of r is based on an arbitrary trade-off between textual and temporal information. We do not want to consider extreme values ($r = 0$ or $r > 2$) so that we exploit both information pieces. Besides, we saw

$\vec{\kappa}(t)$	θ_0	r	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8	κ_9
			0m	10m	20m	30m	40m	50m	60m	70m	80m
Minute (m)	**0.01**	0.5	266	421	407	451	428	403	395	345	121
		1	396	591	580	607	532	575	521	507	224
		1.5	616	937	893	914	955	840	808	810	304
	0.001	0.5	436	509	457	424	371	340	313	178	52
		1	284	435	396	388	343	327	272	187	45
		1.5	208	388	366	353	326	333	290	215	61
			0h	2h	4h	6h	8h	-	-	-	-
Hour (h)	**0.01**	0.5	494	430	502	456	324				
		1	658	538	549	542	451				
		1.5	558	615	532	526	411				
	0.001	0.5	124	149	119	137	92				
		1	154	164	172	149	111				
		1.5	208	244	223	197	156				
			0d	1d	2d	3d	4d	5d	6d	-	-
Day (d)	**0.01**	0.5	44	45	47	46	47	47	37		
		1	70	71	72	68	68	70	61		
		1.5	102	100	101	105	98	105	82		
	0.001	0.5	18	20	21	21	21	21	17		
		1	24	26	24	27	28	26	22		
		1.5	21	41	42	41	41	41	35		

TABLE IV.4: **Interaction range** — All the values for effective interaction are given in ten-thousandth (10^{-5}). Influence tends to decrease over time for all the kernels considered.

in Table IV.3 and Table IV.4 that only slight variations are observed across the range of $r \in \{0.5, 1, 1.5\}$ considered. Finally, we are interested in seeing how the large clusters spanning over the whole period relate to topic-specific smaller clusters seen in Fig. IV.29.

Representing individual clusters

In Fig. IV.31, we represent the raw data that we will use in ulterior visualizations for some selected clusters. In the top part, we represent their textual content as a wordcloud. In this case, we chose to pick clusters about climate change inaction protests, catholic church child abuse scandals, China's Uighur detention camps, "Gilets Jaunes" protests in France, and Brexit. For each cluster, we represent the top temporal influence that other clusters may exert on them. Transparency accounts for the effective interaction W discussed in the previous sections. The bottom plot represents the influence exerted on these clusters by all other clusters at all times –which does not mean this influence led to a publication.

This way of representing the data fits well in the univariate case, as in (Du et al., 2015), but does not allow to capture the complexity of the inferred mechanisms at stake. In the following sections, we propose alternative visualizations in the form of temporal networks.

Globally

In Fig. IV.32, we plot the clusters interaction network over the whole period we considered (12 months). This figure uses the same data as Fig. IV.29. Both the adjacency

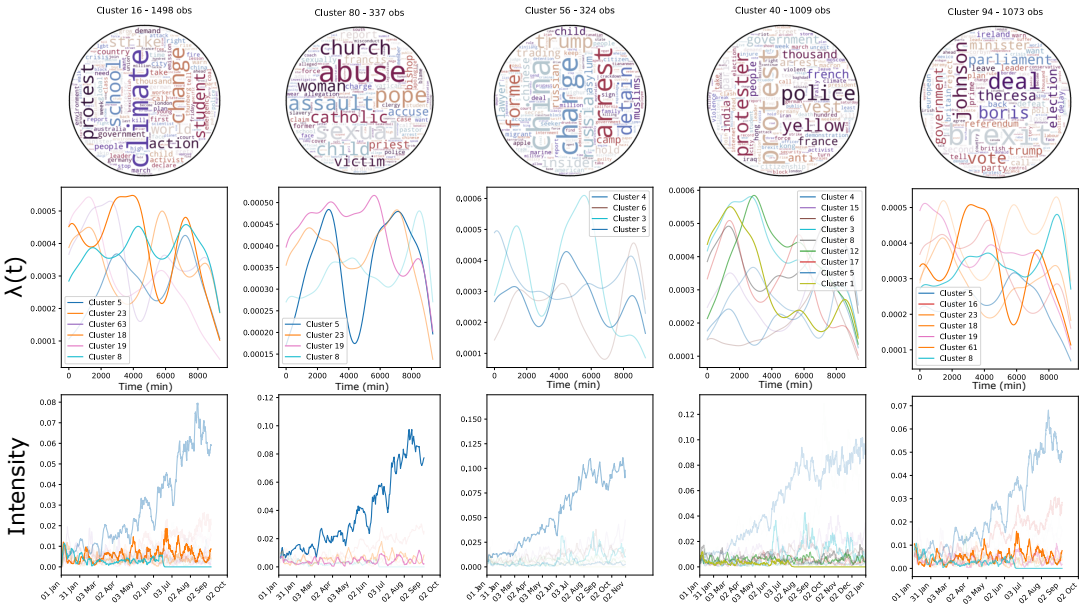

FIGURE IV.31: **A typical MPDHP output** – A set of manually selected clusters along with the vocabulary of their documents (top), their inferred dynamics (middle) and the clusters that influenced them on the real-time axis (bottom).

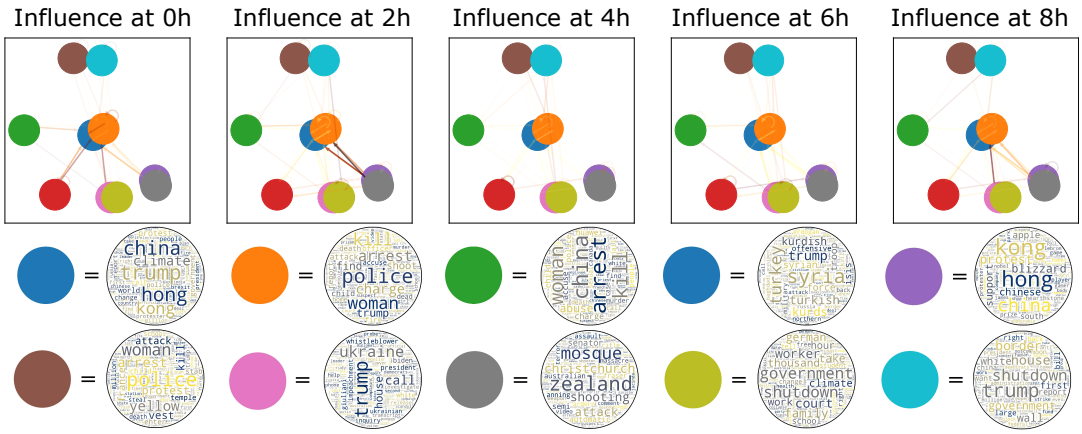

FIGURE IV.32: **Global clusters temporal interaction from 01/2019 to 12/2019** — Each node represents one cluster, and each edge represents the interaction strength at various times – the darker and the less transparent the stronger the effective interaction. The clusters composition is given below the network.

matrix and the transparency matrix are normalized by their highest value. We see that the strongest interaction happens between the blue and the orange clusters. Besides, this interaction seems to be stronger at smaller times. The interaction between other clusters is essentially directed toward these two clusters. Shortly after a news cluster appears, it is likely to find an echo on either r/news or r/worldnews. After existing for a while, subsequent publications are merged into one of these clusters depending on where they got published. This explains why most clusters last so few over time in Fig. IV.29.

We see that there are very few interactions between the clusters which account for actual news –in opposition to the clusters that account for a subreddit, see Section IV.6.4.b.

Monthly

In Fig. IV.33, we plot the same interaction between clusters as in Fig. IV.32, but for each month. This figure uses the same data as Fig. IV.29, except it is now broken down into temporal slices. From this figure, we recover that at the early times of the algorithm, MPDHP has not converged yet. It makes the interaction plot messy, with several interactions that are likely to be inaccurate. As time goes by, interactions are better defined –in particular between the blue and orange clusters. Another interesting fact is that most interactions happen in a 1h time frame: the publication of a document on r/news is highly likely to trigger a similar publication on r/worldnews within the hour. Here again, we see that most side clusters do not interact significantly with each other.

IV.6.5 Conclusion

A real-world application

In this section, we conducted extensive experiments on a single real-world large-scale dataset from Reddit. We described how we gather a year worth of news data and pre-processed it. We then argued how the dataset was fit to apply to MPDHP. We extensively discussed the various hyper-parameters that had to be tuned to run our experiments (in particular $\kappa(t)$, r and θ_0). In the end, we conducted 6 different experiments, one for each combination of parameters. The choice of other hyper-parameters has also been justified.

Interactions do not appear to play a significant role in this dataset

We proposed several ways to assess the role of interactions in the dataset. In particular, we introduced the notion of *effective interaction* as a way to evaluate how confident we can be in MPDHP's output. On this basis, we analysed the importance of interactions in general, as well as from a temporal perspective. We recovered our conclusions from Chapter II: interactions are sparse. We also recovered our conclusions from Chapter III: interaction strength decays over time. These observations emphasize the adequateness of our approach to model interactions. Building on these considerations and looking at the global effective interaction average, we conclude that interactions play a minor role in such a dataset. Overall, they only increase the instantaneous probability for a new observation to appear by 1%. Even the most extreme values represented in Fig. IV.30 seem to only increase this probability by 12% top.

FIGURE IV.33: **Monthly clusters temporal interaction from 01/2019 to 12/2019** — Each line stands for one month, each node represents one cluster, and each edge represents the interaction strength at various times – the darker and the less transparent the stronger the effective interaction. The cluster composition is given below the network.

Perspectives

However, despite intending our study as exhaustive, there is room for improvement in interaction modelling using MPDHP. In particular, there are two biases that we could not explore yet. Firstly, the parameter λ_0 has been set according to a heuristic (so that a new cluster is opened with fifty percent chances when we are 95% sure that it does not match the existing one). Its direct inference would robustify the approach, despite seeming complicated. Indeed, λ_0 does not account for individual events realizations, but for Hawkes processes starts, which is not a trivial difference. Secondly, another possible improvement would be to allow clusters to passively replicate –without the need for an interaction. We expect that this would boil down to adding a time-independent kernel entry to $\vec{\kappa}$, in a similar fashion as in Chapter III. However, non-technical questions may arise from such modification: when to consider a cluster as extinct given a non-fading kernel? How should this kernel relate to the temporal concentration parameter λ_0?

We believe such improvements would make MPDHP more robust and interpretable, and find applications beyond interaction modelling.

Chapter V

Conclusion

V.1 Contributions

V.1.1 Overview

Throughout this manuscript, we developed seven original models that allowed us to tackle various aspects of interactions between pieces of information in their spread: IMMSBM, SIMSBM, SDSBM, InterRate, PDP, PDHP, MPDHP. Along the road, this research led us to study seemingly distant fields of machine learning. A first venture in stochastic block modelling provided us with insights on the frequency at which interactions happen. An exploration of convex network inference methods allowed us to conclude on the time range of interactions. Driven by the need for more global models, a dive into fundamental bricks of machine learning led us to explore and modify canonical Dirichlet processes. Finally, answering our problematic required a junction between Dirichlet process and point processes. We recall the summary of our contributions in Table V.1 –this table is identical to Table I.2 and is simply recalled here.

V.1.2 Answers to our problematic

V.1.2.a Q1: how frequent are interactions? •

To answer our first question **Q1**, we considered Stochastic Block Modelling to be a good approach.

Indeed, a straightforward way to represent interactions is to embed them as a network. Entities are nodes of this network, and interactions between these entities are link between these nodes. These links can account for distinct types of relations that we can associate with different labels. Given the number of *possibly* interacting entities can be high in some real-world systems, we need to group them to see whether groups of entities interact in a similar way instead of dealing with each

TABLE V.1: **Contributions presented in this manuscript** — Our contributions are listed below in the order of their appearance in the text.

	SIMSBM Chap. II	IMMSBM Chap. II	SDSBM Chap. II	InterRate Chap. III	PDP Chap. IV	PDHP Chap. IV	MPDHP Chap. IV
Self-interactions	x	x	x	x	x	x	x
Pair-interactions	x	x	x	x			x
N-interactions	x		x				
Clustering	x	x	x		x	x	x
Discrete time			x	x		x	x
Continuous time				x		x	x
Online inference					x	x	x

entity individually. In practice, typically only a few of these possible interactions actually happen, but we did not know this beforehand. Besides, we considered that entities could interact as pairs, triplets or more, and that they could interact with different elements of their environment (time, user's metadata, etc.). After a careful review of the literature, we noticed that several state-of-the-art models could be expressed as special cases of a more global framework: the Serialized Interacting Mixed membership Stochastic Block Model (SIMSBM, Section II.2.2.a). This generalization allows modelling an arbitrarily large context (number of input information pieces) as well as an arbitrarily high order of interactions.

Along with this generalization, we introduced a simple procedure to incorporate time modelling into any iteration of SIMSBM. This extension takes the form of a prior on the model's parameters. This approach relies on the single assumption that dynamics are not abrupt.

We considered a special case of SIMSBM, SIMSBM(2) or Interacting Mixed Membership SBM (IMMSBM), and used it to model interactions between entities in several real-world datasets. We investigated the role of interactions between different spreading entities (hashtags, words, memes, etc.) and quantified their importance in several corpora –see Section II.2.3.

In particular, we focused on the study of a Twitter dataset and investigated the role of interaction between Twitter URLs on their spreading probability.

The conclusion of this study answered the first question **Q1** raised in the introduction: **interactions are sparse**. On the Twitter dataset, significant interactions took place only between a limited fraction of cluster pairs, and between an even smaller fraction of entity pairs.

V.1.2.b Q2: how persistent are interactions? ●

Despite the generality of the proposed SIMSBM, this model was not fit for modelling the time range over which interactions may take place. However, taking the interaction range into account is fundamental following a simple consideration: that the influence of entities on someone cannot last indefinitely as they get gradually forgotten by the users.

We introduced a convex multi-kernel model designed for this task: InterRate –see Chapter III. This model was able to infer the evolution of pair interactions' intensity over time, for any pair of entities, called the interaction profile. It allowed us to study how users exhibit different behaviours according to *combinations* of exposures they have been exposed to in continuous time. We showed that the joint effect of two exposures on a user is more than the disjoint sum of their individual effect, which means there is an interaction.

The answer of this study to the second question (**Q2**) raised in the introduction is that **interactions are brief**. A study of the processes at stake in the Twitter dataset revealed that interaction between two entities is significant only when those are close to each other in time. Typically, the intensity of an interaction decreases exponentially with the time separating the interacting entities.

V.1.2.c Q3: Can we efficiently model interactions? ● ●

Our third question raised the problem of efficiently model interactions. Designing a dedicated study to get insights into one aspect or the other of the interacting processes is one thing. Designing a scalable and useful method that answers the possible challenges raised by the task at hand is quite another.

The conclusions drawn from Chapter II and Chapter III are that interactions are sparse (we need clusters to model them) and that interactions are brief (we need to consider time). Besides, we discussed the fact that interacting entities may have a short lifespan, and that their semantic content must be taken into account. In Chapter IV, we introduced the steps paving our way to such a model.

Our approach was based on in-depth modifications of an existing Bayesian prior: the Dirichlet-Hawkes Process. This model was promising regarding our problem: it allowed us to create clusters (Q1) that can have a lasting influence in time on ulterior observations (Q2). Moreover, it could perform this task in linear time with the size of the dataset. However, it also suffered from several limitations. In particular, it was not fit to retrieve meaningful clusters when textual information conveyed little information or when temporal dynamics were hard to unveil. Furthermore, it assumed that the textual content of a document is perfectly correlated to its temporal dynamics, which cannot be rigorously true in real-world processes.

As an indirect way to overcome these limits, we explored the Powered Dirichlet Process as an alternative to the Dirichlet Process. This new prior alleviates the "rich-get-richer" property of vanilla Dirichlet processes.

We then incorporated this Powered Dirichlet Process into the standard Dirichlet-Hawkes process to create the Powered Dirichlet-Hawkes Process. This formulation improved the results when temporal information or textual content were weakly informative and alleviated the hypothesis that textual content and temporal dynamics were always perfectly correlated. Thus, our approach eventually allowed us to correctly model self-interactions within a cluster.

As a final improvement on the proposed approach, we extended the Powered Dirichlet-Hawkes Process to the multivariate case. This way, we can efficiently model interactions between all pairs of clusters, and by extension for all pairs of entities they comprise. The model can also run in constant time, and typically takes 300ms per document on large-scale real-world datasets.

We therefore provided a way to efficiently model interactions, positively answering our third introductory question (**Q3**). Efficiency can be understood both as technical efficiency and relevance efficiency. This former because our approach scales well on large real-world datasets and takes a minimal time to process a consequent amount of data. The latter because the final model can answer all the crucial challenges raised by interaction modelling, as it...

- considers entities' content. An entity is no more described as a unique identifier, but instead by its semantic content. Two entities that convey the same information are now considered as such and clustered together as a more global entity –a topic.

- models sparse interactions. Entities are now clustered together into temporal clusters. It makes it feasible to spot interaction terms between sets of entities. The lifespan of entities is no more a problem since clusters can comprise entities spanning over extended periods, which also increases the data available for each cluster.

- models dynamic interactions. Each cluster is associated with its own intensity function, which determines its effect on ulterior observations. Eventually, entities' influence fades away as time goes by.

- models multivariate interactions. Every cluster interacts as a pair with other clusters, enabling the study of interactions between the inferred topics.

V.1.2.d Q4: Do interactions play a significant role in spreading processes? •

We conducted extensive experiments on a real-world large-scale dataset made of one year of news headlines gathered from Reddit. This dataset seems fit for interaction modelling: contents are user-generated, dynamic, and mutating. We proposed several ways to assess the role of interactions in this specific dataset. The notion of effective interaction has been chosen as a privileged way to evaluate how much interaction has a role in the assumed data generation process. Firstly, we recovered our conclusions from Chapter II: interactions are sparse. Secondly, we recovered our conclusions from Chapter III: interaction strength decays over time. By looking at the average role of interactions in this dataset, we saw that interactions play a minor role in such a dataset. On average, interactions increase the instantaneous probability for a new observation to appear by 1% over its assumed random appearance probability. Without looking at the aggregated statistics, extreme values typically increase this probability by $\sim$10%, which does not hint toward strong interaction effects.

V.1.3 General uses for our models

Each of the models introduced in this manuscript has been presented as a novel way to tackle the interaction modelling problem. However, efforts have systematically been made to illustrate other, more general use cases. As a final note, we argue and detail some use cases of our approaches outside of interaction modelling.

V.1.3.a Powered Dirichlet Processes

The Dirichlet process (DP) is canonical in Bayesian nonparametric modelling. Enumerating the models based on DP is not feasible; among the most popular ones, we find the Latent Dirichlet Allocation model, the Infinite Gaussian Mixture Model, and the non-parametric MMSBMs and the Dirichlet-Hawkes process, all of which have seen numerous extensions depending on the use context. The popularity of Dirichlet processes is explained by the possibility to automatically infer the number of latent groups along with their composition, and the possibility to process data sequentially, in the order of arrival.

A current way to extend models based on DP is to make them hierarchical. Using the hierarchical DP, one can consider a mixture of partitions, by drawing the base distribution of a DP from another DP. Other extensions such as the Pitman-Yor process and the nested DP have been discussed in Section IV.3. By proposing the Powered DP (PDP), we paved the way for a whole new class of extensions. We illustrated the impact of redefining DP by adding a nonlinear dependence term. Possible applications for this advance would already comprise all existing models based on DP and models based on its extension, as the Powered DP is not exclusive. In the case of hierarchical DP for instance, the base distribution from which is drawn a Powered DP would also be drawn from a Powered DP, making the Powered hierarchical DP. Similar extensions and combinations are possible for the nested DP, the hierarchical nested DP, etc.

V.1.3.b Stochastic Block Models

When working on stochastic block models, our main goal was to design a framework (SIMSBM, Section II.2.2.a) that allows us to model almost any type of categorical data. It has been designed so that it can take as many information pieces

as needed as an input context, as many symmetric interactions between these entities as needed, and consider the temporal dimension to model dynamic cluster memberships (SDSBM extension, Section II.3). The flexibility provided by this work fits interaction modelling purposes, but also many other applications. The IMMSBM (Section II.2.3.a) has been accepted as a contribution to recommender systems, which is not explicitly about interactions modelling (Poux-Médard, Velcin, and Loudcher, 2021b).

Already discussed use cases in recommender systems include product recommendations on online retail websites, movies recommendation on streaming platforms, and songs on music streaming platforms. We also showed these models could be used to predict players' next move on real-world datasets and replicated studies of (Poux-Médard et al., 2021) that identified elementary players' behaviours. We argued for a possible application in automated medical diagnosis, where the combination of a few symptoms leads to high-quality diagnoses. We showed it could help predict the next tweet retweeted by Twitter users.

Further uses are encouraged in the field of humanities, where data often lies unexploited due to the lack of explainable, readily usable models. We illustrated a possible use case using a Latin epigraphy database. Projects using SIMSBM to infer the gender and age of antic remains given the tools and objects found in their tombs are currently ongoing. We strongly argued for such use in the SDSBM section. We paid particular attention to humanities recurrent challenges on data availability; our approach is shown to work even with scarce data.

The point is, that the field of applications for (dynamical) SIMSBM ranges way beyond simply interaction modelling. Dedicated studies using these new models may provide interesting results in social sciences, medicine and online recommendation.

V.1.3.c Dirichlet-Point processes

In Chapter IV and in this Conclusion, we extend the idea of Dirichlet-Hawkes processes to a broader class of models. The challenges inherent to Dirichlet-Hawkes processes have been answered by developing the Multivariate Powered Dirichlet-Hawkes process. This model can specifically be used to create a time-lined summary of event streams. At a time when internet content appears at an unprecedented pace, dedicated tools for big data summarizing will become increasingly necessary in many applications. Understanding ongoing trends on social platforms would help identify topics of interest, or simply provide these platform's users with an overview of what is going on. Several digital newspapers manually compile trends of interest that appeared over a day, a week; such tasks could be helped if not automated by such tools.

Another possible application specific to the MPDHP is the understanding of publication mechanisms. Significant research is being done in identifying and countering fake-news diffusion on social media. The tool we developed allows us to get insights into the way topics relate to each other. Dedicated studies on the interplay between fake-news topics and disclaimers could help develop countering strategies. Going one step further, we showed Dirichlet-Point processes could be used to make unsupervised modelling of topic-dependent spreading subnetworks. In the same line as before, automatically identifying fake news spreading subnetworks could help to surgically burst opinion bubbles by encouraging new links to other communities. As a more direct approach, it could help target specific nodes with disclaimers.

V.2 Perspectives

V.2.1 Towards more general block-modelling approaches

In Chapter II, we developed a global framework that allows us to model interactions of any given order using any size of context. We then proposed an extension to our method that allows us to consider time. We believe that this modelling flexibility can serve as a base to develop improved, even more flexible models to tackle a range of problems. We consider two possible extensions below.

V.2.1.a Considering time as a continuous variable

In Section II.3, we proposed a way to model time as an additional constraint to the stochastic block modelling approach. There, time is discrete and one model is inferred for each time slice, conditional on models from other time slices. However, we discussed in subsequent sections how slicing time in discrete intervals can induce biases in the modelling. A possible lead to alleviate this problem would be to merge our work on SIMSBM (Section II.2.2) with the Dirichlet-Point processes discussed in Chapter IV. In particular, SIMSBM uses a Dirichlet prior as *a priori* on its membership vectors. It happens that some works explored this direction to derive a non-parametric version of MMSBM, by expressing the Dirichlet prior as a Chinese Restaurant process (Fan, Cao, and Da Xu, 2015) –as what we did in Chapter IV. Once the MMSBM is expressed as a sequential Dirichlet process, it might be possible to include the advances in Dirichlet-Point processes as an explicit way to model time as a continuous variable. In general, this approach could make SIMSBM-based approaches fit to consider continuous data in general, provided it obeys an underlying point process.

V.2.1.b Considering nodes' metadata

Up to now, we modelled interactions at the level of the interacting entities themselves. Using our Stochastic Block Modelling framework, it is now possible to account for the entities' content and interaction time. However, this is not the most elegant way to consider the context in which a piece of information interacts with others. Recent works based on similar models proposed to model nodes' metadata as an additional layer, whose links can be activated or deactivated (Fajardo-Fontiveros, Guimerà, and Sales-Pardo, 2022). In this case, metadata does not have to be of a given type, not it is mandatory for it to be useful. The inclusion of these advances into the proposed framework would make a significant step towards an SBM that could be applied to any problem at hand with minimal model designing effort.

V.2.2 Improving the Multivariate Powered DHP

V.2.2.a Accounting for exogenous data generation

A consideration that we factored out from our analysis is the role of exogenous data generation. It has been underlined on some occasions that documents can get published according to dynamics external to the dataset. Its apparition could have been conditioned by other media sources (TV, radio, ...), social links not accounted for on Twitter or simply a demonstration of free will, and therefore should not be included in our dataset generative assumption. In (Myers, Zhu, and Leskovec, 2012), the authors model the rate of arrival of documents from such exogenous influence using temporal point processes. They conclude that the role of external influence is

not trivial. Supporting this claim, (He et al., 2015) introduced a term that accounts for exogenous events in Hawkes-based modelling. The fact that both works make extensive use of temporal point processes in their modelling suggests that the inclusion of their findings into MPDHP is doable and would certainly yield insightful results.

V.2.2.b Going further than Dirichlet-Hawkes processes

In Chapter IV, we studied in-depth the Dirichlet-Hawkes process and various models that can be built on them. However, the full picture is broader than simply the association of Dirichlet processes and Hawkes processes. Instead, the method described throughout Chapter IV can be applied to merge any Dirichlet process (hierarchical, nested, or powered) or variants (Indian Buffet Process, Pitman-Yor process, etc.) with any point process (Hawkes, Cox, Poisson, Determinantal, Geometric, etc.). We believe that the resulting Dirichlet-Point processes are powerful tools that can adapt to many modelling problems. This field of such combinations has been little explored up to now and may offer interesting insights for further studies.

V.2.3 Considering the network structure

In this manuscript, we considered both the content and the dynamics of spreading entities to unveil interactions. However, our models are not fit to consider the structure of the network entities spread on. An interesting lead to explore is how interactions between information pieces differ at the user level, depending on their position in the spreading network.

Both as a final perspective and as an illustration of a broader use for Dirichlet-Point processes, we propose to develop a model that considers the network structure based on Dirichlet-Point processes. We sketch a model that can jointly infer dynamic (Chapter III) clusters (Chapter II) of textual documents (Chapter IV) spreading online *and* the subnetworks they spread along. We did not find any previous attempt to *jointly* infer these parameters, by using an iteration of the Dirichlet-Point processes discussed in this conclusion.

V.2.3.a Possible lead: Dirichlet-Survival process

We can bridge the gap between network inference and dynamic clustering models by defining the **Dirichlet-Survival process**.

Network inference as a survival process

In (Gomez-Rodriguez, Leskovec, and Schölkopf, 2013a), the authors demonstrate that most of the then-existing underlying network inference models can be derived as a special case of a global framework. Without entering the details here, it is shown that using a specific formulation of Survival processes allows retrieving network inference models such as NetRate (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011), KernelCascade (Du et al., 2012), MoNet (Wang, Ermon, and Hopcroft, 2012), InfoPath (Gomez-Rodriguez, Leskovec, and Schölkopf, 2013b), etc.

Each of these models can be characterized by a single intensity function, similar to the Hawkes intensity discussed in Chapter IV, named the hazard rate. For the interested reader, we detail how to get to this function for the NetRate model (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011) in Appendix C.

We write this intensity function $\lambda(t_i^c|t_j^c, \alpha_{j,i})$. It represents the instantaneous probability for a node i to be infected at time t_i^c by an infection c because of the node j that got infected by c at time t_j^c. The strength of the link between i and j is encoded in $\alpha_{j,i}$. As this intensity function results from a survival point-process similar to the Hawkes process, we will see that we can substitute it to the counts of a Dirichlet process to create a Dirichlet-Survival process.

Dirichlet-Survival prior

As in (Du et al., 2015; Mavroforakis, Valera, and Gomez-Rodriguez, 2017; Tan, Rao, and Neville, 2018) and what we did throughout Chapter IV, we will create a new Dirichlet-Point process by merging the Dirichlet Process in its Chinese Restaurant iterative metaphor, to a Point process such as the one characterizing NetRate. We write the intensity of such process $\lambda(t_i^c|t_j^c, \alpha_{j,i})$.

Doing so essentially breaks down the temporal dynamics at the level of the network's nodes' in-going edges. Each edge is now associated with its own point process. This makes a yet unexplored bridge between Dirichlet processes and Survival analysis.

Instead of considering a single network on which data spreads, we assume there is any number of such subnetworks, each associated with a cluster. Instead of associating one point process to one cluster as in Chapter IV, we associate a collection of point processes to one cluster –one per edge in the network. We write $\lambda(t_i^c|t_j^c, \alpha_{j,i}^{(k)})$ the intensity associated with the edge between i and j given their infection times by c are separated of a time $\Delta t = t_j - t_i$ cluster k.

An infection event from cascade c is now assumed to have a given probability of being trigger by any of the k existing clusters (or subnetworks) on which information spread. By substituting the counts in the Dirichlet process with the total intensity on node i due to all its neighbours $\mathcal{H}_{i,c}$ that got infected earlier by c in subnetwork k, noted $\Lambda_i^{c,(k)} = \sum_{j \in \mathcal{H}_{i,c}} \lambda(t_i^c|t_j^c, \alpha_{j,i}^{(k)})$. Let λ_0 be the probability that the observation did not get triggered by any existing subnetwork –the concentration parameter.

$$P(s_n = k|\mathcal{H}_{i,c}) = \begin{cases} \frac{\Lambda_i^{c,(k)}}{\lambda_0^{(K+1)} + \Lambda_i^{c,(k)}} & \text{if k = 1, ..., K} \\ \frac{\lambda_0^{(K+1)}}{\lambda_0^{(K+1)} + \Lambda_i^{c,(k)}} & \text{if k = K+1} \end{cases} \tag{V.1}$$

TABLE V.2: Numerical results of Dirichlet-Survival process, TopicCascade, Dirichlet-Hawkes process and NetRate models. The AUC, F1 score and MAE are computed considering every top cluster's edges at once so there is no error to report.

		Dir-Surv	TC	DHP	NetRate
ER	NMI	**0.787**	0.711	0.638	-
	ARI	**0.631**	0.488	0.411	-
	AUC	**0.849**	0.800	-	0.659
	F1	**0.263**	0.176	-	0.005
	MAE	**0.229**	0.278	-	0.481

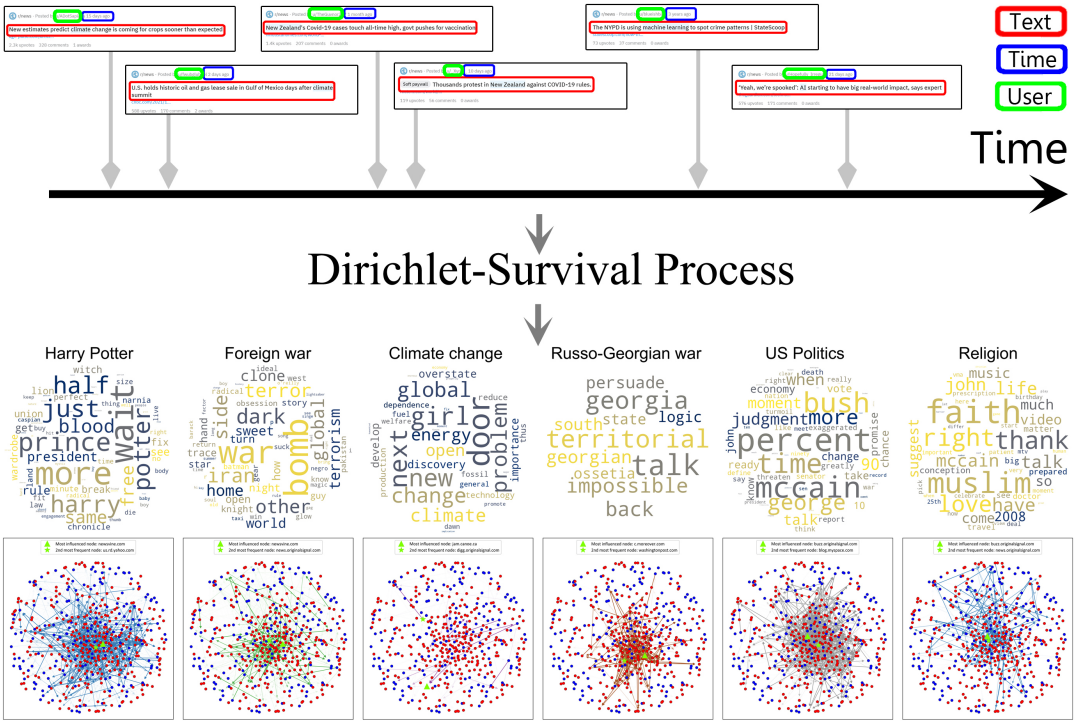

FIGURE V.1: Dirichlet-survival process applied to real data. (Top row) The most frequent words within the inferred cluster. (Bottom row) The inferred subnetworks associated with each cluster. Mass media nodes are represented in red, and blogs nodes are represented in blue. The most influenced node is the one having the highest total in-going transmission rate. The nodes' positions are identical for every network to ease comparison.

Some preliminary experimental results

Similarly to what we did in Chapter IV, we coupled the Dirichlet-Survival prior to a Dirichlet-Multinomial language model. In this section, we sketch some experimental results as a way to support the relevance of further studies using Dirichlet-Point processes, especially for modelling interactions in information spread.

We run the Dirichlet-Survival process on synthetic data first. We generate an Erdös-Renye network (**ER**) (Erdős and Rényi, 1960), on which we simulate information spread using a Dirichlet-Survival process. Using 500 nodes, we create 5 subnetworks of size of approximately 250 nodes and 700 edges, one per different topic.

We present the results obtained using Dirichlet-Survival processes in Table V.2. We compare to several models: TC (Du et al., 2013), DHP (Du et al., 2015) and NetRate (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011). The NetRate model infers edges without taking clusters or textual content into account. The DHP considers the dynamics and the textual content, but not the network's structure. The TC model first infers textual clusters and then infers one subnetwork for each of them. Overall, we see that accounting for all three data types (content, time, and structure) increases the model's performance.

We finally illustrate a possible use case on real-world data in Fig. V.1. We used data from the Memetracker dataset (Leskovec, Backstrom, and Kleinberg, 2009).

V.2.3.b Perspectives on interaction modelling

We briefly showed a possible use case for Dirichlet-Point processes in information spread modelling by defining the Dirichlet-Survival process. In this case, interactions are not modelled since cascades spread independently from each other.

However, it paves the way for defining even more complex Dirichlet-Point processes that consider time, content and structure of information spread in interaction modelling. In our case, substituting the non-interacting cascades model with a more elaborated underlying network inference model might allow us to uncover interaction mechanisms. In particular, recent years have seen some works tackling the network inference problem by assuming a Hawkes process on each edge of a network. Considering a multivariate Hawkes process instead, similarly to what we did Section IV.5 would also be an interesting lead.

It should be noted, however, that the Dirichlet-Survival process introduced in this chapter may not be the best approach to model such phenomena. In particular, comparing its performances to (He et al., 2015; Barbieri, Manco, and Ritacco, 2017) is needed to get solid results on this specific application. However, the mere fact that we can almost instantly elaborate new models to tackle new problems argues in favour of the use of Dirichlet-Point processes in a broader context.

V.3 Final words

The work presented in this manuscript is the outcome of three years of questioning, exploring, and discovering various aspects of interactions at stake in information spread. As it seems to be the norm in research, this initial question served as a guiding thread. A thread that sewed this manuscript through large and disconnected areas of the machine learning canvas: stochastic block models, dynamic networks inference, Dirichlet processes, temporal point processes. Here, our developments over those pieces are used to answer our problematic. However, we essentially focused on improving the backbone of these areas, whose specific application to interaction modelling yields interesting insights. From a broader perspective, our work is also intended as a contribution to machine learning in general; Formulating alternative Dirichlet Processes, granting flexibility and dynamism to block models, merging Dirichlet and Point processes, have implications that range beyond solely interactions modelling. Our efforts resulted in explainable, scalable and flexible methods that can tackle a wide range of problems. It is our sincere hope that the advances presented in this manuscript will be of greater help for researchers of all horizons, either to improve over them or to use them as tools for answering concrete real-world questions.

Bibliography

Acar, Evrim et al. (2010). "Scalable Tensor Factorizations with Missing Data". In: *SDM10: Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 701–712.

Ahmed, Amr and E. Xing (2008). "Dynamic Non-Parametric Mixture Models and the Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering". In: *SIAM International Conference on Data Mining*, pp. 219–230.

Ahmed, Amr and Eric P. Xing (2007). "Seeking The Truly Correlated Topic Posterior - on tight approximate inference of logistic-normal admixture model". In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. Proceedings of Machine Learning Research 2, pp. 19–26.

Airoldi, Edoardo et al. (2008). "Mixed Membership Stochastic Blockmodels". In: *Journal of Machine Learning Research* 9, pp. 1991–1992.

AlSumait, Loulwah, Daniel Barbará, and Carlotta Domeniconi (2008). "On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking". In: pp. 3–12.

Arratia, Richard, A. D. Barbour, and Simon Tavaré (1992). "Poisson Process Approximations for the Ewens Sampling Formula". In: *The Annals of Applied Probability* 2.3, pp. 519–535.

Bacry, Emmanuel et al. (2017). "Tick: A Python Library for Statistical Learning, with an Emphasis on Hawkes Processes and Time-Dependent Models". In: *J. Mach. Learn. Res.* 18.1, pp. 7937–7941.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations*. ICLR.

Barbieri, Nicola, Francesco Bonchi, and G. Manco (2012). "Topic-aware social influence propagation models". In: *Knowledge and Information Systems* 37, pp. 555–584.

Barbieri, Nicola, Giuseppe Manco, and Ettore Ritacco (2017). "Survival Factorization on Diffusion Networks". In: *Machine Learning and Knowledge Discovery in Databases*, pp. 684–700.

Bassiou, Nikoletta K. and Constantine L. Kotropoulos (2014). "Online PLSA: Batch Updating Techniques Including Out-of-Vocabulary Words". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.11, pp. 1953–1966.

Baumgartner, Jason et al. (2020). "The Pushshift Reddit Dataset". In: *Proceedings of the International AAAI Conference on Web and Social Media* 14.1, pp. 830–839.

Bereby-Meyer, Y. and A. E. Roth (2006). "The speed of learning in noisy games: Partial reinforcement and the sustainability of cooperation". In: *American Economic Review* 96.4, pp. 1029–1042.

Beutel, Alex et al. (2012). "Interacting viruses in networks: Can both survive?" In: *Proceedings of the ACM SIGKDD*.

Bhargava, Preeti et al. (2015). "Who, What, When, and Where: Multi-Dimensional Collaborative Recommendations Using Tensor Factorization on Sparse User Generated Data". In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15, pp. 130–140.

Bishop, Christopher (2006). *Pattern Recognition and Machine Learning*. Springer, pp. 450–455.

Blei, David M. and Peter Frazier (2010). "Distance Dependent Chinese Restaurant Processes". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10, pp. 87–94.

Blei, David M. and John D. Lafferty (2006). "Dynamic Topic Models". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06, pp. 113–120.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3, pp. 993–1022.

Bourigault, Simon, Sylvain Lamprier, and Patrick Gallinari (2016). "Apprentissage de représentations probabilistes pour la prédiction de diffusions d'informations sur les réseaux sociaux". In: *CORIA-CIFED*.

Cao, Bin et al. (2007). "Detect and Track Latent Factors with Online Nonnegative Matrix Factorization". In: *IJCAI*.

Cao, Junyu and Wei Sun (2019). "Sequential choice bandits: learning with marketing fatigue". In: *AAAI-19*.

Carroll, J.D. and JJ. Chang (1970). "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition". In: *Psychometrika* 35, pp. 283–319.

Chen, Feng and Wai Hong Tan (2018). "Marked Self-Exciting Point Process Modelling of Information Diffusion on Twitter". In: *The Annals of Applied Statistics* 12, pp. 2175–2196.

Christakopoulou, E. and G. Karypis (2014). "HOSLIM: Higher-Order Sparse LInear Method for Top-N Recommender Systems". In: *Advances in Knowledge Discovery and Data Mining*.

Clauss, Manfred et al. (2021). *Epigraphik-Datenbank Clauss Slaby*.

Cobo-López, S., A. Godoy-Lorite, and J. Duch (2018). "Optimal prediction of decisions and model selection in social dilemmas using block models". In: *EPJ Data Sci* 7(48).

Du, Nan et al. (2012). "Learning networks of heterogeneous influence". In: *NIPS* 4, pp. 2780–2788.

Du, Nan et al. (2013). "Uncover Topic-Sensitive Information Diffusion Networks". In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. AISTATS 31, pp. 229–237.

Du, Nan et al. (2015). "Dirichlet-Hawkes Processes with Applications to Clustering Continuous-Time Document Streams". In: *21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Erdős, P. and A Rényi (1960). *On the Evolution of Random Graphs*, pp. 17–61.

Fajardo-Fontiveros, Oscar, Roger Guimerà, and Marta Sales-Pardo (2022). "Node Metadata Can Produce Predictability Crossovers in Network Inference Problems". In: *Phys. Rev. X* 12 (1), p. 011010.

Fan, Fenglei et al. (2021). "On Interpretability of Artificial Neural Networks: A Survey". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 5, pp. 741–760.

Fan, Xuhui, Longbing Cao, and Richard Yi Da Xu (2015). "Dynamic Infinite Mixed-Membership Stochastic Blockmodel". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.9, pp. 2072–2085.

Ferguson, Thomas S. (1973). "A Bayesian Analysis of Some Nonparametric Problems". In: *The Annals of Statistics* 1.2, pp. 209 –230.

Filipović, Marko and Ante Jukić (2015). "Tucker Factorization with Missing Data with Application to Low-nn-Rank Tensor Completion". In: *Multidimensional Syst. Signal Process.* 26.3, pp. 677–692.

Fung, Yik-Hing, Chun-Hung Li, and William K. Cheung (2007). "Online Discussion Participation Prediction Using Non-Negative Matrix Factorization". In: *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*. WI-IATW '07, pp. 284–287.

Funke, Thorben and Till Becker (2019). "Stochastic block models: A comparison of variants and inference methods". In: *PloS one*.

Ghitza, Yair and Andrew Gelman (2013). "Deep Interactions with MRP: Election Turnout and Voting Patterns Among Small Electoral Subgroups". In: *American Journal of Political Science* 57.

Ghosh, Soumya et al. (2014). "Nonparametric Clustering with Distance Dependent Hierarchies". In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. UAI'14, pp. 260–269.

Godoy-Lorite, Antonia, Roger Guimerà, and Marta Sales-Pardo (2016). "Long-Term Evolution of Email Networks: Statistical Regularities, Predictability and Stability of Social Behaviors". In: *PLOS ONE* 11.1, pp. 1–11.

Godoy-Lorite, Antonia et al. (2016). "Accurate and scalable social recommendation using mixed-membership stochastic block models". In: *PNAS* 113.50, pp. 14207–14212.

Goldwater, Sharon, Thomas L. Griffiths, and Mark Johnson (2011). "Producing Power-Law Distributions and Damping Word Frequencies with Two-Stage Language Models". In: *JMLR* 12.68.

Gomez-Rodriguez, Manuel (2013). *Structure and Dynamics of Diffusion Networks*. PhD thesis.

Gomez-Rodriguez, Manuel, David Balduzzi, and Bernhard Schölkopf (2011). "Uncovering the Temporal Dynamics of Diffusion Networks". In: *ICML*, pp. 561–568.

Gomez-Rodriguez, Manuel, Krishna P. Gummadi, and Bernhard Schölkopf (2014). "Quantifying Information Overload in Social Media and its Impact on Social Contagions". In: *ICWSM*.

Gomez-Rodriguez, Manuel, Jure Leskovec, and Bernhard Schölkopf (2013a). "Modeling Information Propagation with Survival Theory". In: *ICML* 28, pp. 666–674.

— (2013b). "Structure and Dynamics of Information Pathways in Online Media". In: *WSDM*.

Guimera, Roger, Alejandro Llorente, and Marta Sales-Pardo (2012). "Predicting Human Preferences Using the Block Structure of Complex Social Networks". In: *PLOS One* 7.9.

Guimerà, Roger and Marta Sales-Pardo (2013). "A Network Inference Method for Large-Scale Unsupervised Identification of Novel Drug-Drug Interactions". In: *PLoS Comput Biol*.

Gutiérres-Roig, Mario et al. (2016). "Market Imitation and Win-Stay Lose-Shift Strategies Emerge as Unintended Patterns in Market Direction Guesses". In: *PLOS One*.

Hanson, John William (2016). *An urban geography of the Roman world, 100 BC to AD 300*. Vol. 18. Archaeopress Oxford.

Hanson J. W.and Ortman, S.G. and J. Lobo (2017). "Urbanism and the division of labour in the Roman Empire". In: *Journal of The Royal Society Interface* 14.136, p. 20170367.

Harper, F. Maxwell and Joseph A. Konstan (2015). "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4.

Harshman, R. A. (1970). "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis". In: *UCLA Working Papers in Phonetics* 16, pp. 1–84.

He, Xinran et al. (2015). "HawkesTopic: A Joint Model for Network Inference and Topic Modeling from Text-Based Cascades". In: *ICML*.

Hidasi, Balázs and Domonkos Tikk (2012). "Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback". In: *Machine Learning and Knowledge Discovery in Databases*, pp. 67–82.

Ho, Qirong, Le Song, and Eric P. Xing (2011). "Evolving Cluster Mixed-Membership Blockmodel for Time-Evolving Networks". In: *AISTATS*.

Hodas, Nathan O. and Kristina Lerman (2014). "The Simple Rules of Social Contagion". In: *Scientific Reports* 4.4343.

Holland, Paul W., Kathryn Blackmond Laskey, and Samuel Leinhardt (1983). "Stochastic blockmodels: First steps". In: *Social Networks* 5.2, pp. 109–137.

Huan, Zhao et al. (2017). "Meta-graph based recommendation fusion over heterogeneous information networks". In: *SIGKDD*.

Ishwaran, Hemant and Lancelot James (2003). "Generalized weighted Chinese restaurant processes for species sampling mixture models". In: *Statistica Sinica* 13, pp. 1211–1235.

Iwata, Tomoharu et al. (2009). "Topic Tracking Model for Analyzing Consumer Purchase Behavior". In: *Proceedings of the 21st International Jont Conference on Artifical Intelligence*. IJCAI'09, pp. 1427–1432.

Jamali, Mohsen, Tianle Huang, and Martin Ester (2011). "A Generalized Stochastic Block Model for Recommendation in Social Rating Networks". In: RecSys '11, pp. 53–60.

Jensen, S. and J. Liu (2008). "Bayesian clustering of transcription factor binding motifs". In: *Journal of the American Statistical Association* 103, pp. 188–200.

Jin, D. et al. (2021). "A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 01, pp. 1–1.

Karatzoglou, Alexandros et al. (2010). "Multiverse Recommendation: N-Dimensional Tensor Factorization for Context-Aware Collaborative Filtering". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10, pp. 79–86.

Karsai, Márton, Hang-Hyun Jo, and Kimmo Kaski (2018). *Bursty Human Dynamics*. Springer.

Kempe, David, Jon Kleinberg, and Éva Tardos (2003). "Maximizing the Spread of Influence through a Social Network". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 137-146.

Khurshid, Anwer, Mohammad Ageel, and Rizwan Lodhi (2005). "On Confidence Intervals for the Negative Binomial Distribution". In: *Revista Investigacion Operacional* 26, pp. 59–70.

Klema, V. and A. Laub (1980). "The singular value decomposition: Its computation and some applications". In: *IEEE Transactions on Automatic Control* 25.2, pp. 164–176.

Koren, Y., R. Bell, and C. Volinsky (2009). "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8, pp. 30–37.

Kumar, Srijan, Xikun Zhang, and Jure Leskovec (2019). "Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks". In: *Proceedings of the 25th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Lagnier, Cédric et al. (2013). "Predicting Information Diffusion in Social Networks Using Content and User's Profiles". In: *Proceedings of the 35th European Conference on Advances in Information Retrieval*. ECIR'13, pp. 74–85.

Larremore, Daniel B. et al. (2012). "Statistical properties of avalanches in networks". In: *Phys. Rev. E* 85 (6), p. 066131.

Lee, Clement and Darren J. Wilkinson (2019). "A review of stochastic block models and extensions for graph clustering". In: *Applied Network Science* 4, pp. 1–50.

Leskovec, Jure, Lars Backstrom, and Jon Kleinberg (2009). "Meme-Tracking and the Dynamics of the News Cycle". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09, pp. 497–506.

Lijoi, Antonio, Ramsés H. Mena, and Igor Prünster (2007). "Controlling the reinforcement in Bayesian non-parametric mixture models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.4, pp. 715–740.

Loreto, Vittorio et al. (2016). "On the Emergence of Syntactic Structures: Quantifying and Modeling Duality of Patterning". In: *Topics in Cognitive Science* 8.2, pp. 469–480.

Markines, Benjamin and Filippo Menczer (2009). "A Scalable, Collaborative Similarity Measure for Social Annotation Systems". In: pp. 347–348.

Matias, C, T Rebafka, and F Villers (June 2018). "A semiparametric extension of the stochastic block model for longitudinal networks". In: *Biometrika* 105.3, pp. 665–680. ISSN: 0006-3444.

Matias, Catherine and Vincent Miele (2017). "Statistical clustering of temporal networks through a dynamic stochastic block model". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.4, pp. 1119–1141.

Mavroforakis, Charalampos, Isabel Valera, and Manuel Gomez-Rodriguez (2017). "Modeling the Dynamics of Learning Activity on the Web". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17, pp. 1421–1430.

McCarthy, Davis J., Y. Chen, and G. Smyth (2012). "Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation". In: *Nucleic Acids Research* 40, pp. 4288 –4297.

McDowell, Ian C et al. (2018). "Clustering gene expression time series data using an infinite Gaussian process mixture model". In: *PLoS computational biology* 14.1, e1005896.

Munafò, Marcus et al. (Jan. 2017). "A manifesto for reproducible science". In: *Nature Human Behaviour* 1, p. 0021.

Myers, Seth A. and J. Leskovec (2012). "Clash of the Contagions: Cooperation and Competition in Information Diffusion". In: *2012 IEEE 12th International Conference on Data Mining*, pp. 539–548.

Myers, Seth A., Chenguang Zhu, and Jure Leskovec (2012). "Information Diffusion and External Influence in Networks". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12, pp. 33–41.

Nay, John J. and Yevgeniy Vorobeychik (2016). "Predicting Human Cooperation". In: *PLoS One* 11.5.

Neal, Radford M. and Geoffrey E. Hinton (1998). "A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants". In: *Learning in Graphical Models*. Dordrecht: Springer Netherlands, pp. 355–368.

Nielsen, Frank (2020). "On a Generalization of the Jensen-Shannon Divergence and the Jensen-Shannon Centroid". In: *Entropy* 22, p. 221.

Pastor-Satorras, Romualdo et al. (2015). "Epidemic processes in complex networks". In: *Reviews of Modern Physics* 87.3, pp. 925–979.

Pitman, Jim and Marc Yor (1997). "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator". In: *The Annals of Probability* 25.2, pp. 855 – 900.

Popper, Karl Raimund (1935). *The Logic of Scientific Discovery*. London, England: Routledge.

Posadas Duran, Juan et al. (May 2019). "Detection of fake news in a new corpus for the Spanish language". In: *Journal of Intelligent and Fuzzy Systems* 36, pp. 4869–4876.

Poux-Médard, Gaël, Romualdo Pastor-Satorras, and Claudio Castellano (2020). "Influential spreaders for recurrent epidemics on networks". In: *Phys. Rev. Research* 2 (2).

Poux-Médard, Gaël, Julien Velcin, and Sabine Loudcher (2021a). "Information Interaction Profile of Choice Adoption". In: *Machine Learning and Knowledge Discovery in Databases (ECML-PKDD). Research Track*, pp. 103–118.

— (2021b). "Information Interactions in Outcome Prediction: Quantification and Interpretation Using Stochastic Block Models". In: *Fifteenth ACM Conference on Recommender Systems (RecSys)*, 199–208.

— (2021c). "Powered Hawkes-Dirichlet Process: Challenging Textual Clustering using a Flexible Temporal Prior". In: *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 509–518.

Poux-Médard, Gaël et al. (2021). "Complex decision-making strategies in a stock market experiment explained as the combination of few simple strategies". In: *EPJ Data Science* 10 (26).

Prakash, B. et al. (2012). "Winner Takes All: Competing Viruses or Ideas on fair-play Networks". In: *WWW*.

Qin, Z. S. et al. (2003). "Identification of co-regulated genes through Bayesian clustering of predicted regulatory binding sites". In: *Nature Biotechnology* 21, pp. 435–439.

Rashed, Ahmed et al. (2020). "MultiRec: A Multi-Relational Approach for Unique Item Recommendation in Auction Systems". In: *Fourteenth ACM Conference on Recommender Systems*. RecSys '20, pp. 230–239.

Rasmussen, Carl Edward (1999). "The Infinite Gaussian Mixture Model". In: NIPS'99, pp. 554–560.

Rathore, Mohit, Dikshant Gupta, and Dinabandhu Bhandari (2018). "Complaint Classification using Word2Vec Model". In: *International Journal of Engineering and Technology(UAE)* 7, pp. 402–404.

Saito, Kazumi et al. (2009). "Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis". In: 5828, pp. 322–337.

Saito, Kazumi et al. (2011). "Learning Diffusion Probability Based on Node Attributes in Social Networks". In: pp. 153–162.

Senanayake, Ransalu, Simon O'Callaghan, and Fabio Ramos (2016). "Predicting Spatio-Temporal Propagation of Seasonal Influenza Using Variational Gaussian Process Regression". In: *AAAI*, pp. 3901–3907.

Sethuraman, J. (1994). "A constructive definition of Dirichlet priors". In: *Statistica sinica* 4.4, pp. 639–650.

Shi, Chuan et al. (2016). "A survey of heterogeneous information network analysis". In: *IEEE Transactions on Knowledge and Data Engineering*, pp. 17–37.

Socher, Richard, Andrew Maas, and Christopher Manning (2011). "Spectral Chinese Restaurant Processes: Nonparametric Clustering Based on Similarities". In: *JMLR - Proceedings* 15, pp. 698–706.

Steck, Harald and Dawen Liang (2021). "Negative Interactions for Improved Collaborative Filtering: Don't Go Deeper, Go Higher". In: *Fifteenth ACM Conference on Recommender Systems*, pp. 34–43.

Sudderth, Erik and Michael Jordan (2009). "Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes". In: *NIPS* 21.

Sun, Yizhou (2011). "PathSim: meta path-based top-K similarity search in heterogeneous information networks". In: *VLDB*.

Sun, Yizhou and Jiawei Han (2012). "Mining Heterogeneous Information Networks: A Structural Analysis Approach". In: *SIGKDD Explorations* 14.

Tan, X., Vinayak A. Rao, and Jennifer Neville (2018). "The Indian Buffet Hawkes Process to Model Evolving Latent Influences". In: *UAI*.

Tang, Xuning and Christopher C. Yang (2014). "Detecting Social Media Hidden Communities Using Dynamic Stochastic Blockmodel with Temporal Dirichlet Process". In: *ACM Trans. Intell. Syst. Technol.* 5.2.

Tarrés-Deulofeu, Marc et al. (2019). "Tensorial and bipartite block models for link prediction in layered networks and temporal networks". In: *Phys. Rev. E* 99 (3), p. 032307.

Teh, Yee and Dilan Gorur (2009). "Indian Buffet Processes with Power-law Behavior". In: *Advances in Neural Information Processing Systems* 22.

Villermet, Quentin et al. (2021). "Follow the Guides: Disentangling Human and Algorithmic Curation in Online Music Consumption". In: *Fifteenth ACM Conference on Recommender Systems*, pp. 380–389.

Vosoughi, Soroush, Deb Roy, and Sinon Aral (2018). "The spread of true and false news online". In: *Science*, pp. 1146–1151.

Wallach, Hanna et al. (2010). "An alternative prior process for nonparametric Bayesian clustering". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 892–899.

Wang, Liaoruo, Stefano Ermon, and John E. Hopcroft (2012). "Feature-Enhanced Probabilistic Models for Diffusion Network Inference". In: *Machine Learning and Knowledge Discovery in Databases*, pp. 499–514.

Wang, Wei et al. (2019). "Coevolution spreading in complex networks". In: *Physics Reports* 820. Coevolution spreading in complex networks, pp. 1–51.

Wang, Xuerui and Andrew McCallum (2006). "Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06, pp. 424–433.

Welling, Max (2006). "Flexible priors for infinite mixture models". In: *Workshop on learning with non-parametric Bayesian methods*.

Weng, L., A. Flammini, and al. (2012). "Competition among memes in a world with limited attention". In: *Nature Sci Rep 2* 335.

Wilks, S.S. (1992). *Mathematical statistics*. John Wiley, section 7.

Wilson, James D., Nathaniel T. Stevens, and William H. Woodall (2019). "Modeling and detecting change in temporal networks via the degree corrected stochastic block model". In: *Qual. Reliab. Eng. Int.* 35, pp. 1363–1378.

Wu, Xunxun et al. (2019). "Dynamic Stochastic Block Model with Scale-Free Characteristic for Temporal Complex Networks". In: *Database Systems for Advanced Applications: 24th International Conference, (DASFAA)*, pp. 502–518.

Xing, Eric P., Wenjie Fu, and Le Song (2010). "A state-space mixed membership blockmodel for dynamic network tomography". In: *The Annals of Applied Statistics* 4, pp. 535–566.

Xu, Hongteng and Hongyuan Zha (2017). "A Dirichlet Mixture Model of Hawkes Processes for Event Sequence Clustering". In: *Advances in Neural Information Processing Systems* 30.

Xu, Kevin S. and Alfred O. Hero (2013). "Dynamic Stochastic Blockmodels: Statistical Models for Time-Evolving Networks". In: *Social Computing, Behavioral-Cultural Modeling and Prediction*, pp. 201–210.

— (2014). "Dynamic Stochastic Blockmodels for Time-Evolving Social Networks". In: *IEEE Journal of Selected Topics in Signal Processing* 8, pp. 552–562.

Xu, W., Y. Li, and J. Qiang (2021). "Dynamic clustering for short text stream based on Dirichlet process". In: *Appl Intell*.

Yang, Tianbao et al. (2010). "Detecting communities and their evolutions in dynamic social networks —a Bayesian approach". In: *Machine Learning* 82, pp. 157–189.

Yates, F. (1935). *The Design of Experiments*. Oliver and Boyd.

Yin, Jianhua and Jianyong Wang (2014). "A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14, pp. 233–242.

Yin, Jianhua et al. (2018). "Model-Based Clustering of Short Text Streams". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18, pp. 2634–2642.

Yu, Linyun et al. (2017). "A Temporally Heterogeneous Survival Framework with Application to Social Behavior Dynamics". In: *KDD'17*, pp. 1295–1304.

Yu, Ming, Varun Gupta, and Mladen Kolar (2017). "An Influence-Receptivity Model for Topic Based Information Cascades". In: *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 1141–1146.

Yuchung, J. Wang and Y. Wong George (1987). "Stochastic Blockmodels for Directed Graphs". In: *Journal of the American Statistical Association* 82.397, pp. 8–19.

Zarezade, Ali et al. (2017). "Correlated Cascades: Compete or Cooperate". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 238–244.

Zhou, Xuezhong et al. (2014). "Human symptoms-disease network". In: *Nature communications* 5, p. 4212.

Zhu, Z., C. Gao, and Y Zhang (2020). "Cooperation and Competition among information on social networks". In: *Nature Sci Rep* 3103, p. 12160.

Appendix A

Appendix – Stochastic Block Models

I.1 SIMSBM - Additional experimental results

TABLE A.1: Replication results on two datasets used in (Godoy-Lorite et al., 2016) and (Poux-Médard et al., 2021), referenced in the main text. The standard error on the last digits over all 100 runs is indicated in standard notation – $0.123(12) \Leftrightarrow 0.123 \pm 0.012$. Overall, we retrieve the same results as those presented in (Godoy-Lorite et al., 2016) and (Poux-Médard et al., 2021). The models presented in this chapter are underlined.

			F1	P@1	AUCROC	AUCPR	RAP	NCE
Imdb	**User, Movie**	SIMSBM(1,1)	**0.3995(2)**	**0.3558(3)**	**0.7665(1)**	**0.3406(3)**	**0.5805(2)**	**0.1593(1)**
		TF	0.2570	0.2348	0.5031	0.1541	0.4627	0.2573
		KNN	0.2668	0.2002	0.5558	0.1735	0.3308	0.4834
		NB	0.2585	0.2382	0.5377	0.1660	0.4664	0.2536
		BL	0.2570	0.2349	0.5000	0.1525	0.4647	0.2557
MrBanks	**Ply, Full sit**	SIMSBM(1,1)	**0.7126(2)**	**0.6688(4)**	**0.7126(3)**	**0.7180(4)**	**0.8344(2)**	**0.1656(2)**
		TF	0.6795	0.6037	0.5176	0.5363	0.8019	0.1981
		KNN	0.6940	0.6433	0.6668	0.6430	0.8217	0.1783
		NB	0.6795	0.6037	0.5907	0.5822	0.8019	0.1981
		BL	0.6795	0.6037	0.5000	0.5215	0.8019	0.1981

I.2 IMMSBM - Datasets

I.2.1 Medical records

The Pubmed dataset collect has been inspired by (Zhou et al., 2014). Every article on PubMed is manually annotated by experts with a list of keywords describing the main topics of the publication. We downloaded a list of 322 symptoms and 4,442 diseases provided by (Zhou et al., 2014). Then, we used the PubMed API to query each one of the symptom/disease keywords aforementioned. For each result, we got a list of every publication in which the keyword is among the main topics. Then, we build the dataset by considering every publication in which there is at least one symptom and one disease. Finally, we create the triplets (symptom1, symptom2, disease) by looking at all the pairs of symptoms in an article and linking each one of them to all the diseases observed in the same article. In the end, we are left with a total of 52,833,690 observed triplets, distributed over 15,809,271 PubMed publications.

I.2.2 Spotify

The Spotify dataset has been collected using the Spotify API. We randomly sampled 2,000 playlists using the keywords "english" and "rock", which corresponds to a total of 135,100 songs. Then, for each playlist, we used a running window of 4 songs to build the dataset. The artist of the song immediately after the running window is the output we aim at predicting, x, and the artists of the 4 songs within the running window are the interacting inputs. Once again, we consider all the possible pairs of artists in the running window and associate them to the output artist x. Note that we only considered the artists that appear more than 50 times in the whole dataset, for the sake of statistical relevance. The resulting dataset is consists of 1,236,965 triplets for 2,028 artists.

I.2.3 Twitter

We gathered the Twitter dataset used in (Hodas and Lerman, 2014). It consists in a collection of all the tweets containing URLs posted during the month of October 2010. A first operation consisted in cleaning the dataset of URLs that are considered as aggressive advertising. To do so, we considered only the URLs whose retweets built a chain of length at least 50; this choice comes from the idea that commercial spams are not likely to be retweeted by actual users and therefore do not create chains. Secondly, we considered only the users who have not tweeted a given URL more than 5 times, this behavior being an activity typical of spamming bots. Doing so, we are left with tweets that are mostly coming from the non-commercial activity of human users. Then, we follow a dataset building process similar to (Myers and Leskovec, 2012). For each user, we slice her feed + tweets temporal sequence in intervals separated by the tweets of the user. Every time a user tweets something, the interval ends. An interval therefore consists of the tweet of the user and all the tweets she has been exposed to right before tweeting. Following the suggestion of (Myers and Leskovec, 2012), we only consider the 3 last tweets the user has been exposed to before retweeting one of them. Each one of these intervals form an entry of our message+answer dataset (3 last entries in the feed + next tweets). In the end, we are left with 284,837 intervals containing a total of 2,110 different tweeted URLs. Our dataset then consists of 1,181,543 triplets.

I.2.4 Reddit

Finally, we downloaded the May 2019 Reddit dataset from the data repository pushshift.io, which stores regular saves of the comments posted on the website. We chose to consider only the comments made in the subreddit r/news. Reddit's comments system work as a directed tree network, where each answer to a given comment initiates a new branch. We considered pairs of messages such as one (the answer) has for direct parent the other one (the comment). We then extracted all the named entities in both of them using the Spacy Python library. For each pair of named entities in the comment, we associated every name entity in the answer. We consider here only the named entities that appear at least 200 times in the subreddit, for the same reason as for the Spotify dataset. The final dataset results in 35,364,725 triplets for a total of 1,656 named entities.

I.3 IMMSBM - Upper limit to predictions

We derive an analytical expression for the upper-limit to our model for a given dataset. Explicitly, we analytically maximize the likelihood according to each entry of the dataset.

We enforce the constraint that the sum over the output space of probabilities given any observations made has to sum to 1. To do so, we use a Lagrange multiplier λ_{obs} for every different observation (in the case of our model: for every different triplet). The log-likelihood then takes the following form:

$$\begin{aligned} \ell &= \sum_{(obs,x)} \ln P_{obs}(x) - \sum_{obs} \lambda_{obs} (\sum_{x} P_{(obs,x)} - 1) \\ \Leftrightarrow \frac{\partial \ell}{\partial P_{obs_i}(x_i)} &= \sum_{\partial(obs_i,x_j)} \frac{1}{P_{obs_i}(x_j)} - \lambda_{obs_i} = 0 \\ \Leftrightarrow P_{obs_i}(x_j) &= \frac{1}{\lambda_{obs_i}} \sum_{\partial(obs_i,x_j)} 1 \end{aligned} \tag{A.1}$$

Where obs_i correspond to any given couple of inputs (i,j) in the model presented in the main paper. We use the following notation: $\partial(obs_i, x_j) = \{obs | (obs_i, x_j) \in R^\circ\}$, with R° the dataset entries. Therefore, we can define $\sum_{\partial(obs_i,x_j)} 1 \equiv N_{obs_i,x_j}$ the number of times obs_i appears jointly with x_j in the dataset. We are now looking for the λ_{obs_i} maximizing the likelihood:

$$\begin{aligned} \frac{\partial \ell}{\partial \lambda_{obs_i}} &= \sum_{x} P_{obs_i}(x) - 1 = 0 \\ \Leftrightarrow \sum_{x} \frac{N_{obs_i,x}}{\lambda_{obs_i}} &= 1 \\ \Leftrightarrow \lambda_{obs_i} &= \sum_{x} N_{obs_i,x} \end{aligned} \tag{A.2}$$

Finally, plugging Eqs.A.1 and A.2 together, we obtain:

$$P_{obs_i}(x_j) = \frac{N_{obs_i,x_j}}{\sum_x N_{obs_i,x}} \tag{A.3}$$

This equation gives the probability maximizing the likelihood for any entry of the dataset. In the main model, it translates to $P_{(i,j)}(x) = N_{(i,j),x} / \sum'_x N_{(i,j),x'}$ with $N_{(i,j),x}$ the number of times output x has been witnessed after a pair of inputs (i,j). Note that this is simply the frequency of an output given a pair of input entities.

Keep in mind that the result we just derived gives perfect predictions only for a particular dataset, and therefore and has no global predictive value. This tool is useful when it comes to assess the performance of other models' results, but is unusable in prediction tasks.

I.4 SDSBM - Explicit derivation of the E-step

I.4.1 Short derivation

This demonstration can be found in (Godoy-Lorite et al., 2016; Tarrés-Deulofeu et al., 2019; Poux-Médard, Velcin, and Loudcher, 2021b). We recall the log-likelihood

as defined in the main paper:

$$\begin{aligned}
\log P(\theta, p|R^\circ) &\propto \log P(R^\circ|\theta, p) \prod_t \prod_i P(\theta_i^{(t)}) \prod_k P(p_k^{(t)}) \\
&= \sum_{(i,o,t)\in R^\circ} \log \sum_{k\in K} \theta_{i,k}^{(t)} p_k^{(t)}(o) \\
&\quad + \sum_t \sum_i \log P(\theta_i^{(t)}) \sum_k \log P(p_k^{(t)}) \\
&\geq \sum_{(i,o,t)\in R^\circ} \sum_{k\in K} \omega_{i,o}^{(t)}(k) \log \frac{\theta_{i,k}^{(t)} p_k^{(t)}(o)}{\omega_{i,o}^{(t)}(k)} \\
&\quad + \sum_t \sum_i \log P(\theta_i^{(t)}) \sum_k \log P(p_k^{(t)})
\end{aligned} \tag{A.4}$$

In Eq.A.4, we introduced a proposal distribution $\omega_{i,o}^{(t)}(k)$, that represents the probability of one cluster allocation k given the observation (i, o, t). The last line followed from Jensen's inequality assuming that $\sum_k \omega_{i,o}^{(t)}(k) = 1$. We notice that Jensen's inequality holds as an equality when:

$$\omega_{i,o}^{(t)}(k) = \frac{\theta_{i,k}^{(t)} p_k^{(t)}(o)}{\sum_{k'} \theta_{i,k'}^{(t)} p_{k'}^{(t)}(o)} \tag{A.5}$$

which provides us with the expectation formula. The prior terms $P(\theta_i^{(t)})$ and $P(p_k^{(t)})$ have no effect on the result as they cancel in the inequality A.4.

I.4.2 Full derivation

The derivation presented in this section follows a well-known general derivation of the EM algorithm, which can be found in C.M. Bishop's *Pattern Recognition and Machine Learning*-p.450 for instance.

We recall that one entry of the dataset R° takes the form of a tuple (i, o, t), where i is the input item and o an associated label at time t. $k \in K$ denotes the latent variable accounting for cluster allocation among K possible values. The total log-likelihood is the sum of each observation's log-likelihood. Without loss of generality, we focus on a single observation (i, o, t). The expression of the log-posterior distribution for one observation reads:

$$\begin{aligned}
&\log P(\theta^{(t)}, p^{(t)}|(i, o, t)) \\
&\propto \log P(R^\circ|\theta^{(t)}, p^{(t)}) P(\theta_i^{(t)}) \prod_k P(p_k^{(t)}) \\
&= \log P^{(t)}(i, o|\theta^{(t)}, p^{(t)}) + \log P(\theta_i^{(t)}) + \sum_k \log P(p_k^{(t)})
\end{aligned} \tag{A.6}$$

For an observation $(i, o, t) \in R^\circ$, we assume a distribution $Q_{i,o}^{(t)}(k)$ on the latent variables associated to it; this distribution is yet to be defined. Because k takes values among K possible ones, we have $\sum_{k\in K} Q_{i,o}^{(t)}(k) = 1$. Given this normalization condition, we can decompose each summed term in Eq.A.6 for any distribution $Q_{i,o}^{(t)}(k)$ as:

$$
\begin{aligned}
&\log P^{(t)}(i,o|\theta^{(t)},p^{(t)}) \\
&= \underbrace{\log P^{(t)}(i,o,k|\theta^{(t)},p^{(t)}) - \log P^{(t)}(k|i,o,\theta^{(t)},p^{(t)})}_{\text{Does not depend on } k} \\
&= \sum_{k\in K} Q_{i,o}^{(t)}(k) \log P^{(t)}(i,o,k|\theta^{(t)},p^{(t)}) \\
&\quad - \sum_{k\in K} Q_{i,o}^{(t)}(k) \log P^{(t)}(k|i,o,\theta^{(t)},p^{(t)}) \\
&= \sum_{k\in K} Q_{i,o}^{(t)}(k) \log \frac{P^{(t)}(i,o,k|\theta^{(t)},p^{(t)})}{Q_{i,o}^{(t)}(k)} \\
&\quad - \sum_{k\in K} Q_{i,o}^{(t)}(k) \log \frac{P^{(t)}(k|i,o,\theta,p^{(t)})}{Q_{i,o}^{(t)}(k)}
\end{aligned}
\tag{A.7}
$$

We note that the term in the last line of Eq.A.7, $-\sum_{k\in K} Q_{i,o}^{(t)}(k) \log \frac{P^{(t)}(k|i,o,\theta^{(t)},p^{(t)})}{Q_{i,o}^{(t)}(k)}$, is the Kullback-Leibler (KL) divergence between $P^{(t)}$ and $Q_{i,o}^{(t)}$, noted $KL(P^{(t)}||Q_{i,o}^{(t)})$. The KL divergence obeys $KL(P^{(t)}||Q_{i,o}^{(t)}) \geq 0$, and is null iif $P^{(t)}$ equals $Q_{i,o}^{(t)}$. Therefore, the term in the before-last line of Eq.A.7, $\sum_{k\in K} Q_{i,o}^{(t)}(k) \log \frac{P^{(t)}(i,o,k|\theta^{(t)},p^{(t)})}{Q_{i,o}^{(t)}(k)}$, is interpreted as a lower bound on the log-likelihood $\log P^{(t)}(i,o|\theta^{(t)},p^{(t)})$.

The aim of the E-step is to find the expression of $Q_{i,o}^{(t)}(k)$ that maximizes the lower bound of the log-likelihood with respect to the latent variables k. Given that the log-likelihood does not depend on $Q_{i,o}^{(t)}(k)$ and $KL(P^{(t)}||Q_{i,o}^{(t)}) \geq 0$, the lower-bound is maximized when $KL(P^{(t)}||Q_{i,o}^{(t)}) = 0$, which occurs when $Q_{i,o}^{(t)}(k) = P^{(t)}(k|i,o,\theta^{(t)},p^{(t)})$. In this case, the lower-bound on the log-likelihood equals the likelihood itself and thus reaches a global maximum with respect to the latent variables k for fixed parameters $\theta^{(t)}$ and $p^{(t)}$.

Given the definition of our simple model, the derivation of $P(k|i,o,\theta^{(t)},p^{(t)})$ is straightforward. The probability of one combination of clusters k among K possible combinations given an input features vector and an output o is proportional to $p_k^{(t)}(o)\theta_{i,k}$. Therefore:

$$
P^{(t)}(k|i,o,\theta^{(t)},p^{(t)}) = \frac{p_k^{(t)}(o)\theta_{i,k}^{(t)}}{\sum_{k'\in K} p_{k'}^{(t)}(o)\theta_{i,k'}^{(t)}} \tag{A.8}
$$

which is the expression of $\omega_{i,o}^{(t)}(k)$ in the main article.

I.5 SDSBM - Explicit derivation of the M-step for p

$$
\begin{aligned}
&\frac{\partial\left(\log P(\theta,p|R^\circ)-\sum_{k',t'}\psi_{k'}^{(t')}(\sum_o p_{k'}^{(t')}(o)-1)\right)}{\partial p_k^{(t)}(o)}=0\\
&\Leftrightarrow \sum_{(i,t)\in\partial o}\frac{\omega_{i,o}^{(t)}(k)}{p_k^{(t)}(o)}+\frac{\beta\langle p_k^{(t)}(o)\rangle}{p_k^{(t)}(o)}-\psi_k^{(t)}=0\\
&\Leftrightarrow \sum_{(i,t)\in\partial o}\omega_{i,o}^{(t)}(k)+\beta\langle p_k^{(t)}(o)\rangle=\psi_k^{(t)}p_k^{(t)}(o)\\
&\Leftrightarrow \sum_{(i,t)\in\partial o}\sum_o\omega_{i,o}^{(t)}(k)+\beta\underbrace{\sum_o\langle p_k^{(t)}(o)\rangle}_{=1}=\psi_k^{(t)}\\
&\Leftrightarrow \frac{\sum_{(i,t)\in\partial o}\omega_{i,o}^{(t)}(k)+\beta\langle p_k^{(t)}(o)\rangle}{\sum_{(i,o,t)\in R^\circ}\omega_{i,o}^{(t)}(k)+\beta}=p_k^{(t)}(o)
\end{aligned}
\tag{A.9}
$$

I.6 SDSBM - Using the prior in related works

Throughout this section, we highlight the changes brought by our method to the EM equations derived in the mentioned papers. In summary, we see that out method allows to make these works dynamic with minimal changes of the original models.

I.6.1 Bi-MMSBM

In (Godoy-Lorite et al., 2016), the authors apply a MMSBM to a labeled bipartite network. The nodes on each side of the bipartite network are associated to their own membership matrix; membership of nodes $i \in I$ over K clusters is encoded into $\theta \in \mathbb{R}^{I\times K}$, and membership of nodes $j \in J$ over L clusters is encoded into $\eta \in \mathbb{R}^{J\times L}$. The block-interaction matrix for the label $o \in O$ is noted $p(o) \in \mathbb{R}^{K\times L}$.

Assuming a temporal version, items i and j to be linked by a label o at time t reads:

$$
P^{(t)}(o|i,j)=\sum_{k\in K}\sum_{l\in L}\theta_{i,k}^{(t)}\eta_{j,l}^{(t)}p_{k,l}^{(t)}(o) \tag{A.10}
$$

Given the same set of observation s R° as in the main article, the posterior distribution follows:

$$
\begin{aligned}
P(\theta,\eta,p|R^\circ) &= P(R^\circ|\theta,\eta,p)\\
&\times\prod_t\left(\prod_i P(\theta_i^{(t)})\prod_j P(\eta_j^{(t)})\prod_{k,l}P(p_{k,l}^{(t)})\right)
\end{aligned}
\tag{A.11}
$$

such that:

$$P(R^\circ|\theta,\eta,p) = \prod_{(i,j,t,o)\in R^\circ} \sum_{k\in K}\sum_{l\in L} \theta^{(t)}_{i,k}\eta^{(t)}_{j,l}p^{(t)}_{k,l}(o) \tag{A.12}$$

$$P(\theta^{(t)}_i) \propto \prod_k {\theta^{(t)}_{i,k}}^{\beta\langle\theta^{(t)}_{i,k}\rangle} \tag{A.13}$$

$$P(\eta^{(t)}_j) \propto \prod_l {\eta^{(t)}_{j,l}}^{\beta\langle\eta^{(t)}_{j,l}\rangle} \tag{A.14}$$

$$P(p^{(t)}_{k,l}) \propto \prod_o {p^{(t)}_{k,l}(o)}^{\beta\langle p^{(t)}_{k,l}(o)\rangle} \tag{A.15}$$

where $\langle x^{(t)}\rangle = \frac{\sum_{t'\neq t}\kappa(t,t')x^{(t')}}{\sum_{t'\neq t}\kappa(t,t')}$. The expectation step is not influenced by the priors choice and is the same as in (Godoy-Lorite et al., 2016) for each temporal slice. The new maximization steps are:

$$\theta^{(t)}_{i,k} = \frac{\sum_l\sum_{(o,j)\in\partial(i,t)}\omega^{(t)}_{i,j,o}(k,l)+\beta\langle\theta^{(t)}_{i,k}\rangle}{N_{i,t}+\beta}$$

$$\eta^{(t)}_{j,l} = \frac{\sum_k\sum_{(o,i)\in\partial(j,t)}\omega^{(t)}_{i,j,o}(k,l)+\beta\langle\eta^{(t)}_{j,l}\rangle}{N_{j,t}+\beta}$$

$$p^{(t)}_{k,l}(o) = \frac{\sum_{(i,j,t)\in\partial o}\omega^{(t)}_{i,j,o}(k,l)+\beta\langle p^{(t)}_{k,l}(o)\rangle}{\sum_{(i,j,o,t)\in R^\circ}\omega^{(t)}_{i,j,o}(k,l)+\beta}$$

Here again, β is set fixed for demonstration purposes, but can be tuned at will by the user. This allows to choose the extent to which dynamics shall be smoothed, or ignored.

I.6.2 T-MBM

The T-MBM is essentially the same model as (Godoy-Lorite et al., 2016) but with one type of entry that can appear twice in one observation. Both entries share the same membership matrix θ. The probability of a label of type o given entries h; i and j at time t is now written:

$$P(o|h,i,j,t) = \sum_{k\in K}\sum_{l\in L}\sum_{m\in M}\theta^{(t)}_{h,k}\theta^{(t)}_{i,l}\eta^{(t)}_{j,m}p^{(t)}_{k,l,m}(o) \tag{A.16}$$

The posterior distribution follows the same expression as in Eq.A.11. The expectation step is left unchanged by the choice of the priors, and the new maximization

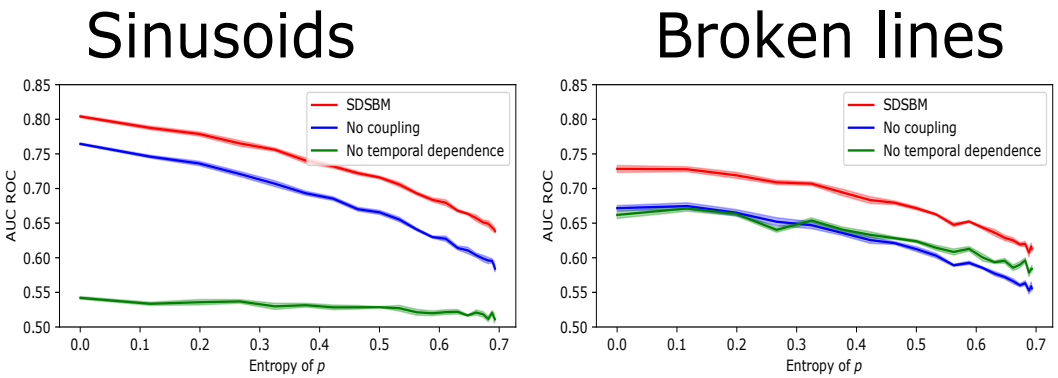

FIGURE A.1: Experimental results when inferring both θ and p jointly. The AUC-ROC is as good as when p is provided to the model.

equations are given below:

$$\theta^{(t)}_{h,k} = \frac{\sum_{l,m}\sum_{(o,i,j)\in\partial(h,t)}\omega^{(t)}_{h,i,j,o}(k,l,m)+\beta\langle\theta^{(t)}_{h,k}\rangle}{N_{h,t}+\beta}$$

$$\eta^{(t)}_{i,l} = \frac{\sum_{k,m}\sum_{(o,h,j)\in\partial(i,t)}\omega^{(t)}_{h,i,j,o}(k,l,m)+\beta\langle\eta^{(t)}_{i,l}\rangle}{N_{i,t}+\beta}$$

$$p^{(t)}_{k,l,m}(o) = \frac{\sum_{(h,i,j,t)\in\partial o}\omega^{(t)}_{h,i,j,o}(k,l,m)+\beta\langle p^{(t)}_{k,l,m}(o)\rangle}{\sum_{(h,i,j,o,t)\in R^\circ}\omega^{(t)}_{h,i,j,o}(k,l,m)+\beta}$$

I.7 SDSBM - Inferring two dynamic matrices of parameters

In the main article, p is provided to the model and only θ has to be inferred. Doing so, we can confront inferred membership vectors to the ground truth while avoiding label-switching issues (Matias and Miele, 2017; Lee and Wilkinson, 2019). When p is also to be inferred, finding a correspondence between the inferred clusters and the ground-truth is not a trivial task, and cannot be performed in unbiased ways. However, the good results yielded by the model, presented in Fig.A.1, when also inferring p hints that the membership vectors are correctly inferred.

I.8 SDSBM - Clusters composition for the Epigraphy experiment

- Cluster 0
  - Roma (98.0%)
- Cluster 1
  - Latium et Campania (32.0%)
  - Venetia et Histria (14.0%)
  - Samnium (11.0%)
  - Umbria (10.0%)
  - Apulia et Calabria (8.0%)

- Cluster 2
  - Pannonia superior (15.0%)
  - Dalmatia (11.0%)
  - Noricum (10.0%)
  - Hispania citerior (7.0%)
  - Gallia Narbonensis (6.0%)
- Cluster 3
  - Dacia (24.0%)
  - Pannonia inferior (17.0%)
  - Moesia inferior (15.0%)
  - Syria (6.0%)
  - Numidia (6.0%)
  - Pannonia superior (5.0%)
- Cluster 4
  - Germania superior (24.0%)
  - Mauretania Caesariensis (11.0%)
  - Asia (11.0%)
  - Etruria (11.0%)
  - Galatia (9.0%)

Appendix B

Appendix – Temporal diffusion networks

II.1 Implementation of Clash of the Contagions

In this appendix, we provide technical details on the way the Clash of Contagions baseline is implemented. Following the directions given in the reference article (Myers and Leskovec, 2012), we implemented a Stochastic Gradient Descent (SGD) method for parameters inference. Given the small number of entities considered in the experiments, each iteration of the SGD is computed using the full dataset instead of slicing it into mini-batches.

II.1.1 Setup

For each corpus, we run the SGD algorithm 100 times, from which we save the parameters maximizing the likelihood the most. At the beginning of each run, parameters M and Δ are randomly initialized. The stopping condition makes the algorithm ends when the relative variation of the likelihood according to the last iteration is been lesser than 10^{-6} for more than 30 times in a row; those numbers have been chosen empirically to maximize the performances of the algorithm. The hyper-parameters have been set to: T=5 (number of clusters) and K=20 (number of considered time steps).

II.1.2 Update rule

In each iteration, we update the parameters in the direction of the gradient descent (noted G). However, a major problem when dealing with SGD is to choose the line step length η (the amplitude of the variation of the parameters in the direction of the gradient G). After each iteration, we compare several update rules, and we select the one maximizing the likelihood. Those rules are as follows:

- AdaGrad: $\eta^{AG} \times G$
- AdaDelta: $\eta^{AD} \times G$
- Line search in the direction of the gradient: $\eta^{LS} \times G$
- Line search in the direction of AdaDelta: $\eta^{LS} \times \eta^{AD} \times G$

The line search snippets consider 50 values of η^{LS} logarithmically distributed in the interval $[10^{-6}; 10^{3}]$.

II.1.3 Constraints on the parameters

The membership vectors entries $M_{i,t}$ (membership of i to cluster t) must be positive and sum to 1 over all the clusters ($\sum_t M_{i,t} = 1$). In order to enforce this constraint, we consider the following variable change: $M_{i,t} \rightarrow \frac{\phi_{i,t}^2}{\sum_{t'} \phi_{i,t'}^2}$. This transformation guarantees the membership vector properties with no need for penalty methods in the implementation.

Besides, as stated in (Myers and Leskovec, 2012), it can happen that a probability is larger then 1 or lesser than 0. In the absence of complementary details in the main paper, we implemented our own method to force the probabilities to stay within reasonable bounds. Here it is impossible to make a simple variable change to enforce this constraint, since the probability results of a non-linear combination of the model's parameters. We added to the likelihood an exponential penalty term. Let P denote a quantity we want to constrain between 0 and 1. The penalty term equals $e^{-\lambda P} + e^{\lambda(P-1)}$. λ here is a parameter that tunes the intensity of the penalty, and is empirically set to $\lambda = 75$. This penalty function has the form of a well with very steep walls in x=0 and x=1. In this way, it seldom happens that probabilities are larger than 1 or lesser than 0, as said in the main article. When such cases happen, we simply set it back to the closest bound for methods comparisons.

Appendix C

Appendix – Dirichlet-Survival Process

III.1 Deriving NetRate

The input of NetRate is a collection of observed cascades $C = \{\vec{c}\}_{\vec{c}=\vec{c_1},\vec{c_2},...}$. Each cascade is a collection of events with timestamps $\vec{c} = \{(u_i^c, t_i^c)\}_i$, where u_i^c is the node on which the i^{th} event occurred and t_i^c the time at which it happened in cascade c. Using survival analysis, (Gomez-Rodriguez, Balduzzi, and Schölkopf, 2011) denotes the likelihood of an infection of node u_i^c at time t_i^c in cascade c by any other node u_j^c previously infected at time t_j^c in the same cascade as $f(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c})$, where $\alpha_{u_j^c,u_i^c}$ is an entry of the objective network's adjacency matrix. In survival analysis' framework, $f(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c})$ is linked to the instantaneous infection rate (or hazard rate) of u_i^c at time t_i^c by u_j^c previously infected at time t_j^c, noted $\lambda(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c})$, and the probability of non-infection of u_i^c up to time t_i^c by u_j^c previously infected at time t_j^c, noted $S(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c})$, by the following relation:

$$f(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) = \lambda(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) S(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) \tag{C.1}$$

Within a cascade, the likelihood that a node get infected by only one neighbour u_j^c previously infected at time t_j^c can be written as the likelihood of infection at time t_i^c by u_j^c previously infected at time t_j^c times its probability of survival to every previously infected node:

$$f(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) \prod_{k \neq j, t_k^c < t_i^c} S(t_i^c|t_k^c, \alpha_{u_k^c,u_i^c}) \tag{C.2}$$

The likelihood of an infection by any neighbour then becomes the sum of those candidate disjoint events:

$$\begin{aligned} &\sum_{j, t_j^c < t_i^c} f(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) \prod_{k \neq j, t_k^c < t_i^c} S(t_i^c|t_k^c, \alpha_{u_k^c,u_i^c}) \\ \overset{\text{Eq.C.1}}{=} &\sum_{j, t_j^c < t_i^c} \lambda(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) \prod_{t_k^c < t_i^c} S(t_i^c|t_k^c, \alpha_{u_k^c,u_i^c}) \end{aligned} \tag{C.3}$$

The likelihood of a cascade then becomes the product of the likelihood of every event it contains, and the total likelihood $\mathcal{L}(C|A)$ of every cascade the product over every cascade. Let A be the network's adjacency matrix whose entries $\vec{\alpha_{i,j}}$ are directed edges from i to j. Then:

$$\mathcal{L}(C|A) = \prod_{\vec{c} \in C} \prod_{t_i^c \in \vec{c}} \sum_{j, t_j^c < t_i^c} \lambda(t_i^c|t_j^c, \alpha_{u_j^c,u_i^c}) \prod_{t_k^c < t_i^c} S(t_i^c|t_k^c, \alpha_{u_k^c,u_i^c}) \tag{C.4}$$

List of Figures

List of Tables

List of Equations

Acronyms

A.R.Prof. Associate Research Professor. i

AP Average Precision. 56

AUCPR Area Under the Precision-Recall Curve. 30

AUCROC Area Under the Receiving Operator Curve. 30, 38, 53, 56

Bi-MMSBM Bipartite Mixed Membership Stochastic Block Model. 19, 31

BL Baseline. 30

CP Canonical Polyadic Decomposition. 16

CRP Chinese Restaurant Process. 46, 83, 103

D.R. Director of Research. i

DHP Dirichlet-Hawkes Process. 11, 81, 83, 106, 128

DP Dirichlet Process. 11, 81, 86–88, 100, 102, 128, 152

EM Expectation-Maximization Algorithm. 20, 83

IMMSBM Interacting Mixed Membership Stochastic Block Model. 13, 27, 31, 32, 71, 149, 150

JS Jensen-Shannon divergence. 70

KNN K-nearest-neighbours. 31, 99

MAE Mean Absolute Error. 112

MMSBM Mixed Membership Stochastic Block Model. 26, 31, 37, 43, 154

MPDHP Multivariate Powered Dirichlet-Hawkes Process. 11, 81, 121, 123, 131, 133, 146, 148, 149, 153, 155

MSE Mean Squared Error. 70

NB Naive Bayes. 30

NCE Normalized Coverage Error. 30, 56

NMF Non-negative Matrix Factorization. 15, 31

NMI Normalized Mutual Information. 102, 108–110, 119, 128

P@ Precision at. 30

PDHP Powered Dirichlet-Hawkes Process. 11, 81, 101, 102, 104, 106, 107, 149

PDP Powered Dirichlet Process. 11, 81, 86, 90, 102, 149, 152

Prof. Professor. i

PY Pitman-Yor Process. 90

RAP Rank Average Precision. 30

RBF Radial basis function kernel. 68, 123, 126, 128, 135

RMSE Root Mean Squared Error. 53

RSS Residual Sum of Squares. 70

SBM Stochastic Block Model. 10, 16, 19, 43, 154

SIMSBM Serialized Interacting Mixed Membership Stochastic Block Model. 13, 20, 32, 57, 149, 150, 152, 154

SMC Sequential Monte-Carlo. 83, 105, 124, 131, 136

SVD Singular Value Decomposition. 16

T-MBM Tensorial Mixed Membership Stochastic Block Model. 19, 31

TF Tensor Factorization. 31

UP Uniform Process. 86, 88, 100, 102, 107, 128

Glossary

cascade A series of identical decisions taken by different spreaders about an entity. For instance, a tweet that gets retweeted n times makes a cascade of size n+1.. 3, 4, 7, 8, 64, 65, 71, 83, 156, 158, 189

cluster Group formed from a set of objects in such a way that objects in the same group are more similar (in some sense) to each other than to those in other groups.. xi, 2, 7, 11, 13, 14, 17–24, 26–28, 30–34, 36, 38, 40–47, 50, 55–58, 72, 79–133, 135–148, 150, 151, 153, 155–157, 181, 182, 190, 193, 195

collaborative filtering A method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). Its underlying assumption is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.. 15, 19, 20

content Mean through which entities carry a semantic meaning: text, pixels, sounds, etc.. 1, 2, 4, 19, 33, 36, 76, 79–82, 85, 86, 102, 105, 109–111, 115, 120, 121, 129, 131, 133, 134, 138, 140, 144, 151–155, 157, 158, 181, 194

decision The action of a spreader when facing an entity. The decision can be endogenous (the spreader acts on the entity alone, e.g. by sharing it, liking it, clicking on it, reacting to it, etc.) or exogenous (the spreader creates a new entity as a consequence of the first one, e.g. a denial, an answer to a mail, to a tweet, etc.).. 2, 8, 14, 61, 73

diffusion A set of individual decisions. For instance: a retweet cascade, an internet buzz, a Youtube trend, etc.. ix, x, 3–5, 7–10, 14, 18, 56, 61, 62, 64, 66, 68, 70, 72, 74, 76, 153, 189, 191

entity Same as "piece of information". Any object that is susceptible to have an influence on spreaders. It carries a semantic meaning. For instance: a tweet, a meme, a song, a virus, a news article, etc.. 2–5, 8–10, 13–15, 17–23, 25–28, 30–43, 47, 58, 59, 61–66, 69, 70, 72, 73, 75–77, 79–83, 87, 120, 121, 130, 132, 149–151, 153–155, 181, 191

independent As opposed to interacting, when there is no interaction; the spreading entities do not affect each other. The whole equals the sum of its parts.. 3, 4, 20, 21, 31, 37, 39, 45, 46, 65, 71, 82, 84, 87, 91, 93, 97, 101, 107, 113, 121, 126, 181, 189

infected When spreaders take a decision on an entity, they are said to be infected by this entity. Infection denotes the transition between a "Susceptible" state and an "Infected" state.. 3, 6, 65, 156

interacting As opposed to independent, when there is an interaction; the spreading entities can affect each other. The whole does not equal the sum of its parts.. ix, xi, 3, 5–7, 13, 14, 19, 21, 26, 28, 30–33, 39, 41–43, 58, 59, 61–64, 66, 69, 71, 72, 76, 77, 79, 80, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 130, 133, 149–151, 154, 158, 181, 190, 191, 193, 194

interaction When several entities are interacting, name given when the whole does not equal the sum of its parts.. ix–xii, 2–11, 13–59, 61–77, 79–82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110–112, 114, 116, 118, 120–122, 124, 126, 128, 130–136, 138, 140–155, 157, 158, 181, 182, 189–192, 195

interaction profile A tool for visualizing the strength of interactions between entities as the time separating them grows.. x, 61–66, 71, 74–76, 81, 150, 181, 192

membership The extent to which an entity belongs to a cluster.. x, 2, 13, 15, 19, 21–23, 26, 27, 32–34, 36, 42–59, 71, 83, 150, 153, 154, 190, 191

outcome A possibly exogenous decision that has been taken given a specific context.. 14, 15, 18–20, 40, 41, 62, 63, 70, 72, 73, 158

piece of information Same as “entity”. Any object that is susceptible to have an influence on spreaders. It carries a semantic meaning. For instance: a tweet, a meme, a song, a virus, a news article, etc.. 2–11, 14, 18, 20, 28, 31, 40, 45, 61, 62, 64–68, 70–72, 80, 141, 149, 154

rich-get-richer In clustering, a property that gives an entity a higher probability of belonging to the most populated clusters and a lesser probability to belong to the less populated ones.. xi, 11, 79, 83, 87–91, 93–95, 97–99, 101–103, 106, 151, 192

spread An endogenous decision that may infect other spreaders.. 2–8, 18, 20, 32, 39, 42, 55–58, 61–64, 80, 111, 115, 132, 133, 135, 137, 149, 155–158, 181

type Entities of the same type have an identical way of expressing their semantic meaning. For instance, two tweets are of the same type ("Tweet"), two news articles are of the same type ("News"), and a tweet and a news article *can* be of the same type ("Textual documents") or not depending on modeling choices.. 18–23, 25–28, 30, 32, 58

virality Probability of a decision on a piece of information in the absence of interactions.. 18, 40–43, 62, 68, 70, 74, 141

Contents

□