Multivariate Powered Dirichlet-Hawkes Process

Anonymous

Anonymous

Abstract. The publication time of a document carries a relevant information about its semantic content. The Dirichlet-Hawkes process has been proposed to jointly model textual information and publication dynamics. This approach has been used with success in several recent works, and extended to tackle specific challenging problems –typically for short texts or entangled publication dynamics. However, the prior in its current form does not allow for complex publication dynamics. In particular, inferred topics are independent from each other –a publication about finance is assumed to have no influence on publications about politics, for instance.
In this work, we develop the Multivariate Powered Dirichlet-Hawkes Process (MPDHP), that alleviates this assumption. Publications about various topics can now influence each other. We detail and overcome the technical challenges that arise from considering interacting topics. We conduct a systematic evaluation of MPDHP on a range of synthetic datasets to define its application domain and limitations. Finally, we develop a use case of the MPDHP on Reddit data. At the end of this article, the interested reader will know how and when to use MPDHP, and when not to.

Keywords: Dirichlet Process · Multivariate Hawkes Process · Clustering · Information spread · Sequential data

1 Introduction

Understanding the data publication mechanisms on online platforms is of utmost importance in computer science. The amount of user-generated content that flows on social networks (e.g. Reddit) daily appeals for efficient and scalable approaches; they should provide us detailed insights within these mechanisms. A favoured approach to this problem is to cluster published documents according to their semantic content [2, 3, 13]. In addition, recent years underlined the importance of the publication time to perform this task [4, 6, 11]; we expect publications to trigger ulterior observations within a short time range.

In previous works, the understanding of large data flows boils down to sorting data pieces (documents) into independent topics (clusters). However, it has been underlined on several occasions that online publication mechanisms are more complex than that. Typically, it has been underlined that a correct description should involve clusters that interact with each other [7, 14].We illustrate the implications of this claim in Fig. 1. In most existing works that explicitly model

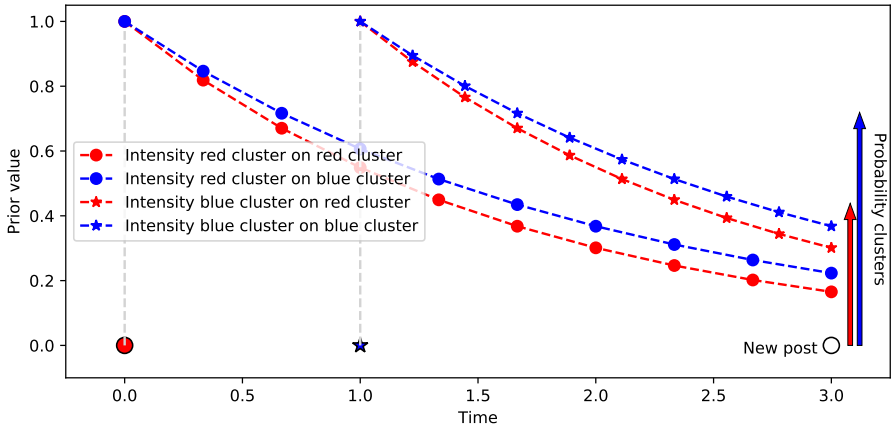

Fig. 1. Illustration of the Multivariate Powered Dirichlet-Hawkes Process prior — A new event appears at time $t = 3$ from a cluster which is yet to be determined. The *a priori* probability that this event belongs to a given cluster c_{red} depends on the sum of the red intensity functions at time $t = 3$. Similarly, the *a priori* probability that this event belongs to a cluster c_{blue} depends on the sum of the blue lines at time $t = 3$. In previous models, this prior probability depends on each cluster self-stimulation only.

both text and time, a given topic is assumed to only trigger observations from the same topic [4, 10] –the red cluster can only trigger observations from the red cluster. Instead, we must allow clusters to trigger publications in any other cluster.

Therefore, we extend a previous class of models (DHP [4] and PDHP [10]) to account for cluster interaction mechanisms. We show that technical challenges arise when considering topical interaction, and solve them. This results in the Multivariate Powered Dirichlet-Hawkes Process (MPDHP). We conduct systematic experiments to test the limits of MPDHP and define its application domain. In particular, we show that it performs well in cases when textual data is scarce and when the number of coexisting clusters is large. Finally, we investigate a real-world use case on a Reddit dataset.

2 Background

The original Dirichlet-Hawkes process (DHP) [4] merges Dirichlet processes and Hawkes processes. It is used as a prior in Bayesian clustering along with a main model –typically a language model. The prior expresses the assumption that a new event from a given topic appears conditionally on the presence of older events from this same topic. The conditional probability is encoded into the intensity function of a Hawkes process. One such Hawkes process is associated to each topic. The temporal (Hawkes) intensity of a topic c is written $\lambda_c(t|\mathcal{H}_c)$; it

depends on the history of all previous events associated to topic c, written $\mathcal{H}_c$. If no Hawkes intensity manages to explain well enough the presence of a new observation happening further in time, the DHP *a priori* guess is that the new observation belongs to a new cluster. DHP have first been used for automated summary generation [4]. A list of textual documents appear in chronological order and are treated as such; the DHP infers clusters of documents that are based both on their textual content and their publication date, and studies their auto-stimulated publication dynamics. This process knew several developments, that essentially consider alternative Dirichlet-based priors combined with Hawkes processes –Hierarchical DHP [6] and Indian Buffet Hawkes process [11].

However, in [13, 10], the authors underline several limits of the standard Dirichlet-Hawkes processes and of the extensions mentioned earlier. For instance, DHP fails in cases where publications content carry few information: when textual content is short (e.g. tweets [13]) or when vocabularies overlap significantly (e.g., topic-specific datasets). In [10], the problem is alleviated by considering a Powered Dirichlet process [9] instead of a standard Dirichlet process. This process is merged with a univariate Hawkes process to make the Powered Dirichlet-Hawkes process. The authors retrieve better results in challenging clustering situations (large temporal and textual overlaps).

However, none of these works allow clusters to interact with each other, despite clues pointing in that direction [7, 14].Indeed in [4, 6, 10, 11], the considered Hawkes processes are univariate: a cluster can only be used to trigger events within itself. Exploring how clusters interact with each other would significantly extend our comprehension of the publication mechanisms at stake in various datasets –such as social media or scientific articles. Identifying which topics trigger the publication of other seemingly unrelated topics might be interesting in the study of fake news spreading. Understanding the dynamics at stake may help to surgically inhibit the spread of such topics using the right refutation. Another possible use case would be nudging users towards responsible behaviours regarding environment, health, tobacco, etc.

In this paper, we extend the (univariate) Powered Dirichlet-Hawkes Process to its multivariate version. There are several reasons why it has not been done in prior works: first of all, the adaptation to the multivariate case is not trivial and poses several technical challenges. As a **first contribution**, we detail the challenges that arise when developing the Multivariate Powered Dirichlet-Hawkes Process (MPDHP). We propose methods to overcome them while retaining a linear time complexity $\mathcal{O}(N)$. Doing so, we also relax the near-critical Hawkes process hypothesis made in [4, 6, 10]. The second reason why the multivariate extension has not been developed in prior works, is that it greatly raises the number of parameters to estimate. The inference task might become harder, and the results irrelevant. Our **second contribution** consists in a systematic evaluation of the MPDHP on a variety of synthetic situations. Our goal is to identify the limits of MPDHP regarding textual overlap, computation time, the amount of available data, the number of co-existing clusters, etc. We show that MPDHP is perfectly fit for solving a variety of challenging situations. Finally, we

illustrate the new insights on topical interaction obtained by running MPDHP on a real-world Reddit dataset.

3 The Multivariate Powered Dirichlet-Hawkes process

3.1 Powered Dirichlet Process

The Dirichlet process can be expressed using the Chinese Restaurant Process metaphor. Consider a restaurant with an infinity of empty tables. A first client enters the restaurant and sits to any of the empty tables with a probability proportional to α –the *concentration parameter*. Another client then enters the restaurant, and sits either at one of the occupied tables with a probability linearly proportional to the number of clients already sat at the table, or to any of the empty tables with a probability proportional to α. The process is then iterated for an infinite number of clients. The resulting clients distribution over the tables is equivalent to a draw from a Dirichlet distribution. The Powered Dirichlet process is intended as a generalisation of the Dirichlet process. The probability for a new client to sit at one of the occupied tables is now proportional to the number of clients at the power r. Let C_i be the table chosen by the i^{th} client and $\mathcal{H}$ the history of table allocation. Formally, the probability for the i^{th} to choose a table reads:

$$PDP(C_i = c|\alpha, r, \mathcal{H}) = \begin{cases} \frac{N_c^r}{\alpha+\sum_c N_c^r} \text{ if c = 1, ..., K} \\ \frac{\alpha}{\alpha+\sum_c N_c^r} \text{ if c = K+1} \end{cases} \quad (1)$$

where N_c is the number of people that already sat at table c, K is the total number of tables, and r a hyper-parameter. Note that when $r = 1$ we recover the regular Dirichlet process, and when $r = 0$ we recover the Uniform Process [12].

3.2 Multivariate Hawkes process

A Hawkes process is a temporal point process where the appearance of new events is conditional on the realisation of previous events. It is fully characterised by an intensity function, noted $\lambda(t|\mathcal{H})$ that depends on the history of previous events. It is interpreted as the instantaneous probability of a new observation appearing: $\lambda(t)dt = P(e \in [t, t + dt])$ with e an event and dt an infinitesimal time interval. For simplicity of notation, we omit the term $\mathcal{H}$ which is implicit anytime the intensity function λ is mentioned.

As in the DHP [4], we define one Hawkes process for each cluster. However in DHP each of them is associated to a **univariate** Hawkes process, that 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 realisation of the i^{th} event belonging to cluster c. We write the intensity function for cluster c

at all times as:

$$\lambda_c(t) = \sum_{t_i^{c'} < t} \alpha_{c,c'} \cdot \kappa(t - t_i^{c'}) \quad ; \quad \kappa_l(\Delta t) = \frac{e^{-\frac{(\Delta t - \mu_l)^2}{2\sigma_l^2}}}{\sqrt{2\pi\sigma_l^2}} \tag{2}$$

In Eq. 2, $\alpha_{c,c'}$ is a vector of L parameters to infer, and $\kappa(t - t_i^{c'})$ is a vector of L temporal kernel functions depending only on the time difference between two events. In our case, we consider a Gaussian RBF kernel, that allows to model a range of different intensity functions.

The log-likelihood of a multivariate Hawkes process for all observations up to a time T reads:

$$\log \mathcal{L}(\alpha|\mathcal{H}) = \sum_c \int_0^T \lambda_c(t)dt + \sum_{t_i^c} \lambda_c(t_i^c) \tag{3}$$

3.3 Multivariate Powered Dirichlet-Hawkes Process

The Multivariate Powered Dirichlet-Hawkes Process (MPDHP) arises from the merging of the Powered Dirichlet Process (PDP) and the Multivariate Hawkes Process (MHP), described in the previous sections. As in [4, 6, 10], 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 to 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. 1, 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:

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

In Eq. 4, λ_c in defined as in Eq. 2, and the parameter λ_0 is the equivalent of the concentration parameter described in Eq. 1. Taking back the illustration in Fig. 1, 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 to its own intensity function.

3.4 Language model

Similarly to what has been done in [4, 10], the MPDHP must be associated to 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 publications [4, 10]. According to this model, the likelihood of the i^{th} document belonging to cluster c reads:

$$\mathcal{L}(C_i = c|N_{<i,c}, n_i, \theta_0) = \frac{\Gamma(N_c + \theta_0)}{\Gamma(N_c + n_i + \theta_0)} \prod_v \frac{\Gamma(N_{c,v} + n_{i,v} + \theta_{0,v})}{\Gamma(N_{c,v} + \theta_0)} \quad (5)$$

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 $\theta_0 = \sum_v \theta_{0,v}$. Note that for any empty cluster, the likelihood is computed using $N_{c_{empty}} = 0$ for every empty cluster c_{empty}.

4 Implementation

4.1 Base algorithm

SMC algorithm We use a Sequential Monte Carlo (SMC) algorithm for the optimisation. The base algorithm is the same as in [4, 6, 10] – a graphical representation of the SMC algorithm is provided in [10]-Fig.1. 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. 4, illustrated Fig. 1) with the textual likelihood (see Eq. 5). 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 probability vector. If the empty $(K + 1)^{th}$ cluster is chosen, the new observation is added to this cluster, and its dynamics are randomly initialised (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 maximising the new likelihood Eq. 3. 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 treated, 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 parameters The parameters α are inferred using a sampling procedure. A number N_{sample} of precomputed vectors is 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 to a likelihood computed from Eq. 3, noted $P(\mathcal{H}|\alpha)$,

where $\mathcal{H}$ represents the data. The sampling procedure returns the average of each of the N_{sample} precomputed α, weighted by the posterior distribution associated to 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 precompute 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 [4, 10]. 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 present our solutions to overcome them. We manage to preserve a constant time complexity for each observation.

4.2 Optimisation challenges

Updating the triggering kernels In the univariate case [4, 6, 10], 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. 2. The likelihood Eq. 3 can be updated incrementally for each sample matrix. A given cluster c has a likelihood value associated to 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 sampled simply by averaging the samples matrices weighted by their likelihood for a given cluster times the prior probability of these vectors being drawn in the first place.

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 with time, and can grow very 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.

However, some older events can be discarded from the computation. When an event is older than 3σ with respect to the longest range entry of the RBF kernel, it can be safely discarded. Clusters whose last observation has been discarded thus have a near-zero chance to get sampled once again. These clusters' contribution to the likelihood Eq. 3 will not change anymore. 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. 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 multivariate modelling is the prior assumption on sample vectors. In [4, 10], each sample vector is sampled from a Dirichlet distribution. This choice is to infer Hawkes processes that are nearly-unstable: the spectral radius of the temporal kernel function $\lambda_c(t)$ is close to 1. However in our case, such 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 from 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 samples rows/columns corresponding to new clusters can be generated one after the other.

5 Experiments

5.1 Setup

We design a series of experiments to determine the use cases of the Multivariate Powered Dirichlet-Hawkes Process[1]. We list the parameters we consider in our experiments. When a parameter does not explicitly vary, it takes a default value given between parenthesis. 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 to 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. 4.

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 [10]. This value is bounded between 0 (perfectly separated functions) and 1 (identical functions):

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

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.

[1] Data and implementations are available at `https://anonymous.4open.science/r/MPDHP-834B/`

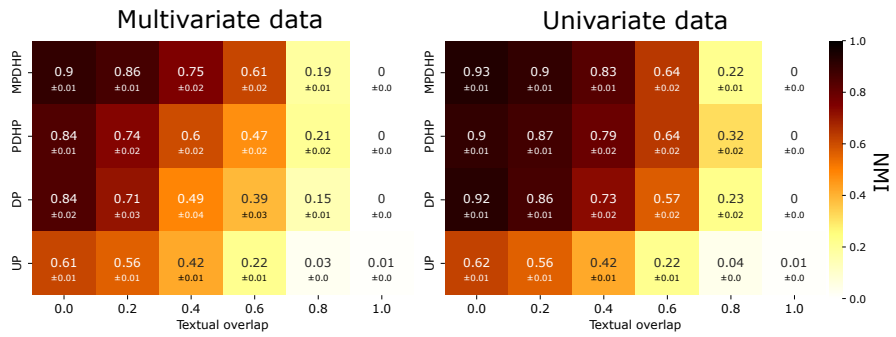

Fig. 2. Numerical results on synthetic data — MPDHP consistently outperforms other baselines designed for the univariate case on both univariate and multivariate data. The standard error has been computed using 100 independent runs.

All datasets are made of 5 000 observations. Observations for each cluster c are generated using a RBF temporal kernel $\kappa(t)$ weighted by a parameter matrix α_c. We set $\kappa(t) = [\mathcal{G}(3; 0.5); \mathcal{G}(7; 0.5); \mathcal{G}(11; 0.5)]$ where $\mathcal{G}(\mu; \sigma)$ is a Gaussian of mean μ and standard deviation σ. We note $L = 3$ the number of entries of κ. The inferred entries of α determine the amplitude (weight) of each kernel entry.

The generation process is as follows. First, we draw a random matrix $\alpha \in \mathbb{R}^{K \times L}$ and normalise it so that its spectral radius equals 1 –near unstable Hawkes process. We repeat this process until we obtain the wanted temporal overlap. Then, we simulate the multivariate Hawkes process using the triggering kernels $\alpha \cdot \kappa(t)$, where $\kappa(t)$ is the RBF kernel as defined earlier. Given the Hawkes process is multivariate, each event is associated to its class it has been generated from among K possible classes. For each 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.

5.2 Baselines

We evaluate our clustering results in terms of Normalised Mutual Information score (NMI). 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). The standard error is computed on 100 runs. We compare our approach to 3 closely related baselines. **Powered Dirichlet-Hawkes process (PDHP)** [10]: in this model, clusters can only self replicate. It means that the intensity function of a cluster c Eq. 2 only considers past events that happened in the same cluster c. r is set to 1. **Dirichlet process (DP)**: this prior is standard in clustering problems. It corresponds to a special case of Eq. 1 where $r = 1$. The prior probability for an observation to belong to a cluster depends on its population. **Uniform process (UP)** [12]: this prior corresponds to a

special case of Eq. 1 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).

5.3 Results on synthetic data

MPDHP outperforms state-of-the-art In Fig. 2, we plot our results for datasets that have 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 our baselines for various values of textual overlap.**MPDHP** systematically outperforms the baselines on multivariate data –when clusters interact with each other. Considering that clusters interacts 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 of MPDHP does not make it unfit to simpler tasks. **PDHP** performs better than MPDHP when the textual overlap is large (textual overlap of 0.8) due to its reduced complexity. Increasing the number of observations would fix this.

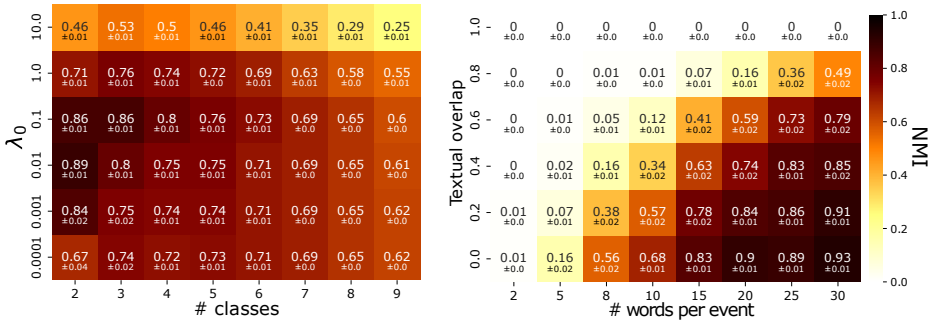

Fig. 3. MPDHP can handle a large number of coexisting clusters and scarce textual information — MPDHP yields good results when a large number of clusters coexist simultaneously (left) and when texts are short or little informative (right). It is also robust against variations of λ_0 over 5 order of magnitude.

Highly interacting processes We test 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. 4), which we vary to see whether MPDHP is robust against it. In Fig. 3 (left), we plot the performances of MPDHP according to these two parameters. We draw two conclusions: MPDHP can handle a large number of coexisting clusters and still correctly identify to which one each document belongs, and MPDHP is robust against large 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. 3 (right), 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 low number of words to yield good results over 5 000 observations. For reference, the overlap between topics can be estimated around 0.25 ([8], 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.

5.4 Real-world application on Reddit

Data We conclude this work with an illustration of MPDHP in a real-world situation. We investigate the interplay between topics on news subreddits, that is how much influence a topic can exert on other ones. The dataset is collected from the Pushshift Reddit repository [1]. We limit our study to headlines from popular English news subreddits in January 2019: inthenews, neutralnews, news, nottheonion, offbeat, open news, qualitynews, truenews, worldnews. We remove headlines that contain less than 3 words as they only add noise to the modelling. After curating the dataset in the way described above, we are left with roughly 8,000 news headlines, which makes a total of 65,743 tokens drawn from a vocabulary of size 7,672.

Parameters We run our experiments using a RBF kernel made of Gaussians centred around $[0, 2, 4, 6, 8]$ hours, with a standard deviation σ of 1 hour, $\lambda_0 = 0.001$, and $r = 1$. We use a Dirichlet-Multinomial language model as in the synthetic experiments with $\theta_0 = 0.01$. As for the SMC algorithm, we set $N_{samples} = 100000$. From our observations, the number of coexisting clusters remains around 10 coexisting clusters (roughly 1,000 parameters), allowing sampling each parameter from approximately 100 candidate values. Each sample parameter is drawn from an identical Beta distribution of concentration parameter $\alpha_0 = 2$. We consider 8 particles for the SMC algorithm, similarly to [4].

Results We present the results of MPDHP on real-world data in Fig. 4. Fig.4a. illustrates a typical output from the model. We can make two interesting observations from this figure. Firstly, the interaction strength between clusters seems to fade as time passes (Fig.4b.). Cluster interactions are more likely to happens within short time ranges. Secondly, the first two clusters seem to be consistently used across the whole month (Fig.4c.). 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. Similarly, the second cluster comprises 46% of r/news articles, which is also roughly

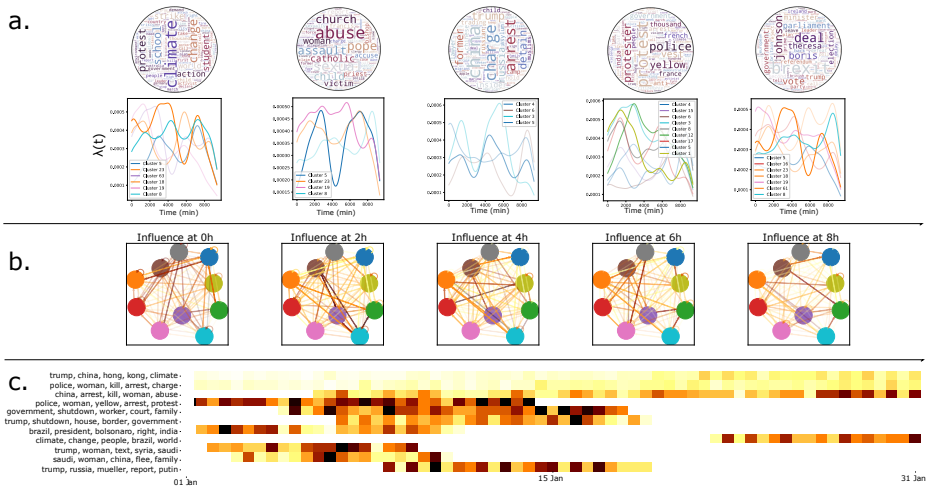

Fig. 4. Real-world application on Reddit – **a.** Examples of clusters along with their inferred reproduction dynamics. **b.** Visualisation of interaction patterns at different times as a network; each dot is a cluster, each edge accounts for the value of $\lambda(t)$ at a given time $t \in [0; 2; 4; 6; 8]$ hours. **c.** Most used clusters represented over real time.

+20% from expected at random. These two clusters therefore significantly account for publications from either of these subreddits, independently from the textual content. Both are general news forums with a large audience; an article that gets posted on other subreddits is likely to also appear on these. Other clusters follow a bursty dynamic, which concurs with [5].

6 Conclusion

In this paper, we extended the Powered Dirichlet-Hawkes process so that it can consider multivariate processes, resulting in the Multivariate Powered Dirichlet-Hawkes process (MPDHP). This new process can infer temporal clusters interaction networks from textual data flow. We overcome several optimisation challenges to preserve a time complexity that scales linearly with the dataset.

We showed that MPDHP outperforms existing baselines when clusters interact with each other, and performs at least as well as the PDHP baseline when clusters do not (which PDHP is designed to model). MPDHP can handle cases where textual content is lesser informative better than other baselines. It is robust against tuning of the temporal concentration parameter λ_0, which allows to handle highly intricate processes. We finally showed that MPDHP performs well with scarce textual data. Our results suggest that MPDHP can be applied in a robust way to a broad range of problems, which we illustrate on a real-world application, that provides insights in topical interactions mechanisms between news published on Reddit.

References

1. Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., Blackburn, J.: The pushshift reddit dataset. Proceedings of the International AAAI Conference on Web and Social Media **14**(1), 830–839 (2020)
2. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd International Conference on Machine Learning. p. 113–120. ICML '06, Association for Computing Machinery, New York, NY, USA (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Du, N., Farajtabar, M., Ahmed, A., Smola, A., Song, L.: Dirichlet-hawkes processes with applications to clustering continuous-time document streams. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015)
5. Haralabopoulos, G., Anagnostopoulos, I.: Lifespan and propagation of information in on-line social networks: A case study based on reddit. JNCA **56** (03 2014)
6. Mavroforakis, C., Valera, I., Gomez-Rodriguez, M.: Modeling the dynamics of learning activity on the web. In: Proceedings of the 26th International Conference on World Wide Web. p. 1421–1430. WWW '17 (2017)
7. Myers, S.A., Leskovec, J.: Clash of the contagions: Cooperation and competition in information diffusion. 2012 IEEE 12th International Conference on Data Mining pp. 539–548 (2012)
8. Posadas Duran, J., Gomez Adorno, H., Sidorov, G., Moreno, J.: Detection of fake news in a new corpus for the spanish language. Journal of Intelligent and Fuzzy Systems **36**, 4869–4876 (05 2019)
9. Poux-Médard, G., Velcin, J., Loudcher, S.: Powered dirichlet process for controlling the importance of "rich-get-richer" prior assumptions in bayesian clustering. ArXiv (2021)
10. Poux-Médard, G., Velcin, J., Loudcher, S.: Powered hawkes-dirichlet process: Challenging textual clustering using a flexible temporal prior. ICDM (2021)
11. Tan, X., Rao, V.A., Neville, J.: The indian buffet hawkes process to model evolving latent influences. In: UAI (2018)
12. Wallach, H., Jensen, S., Dicker, L., Heller, K.: An alternative prior process for nonparametric bayesian clustering. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 892–899. JMLR (2010)
13. Yin, J., Chao, D., Liu, Z., Zhang, W., Yu, X., Wang, J.: Model-based clustering of short text streams. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 2634–2642. KDD '18, Association for Computing Machinery, New York, NY, USA (2018)
14. Zarezade, A., Khodadadi, A., Farajtabar, M., Rabiee, H.R., Zha, H.: Correlated cascades: Compete or cooperate. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence pp. 238–244 (2017)