Decomposing Twitter Graphs Based On Hashtag Trajectories: Mining And Clustering Paths Over MongoDB

Georgios Drakopoulos
Ionian University
Kerkyra, Hellas
c16drak@ionio.gr

Aristeidis Karras
University of Patras
Patras, Achaia, Hellas
akarras@ceid.upatras.gr

Christos Karras
University of Patras
Patras, Achaia, Hellas
c.karras@ceid.upatras.gr

Konstantinos C. Giotopoulos
University of Patras
Patras, Achaia, Hellas
kgiotop@upatras.gr

Phivos Mylonas
Ionian University
Kerkyra, Hellas
fmylonas@ionio.gr

Spyros Sioutas
University of Patras
Patras, Achaia, Hellas
sioutas@ceid.upatras.gr

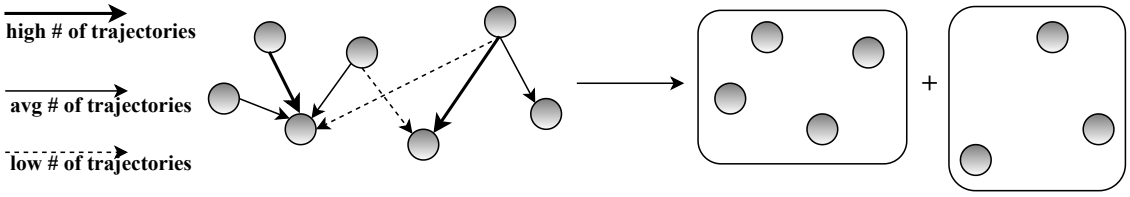

Figure 1: Discovering Twitter communities based on trajectory density.

ABSTRACT

Social media are widely considered as reflecting to a great extent human behavior including thoughts, emotions, as well as reactions to events. Consequently social media analysis relies heavily on examining the interaction between accounts. This work departs from this established viewpoint by treating the online activity as a result of the diffusion in a social graph of memes, namely elementary pieces of information, with hashtags being the most known ones. The groundwork for a general theory of decomposing a social graph based on hashtag trajectories is lain here. This line of reasoning stems from a functional viewpoint of the underlying social graph and is in direct analogy with the biology tenet where living organisms act as gene carriers with the latter controlling up to a part the behavior of the former. To this end hashtag diffusion properties are studied including the retweet probability, higher order distributions, and the mutation dynamics with patterns drawn from a MongoDB collection. These are evaluated on two benchmark Twitter graphs. The results are encouraging and strongly hint at the possibility of formulating a meme-based graph decomposition.

CCS CONCEPTS

• **Information systems**; • **Mathematics of computing** → **Probability and statistics**;

SETN 2022, September 7–9, 2022, Corfu, Greece

ACM ISBN 978-1-4503-9597-7/22/09…$15.00
https://doi.org/10.1145/3549737.3549768

KEYWORDS

hashtag diffusion, hashtag trajectories, graph analytics, graph partitioning, graph reconstruction, higher order statistics, MongoDB

ACM Reference Format:
Georgios Drakopoulos, Aristeidis Karras, Christos Karras, Konstantinos C. Giotopoulos, Phivos Mylonas, and Spyros Sioutas. 2022. Decomposing Twitter Graphs Based On Hashtag Trajectories: Mining And Clustering Paths Over MongoDB. In *12th Hellenic Conference on Artificial Intelligence (SETN 2022), September 7–9, 2022, Corfu, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3549737.3549768

1 INTRODUCTION

Decomposing a social graph to simpler ones based on a certain criterion has long been a major line of research in social network analysis as it allows the easy identification of important entities such as influential accounts, bridges, or community structure. Currently most of these criteria are structural in nature as they mostly refer to connectivity patterns like degrees, paths, or local triangle clustering and have led to algorithms which offer deeper insight into graph structure such as partitioning schemes based on graph Laplacian [19][15]. Still, as in social media nuanced and multifaceted interaction is intentionally encouraged, it makes perfect sense to consider criteria related to graph functionality.

Cultural memes or simply *memes*[1] can be considered, according to [4], to be the rudimentary building blocks of sophisticated constructs and ideas. Depending on the context and the underlying domain, a meme may be a text snippet, a logo, or a short tune such as the *Tetris* theme. Perhaps one of the most well-known memes is *Kilroy* and the associated catchphrase *Kilroy was here* dating back to WWII which found its way even to post-modern literature in

[1]Not to be confused with the now omnipresent Internet memes which are part of the mainstream pop culture and are colloquially also known only as memes.

the classic *The cry of lot 49*. The very definition of meme leads to an analogy with genes [7] and to the viewpoint that social graphs constitute a space for meme evolution [20]. By clustering meme trajectories it is possible to identify latent diffusion channels in the social graph which represents a functional decomposition thereof. Extracting these channels from the graph itself is similar to the principle behind the Hilbert-Huang spectrum. The latter is a decomposition of a signal to elementary ones which, in contrast to the Fourier transform, do not come from a lexicon of known bases but rather depend on the original signal itself [30][39].

The primary research objective of this conference paper is a social graph partitioning algorithm based on its intrinsic diffusion properties whose only hard requirement is that meme trajectory be well defined. Since memes vary across platforms, Twitter was selected without loss of generality as a concrete example. Additionally memes take the form of hashtags which resolves what constitutes an elementary piece of information. Two segments of political Twitter obtained with topic sampling serve as benchmarks and it is shown here that a considerable part of their diffusion capacity is discovered by the proposed algorithm.

The remainder of this conference paper is structured as follows. The recent scientific literature regarding graph partitioning and information diffusion in graphs is briefly reviewed in section 2. In section 3 the basic properties of the hashtags when treated as memes and the proposed algorithm are explained. The results of executing the latter against the benchmark graphs are explained in section 4, whereas in section 5 are given possible extensions of this work. Matrices and vectors are denoted by boldface capital and small letters respectively. Technical acronyms are explained the first time they are encountered in the text. Finally, the notation of this work is summarized in table 1.

Table 1: Notation of this conference paper.

Symbol	Meaning	First in
$\triangleq$	Definition or equality by definition	Eq. (1)
$\{s_1, \ldots, s_n\}$	Set with elements $s_1, \ldots, s_n$	Eq. (6)
$\lvert S \rvert$	Tuple or set cardinality functional	Eq. (6)
$\langle s \rangle$	Length of string s	Eq. (7)
$\mathrm{d}(s, s')$	Levenshtein distance between strings	Eq. (7)
$\lVert \cdot \rVert$	Matrix or vector norm	Eq. (20)
$\deg(u)$	Degree of vertex u	Eq. (24)
$\mathrm{prob}\{\Omega\}$	Probability of event Ω occurring	Eq. (2)
$\mathrm{E}[X]$	Mean value of r.v. X	Eq. (18)
$\mathrm{Var}[X]$	Variance of r.v. X	Eq. (27)
$\langle p \,\lvert\lvert\, q \rangle$	Kullback-Leibler divergence	Eq. (21)

2 PREVIOUS WORK

Graph partitioning or graph segmentation based on structural properties has been historically the first form of graph decomposition [38] which is generic enough to be applied across a broad spectrum of applications including traffic distribution, transportation flows, data scheduling, and load balancing problems to name only a few [16]. Current methodologies rely on graph Laplacian which codifies higher order connectivity patterns [33] as obtained from the graph adjacency matrix eigenexpansion [19]. Graph signal processing (GSP) is an emerging field which treats graphs as two-dimensional signals coming from irregular domains [25] with operations such as sampling, shifting, and reconstruction [23]. In this context graphs partitioning takes place with techniques like variational autoencoders [35], subspace learning [36], and statistical processing [29]. for weighted graphs [15]. Error bounds for graph reconstruction are given in [24]. Graphs compressed with two-dimensional discrete cosine transform are adaptively reconstructed with tensor stack networks [11]. Clustering based on tensor distance metrics has been proposed in [10]. Finally space efficient data structure for evolving graphs for GSP applications is described in [21].

Information diffusion has become a central point in graph mining [6] and existing studies concerning meme diffusion mainly focus on constructing theoretical models from different views [26]. Regularized versions have been proposed [32]. Also diffusion has close ties to graph machine learning [18]. A meme is a rudimentary idea, behavioral pattern, subject, or even style which can be the building block of more elaborate ideas [4] with social media being an excellent vehicle for their rapid spreading [31], possibly in the form of (tiny) links, pictures, videos, or hashtags [37]. Meme diffusion is typically done by identifying patterns associated with a specific meme [14]. Meme diffusion models can be categorized into three groups, namely cascade, epidemic, and competitive models respectively [17]. Hashtags have been used among others in Twitter political campaigns [3], perhaps most notably during the Arab spring of 2011 [1], in Twitter community structure discovery [8], and in spammer detection [2].

3 FUNDAMENTAL NOTIONS

3.1 Hashtags: A Dual Graph View

Memes can be defined as the elementary piece of information which is simultaneously indivisible and meaningful [4]. In spite of the conciseness of this statement and in sharp contrast to genes, in practice what really is a meme depends heavily on the nature of the underlying field and its respective semantics [34].

In the context of this work this field is a Twitter graph, while memes are the hashtags moving along the latter with each tweet or retweet. Hashtags are integral for communication in social media in general and in Twitter in particular as they carry added semantic potential compared to ordinary terms [5]. Any vertex can belong to different trajectories, which is normal Twitter activity.

Definition 3.1 (Hashtag trajectory). The sequence of edges a given hashtag crosses before its first mutation is its respective path.

Channels are the final product of the proposed methodology and they represent inherent communication dynamics within a Twitter graph. Their strength is proportional to the number of hashtags crossing them as well as to their length.

Definition 3.2 (Hashtag channel). A cluster of paths regardless of the clustering method utilized is termed a hashtag channel.

Hashtag sets are by definition important for evaluating path similarity and by extension graph reconstruction error in terms of functionality as proposed here. Please note that for linguistic variety the terms vertex and account will be interchangeably used.

Definition 3.3 (Hashtag set). The set of hashtags contained in the tweets and retweets of a given account constitute its hashtag set.

When seen as individual and autonomous entities, their major properties in the proposed methodology include the following:

- **Diffusion:** The diffusion of hashtags over time with each tweet or retweet is central to the proposed technique. The more hashtags cross a channel, the stronger it is as there is higher topical coherence across its vertices.
- **Asynchronicity:** For each hashtag time runs differently as each one is generated independently and propagates with its own rate depending primarily on its semantics and its association with external events.
- **Mutation:** Changes are usually not random but instead they tend to correspond to a new event or to a development to an existing one. In light of this, mutations are paramount in discovering new evolutionary directions [27].
- **Coherency:** Certain hashtag combinations can be frequently carried over a channel, implying its accounts have similar hashtag sets. The converse need not be true, as accounts with similar set may belong to different channels.

In contrast to the majority of account-oriented techniques, there is no need for anonymity since hashtags as well as other forms of memes for that matter are by definition public and typically not associated with any particular account.

3.2 Mutation

Mutation may appear when a meme moves from an account to another. In other words, it may well be regarded as an imperfect copy operation which can have long running effects depending on the information content of the meme in question or the corresponding content of other coexisting memes. Irrespective of the mutation consequences, the effectiveness of hashtag propagation through Twitter can be evaluated by the mutation probability p_0 of (1) defined as the ratio of the number of distinct mutations n_m to the total number of posts n_s:

$$p_0 \triangleq \frac{n_m}{n_s} \tag{1}$$

Equation (1) expresses the average probability of a mutation occurring during a retweet. Therefore, following a path of L_0 consecutive and independent retweets the mutation distribution is binomial as shown in equation (2):

$$\text{prob}\{c \text{ mutations}\} \triangleq \binom{L_0}{c} p_0^c (1-p_0)^{L_0-c} \tag{2}$$

The rationale behind the binomial distribution is that it counts every possible path of length L_0 where exactly c mutations occur and adds their individual probabilities because of their stochastic independence. Besides being aligned with intuition, the very form of the binomial distribution renders easy the construction of estimators as it depends only on two parameters and moreover is symmetric.

Alternatively, for small values of the average mutation probability p_0 and large path lengths L_0 in the social graph the binomial distribution of (2) can be approximated by the Poisson distribution of (3). The intuition behind this is that the Poisson distribution is frequently a good model for rare events.

$$\text{prob}\{c \text{ mutations}\} \triangleq \frac{\lambda_0^c}{c!} e^{-\lambda_0} \tag{3}$$

To see why this holds true, consider the following pair of approximations of equation (4). These can be proven to hold under mild conditions for a broad range of p_0 and L_0:

$$\begin{aligned} \binom{L_0}{c} p_0^c &\approx \frac{L_0^c}{c!} p_0^c = \frac{(p_0 L_0)^c}{c!} \\ (1-p_0)^{L_0-c} &\approx (1-p_0)^{L_0} \approx e^{-L_0 p_0} \end{aligned} \tag{4}$$

From the pair of equations of (4) it follows that the single parameter λ_0 of the Poisson distribution represents both the parameters of the binomial distribution p_0 and L_0, resulting thus in an information loss. Equivalently this implies that a single Poisson distribution can approximate a class of binomial ones.

Besides the mutation model another important parameter is the distribution of the discrete time steps to the first mutation for each hashtag. The time steps until the first mutation is an important indicator as it may suggest a branch point, usually as a response to an event. Since the average mutation probability p_0 is constant and global, the geometric distribution of (5) models exactly this situation as a sequence of unsuccessful trials until a successful one.

$$\text{prob}\{L \text{ steps to first mutation}\} = p_0 (1-p_0)^{L-1} \tag{5}$$

Even though in most of not all social graphs have a plethora of hashtags, not all of them propagate in the same way. To this end, only the most representative of them constitute H_0, namely the reference hashtag set. Moreover, for each hashtag $h \in H_0$ the set $V[h]$ consisting of itself and its variants as in (6).

$$V[h] \triangleq \{\langle h \rangle\} \cup \left\{ \langle h' \rangle \mid h' \text{ is a mutation of } h \right\} \tag{6}$$

Concerning the selection of the starting vertex for each hashtag, there are a number of strategies. Thus S_0 can be populated in a number of ways. Alternatives include high degree or high centrality vertices, verified accounts, or timestamps. Still, for paths long enough the starting vertex may not be very influential as they may contain the important patterns in later segments.

Since hashtags are essentially text despite their special functional role, a mutation h' of h can be defined based on the Levenshtein distance $\mathrm{d}(h, h')$ between them as defined in (7). Therein $t_{h \to h'}$ is any sequence of elementary transformations, typically insertion, deletion, and replacement, starting from h and resulting in h'. In general, each such transformation may be mapped to different time and memory costs, but this applies to sophisticated operations.

$$\mathrm{d}(h, h') \triangleq \frac{\min |t_{h \to h'}|}{\max \{\langle h \rangle, \langle h' \rangle\}} \tag{7}$$

Addition, subtraction, and substitution of a single character are among the elementary string operations which can be easily discovered by most implementations of the Levenshtein distance. In the context of this work, h' is a mutation of h if and only if ℓ_0 or more elementary operations are required.

3.3 Framework

The above descriptions are summarized in algorithm 1 which essentially is a depth first search (DFS) for hashtags instead of vertices as in the original version. As a consequence, a vertex may well appear

multiple times in the channel structure. The clustering technique, as well as the sets H_0 and S_0 are not part of the algorithm per se, allowing its full parameterization.

Algorithm 1 Graph decomposition based on hashtag trajectories.

Require: Social graph G, sets H_0 and S_0 and clustering scheme
Ensure: Find the graph structure based on the hashtag channels

```
for all hashtags h ∈ H_0 do
    for all tweets or retweets containing h do
        if mutation discovered then
            mark the start of a new path and backtrack
        else
            extend the path by one vertex
        end if
    end for
end for
return hashtag path clustering to channels
```

3.4 ACID or BASE?

Since the patterns mined and used to designate hashtag channels will be stored in a MongoDB instance, a few words will be dedicated here to the SQL vs NoSQL question. Mining Twitter graphs for hashtags and monitoring their trajectories and mutations requires a dedicated database. Although the heading might seem as a chemistry question (pun intended), it is actually a choice over database operating requirements. Recently, partly due to the advent of the Internet of Things (IoT) and social graphs, new database paradigms have been developed resulting in a family collectively known as NoSQL. Its key points are summarized in properties 2 and 1, while table 2 lists the four primary NoSQL technologies [28].

Table 2: NoSQL data types.

Database	Data type
Graph	Linked data and conceptual graphs
Key-value	Associative or key-value array
Document	JSON or BSON documents
Column family	Wide and recursively nested tables

Property 1 (BASE). *The characteristics of NoSQL databases are:*

- ***B**asic **A**vailability. The database is operational most of the time. The percentage of downtime depends on the local operational requirements.*
- ***S**oft state. The database does not have to be written consistent. Also replicas do not have to be mutually consistent sometimes.*
- ***E**ventual consistency. Replicas may be inconsistent temporarily.*

Property 2 (No schema). *NoSQL databases are schemaless.*

The reasons for selecting MongoDB for storing and monitoring meme trajectories in any social graph are the following:

- Twitter exports data directly in Javascript object notation (JSON) format, which is also the native data format of MongoDB.
- MongoDB inherently and fully supports regular expressions which greatly facilitates pattern location in hashtag collections.

In figure 2 the consistency, availability, and partition tolerance (CAP) theorem which states that NoSQL databases can at most have any two of these three properties. MongoDB stands on the CP side along with HBase [22].

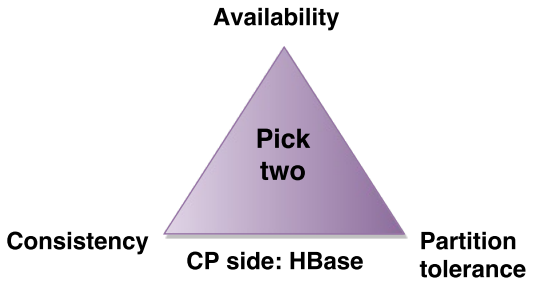

Figure 2: CAP theorem and MongoDB.

4 RESULTS

4.1 Dataset And Experimental Setup

The system architecture which performed the topic sampling to fetch the two benchmark graphs from Twitter is shown in figure 3.

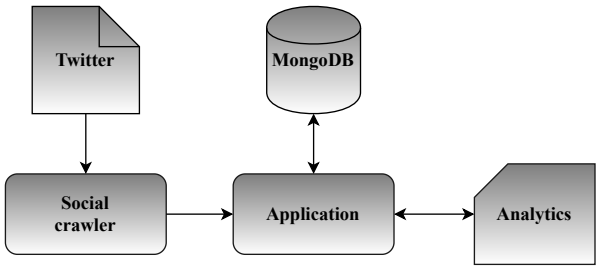

Figure 3: System architecture.

Both graphs have been studied [13][9][12] and their main characteristics are as follows:

- **1821:** The year 2021 marks the 200 years from the Greek Revolution of Independence. This results in a general pleasant climate with long conversations and occasional disagreements over the historical significance of persons or events. Thus, it focuses on a generally acceptable topic of the past.
- **US2020:** The US 2020 Presidential Elections took place in a highly polarized political climate. This is reflected in heated conversations which may be cut short among accusations or frequent live updates about news of local interest. Therefore, its core is a disputed topic unfolding in almost real time.

In table 3 are shown the structural and functional properties of both graphs. There the density ρ_0 of a graph is defined as the ratio of the number of its edges to that of its vertices as shown in equation (8). Along a similar line of reasoning, the logdensity ρ_0' is

Table 3: Dataset synopsis (from [12][13]).

Property	1821 graph	US2020 graph
Number of vertices $\|V\|$	132.317	147.881
Number of edges $\|E\|$	2.225.177	2.447.224
Density ρ_0/ Log-density ρ_0'	16.8170 / 1.2393	16.5486 / 1.2357
Completeness σ_0 / Log-completeness σ_0'	$2.54e^{-4}$ / 0.6196	$2.38e^{-4}$ / 0.6173
Number of triangles	446.513	489.773
Number of squares	215.387	218.633
Number of cliques of size four	102.044	125.806
Graph diameter	10	11
Percentage of vertices reachable at diameter-1	95.33%	98.17%
Percentage of vertices reachable at diameter-2	93.26%	96.44%
Percentage of vertices reachable at diameter-3	89.11%	91.22%
Percentage of vertices reachable at diameter-4	84.73%	87.47%
Number of favorites	36.994.815	42.114.509
Number of tweets	17.465.844	22.773.674
Number of hashtags	21.362.511	27.901.224
Number of distinct hashtags	567.334	793.512

the ratio of the respective logarithms. Observe that the logarithm base does not affect the actual numerical value.

$$\rho_0 \triangleq \frac{|E|}{|V|} \qquad \text{and} \qquad \rho_0' \triangleq \frac{\log|E|}{\log|V|} \tag{8}$$

The completeness σ_0 of a graph is defined as the ratio of its number of edges to the number of edges of a complete directed graph with the same number of vertices as shown in (9). Similarly, the logcompleteness σ_0' is the ratio of the respective logarithms or equivalently of the respective sizes of magnitude.

$$\begin{aligned} \sigma_0 &\triangleq \frac{|E|}{2\binom{|V|}{2}} = \frac{|E|}{|V|\,(|V|-1)} \approx \frac{|E|}{|V|^2} = \frac{\rho_0}{|V|} \\ \sigma_0' &\triangleq \frac{\log|E|}{\log\left(2\binom{|V|}{2}\right)} \approx \frac{\log|E|}{2\log|V|} = \frac{\rho_0'}{2} \end{aligned} \tag{9}$$

The hashtag set H_0 consists of frequently appearing hashtags. One way to determine its cardinality is to fit a model to the set of raw frequencies and then to extract H_0 from it. Given the recall-precision law in information retrieval (IR), a power law of (10) is chosen as model. If the raw frequencies are sorted in descending frequency in a vector $\mathbf{f}$ of length n_h, then the model yields:

$$\mathbf{f}[k] = \alpha_0 k^{-\gamma_0}, \qquad \alpha_0 > 0, \gamma_0 \geq 1 \tag{10}$$

Taking the logarithm of (10) gives the linearized model of (11):

$$\ln \mathbf{f}[k] = \ln \alpha_0 - \gamma_0 \ln k \tag{11}$$

The detailed structure of (11) is the following:

$$\begin{bmatrix} 1 & 0 \\ 1 & \ln 2 \\ \vdots & \vdots \\ 1 & \ln(n_h-1) \end{bmatrix} \begin{bmatrix} \ln\alpha_0 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} \ln \mathbf{f}[0] \\ \ln \mathbf{f}[1] \\ \vdots \\ \ln \mathbf{f}[n_h-1] \end{bmatrix} \Leftrightarrow \mathbf{Mu} = \mathbf{f} \tag{12}$$

Since (12) is overdetermined, one strategy is to compute its least squares (LS) solution through the normal equations. Without regularization the solution of equation (13) is obtained:

$$\mathbf{u}_{LS} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{f} = \mathbf{M}_{LS}^{-1}\mathbf{f}_{LS} \tag{13}$$

The structure of the coefficient matrix $\mathbf{M}_{LS}$ and the vector $\mathbf{f}_{LS}$ of the 2×2 system of (13) are given respectively in (14) and (15):

$$\mathbf{M}_{LS} = \begin{bmatrix} n_h & \sum_{k=1}^{n_h-1}\ln k \\ \sum_{k=1}^{n_h-1}\ln k & \sum_{k=1}^{n_h-1}\ln^2 k \end{bmatrix} \tag{14}$$

$$\mathbf{f}_{LS} = \begin{bmatrix} \sum_{k=0}^{n_h-1}\mathbf{f}[k] \\ \sum_{k=1}^{n_h-1}\mathbf{f}[k]\ln k \end{bmatrix} \tag{15}$$

Given the above, the LS solutions are given in (16).

$$\begin{aligned} a_{0,LS} &= \exp(a_0) \\ a_0 &= \frac{\sum_{k=0}^{n_h-1}\mathbf{f}[k]\left(\sum_{k=1}^{n_h-1}\ln^2 k\right) - \left(\sum_{k=1}^{n_h-1}\mathbf{f}[k]\ln k\right)\sum_{k=1}^{n_h-1}\ln k}{n\sum_{k=1}^{n_h-1}\ln^2 k - \left(\sum_{k=1}^{n_h-1}\ln k\right)^2} \\ \gamma_{0,LS} &= \frac{n_h\sum_{k=0}^{n_h-1}\mathbf{f}[k]\ln k - \left(\sum_{k=1}^{n_h-1}\mathbf{f}[k]\right)\sum_{k=1}^{n_h-1}\ln k}{n\sum_{k=1}^{n_h-1}\ln^2 k - \left(\sum_{k=1}^{n_h-1}\ln k\right)^2} \end{aligned} \tag{16}$$

From the LS parameters of (12) a Pareto bound τ_0 can yield the cardinality of H_0. The selection of S_0 was based on the earliest timestamp for each hashtag.

Table 4 shows the parameters used in the experiments. Table 5 lists the results with the Wiener filter variants in ascending order from left to right and then the DTW. The meaning of each entry is given in the respective subsection.

4.2 Hashtag Trajectory Graph Decomposition

Regarding path clustering, two alternatives were implemented here. The first relies on the dynamic time warping (DTW), a dynamic programming technique for determining sequence similarity which

Table 4: Parameters of the experiments.

Parameter	Value	Parameter	Value
Bound τ_0	0.2	Threshold ℓ_0	2
p_0 (1821)	0.0011	p_0 (US2020)	0.019
Runs N_r	100.000	Wiener filter length q	11, 21, 31

can handle trajectories of varying length. The second is the Wiener filter, which is restricted to fixed length trajectory transforms. The Wiener filter coefficients are computed from (17).

$$\mathbf{R}_{xx}\mathbf{h} \;=\; \mathbf{r}_{xy} \tag{17}$$

The Wiener filter for a wide sense stationary (WSS) and independent and identically distributed (iid) input random process X and output process Y mines self- and cross-similarity patterns between them and encodes them to the coefficients of a finite impulse response (FIR) filter. In turn these are clustered with k-means. The structure of $\mathbf{R}_{xx}$ in shown in (18). Observe it has a rich structure as it is symmetric and Toeplitz, suggesting an efficient solution method, perhaps in iterative form. In fact solutions of the latter type exist and have been widely used in adaptive signal processing.

$$\begin{aligned}\mathbf{R}_{xx} &\triangleq \begin{bmatrix} \mathrm{E}\left[X^2[0]\right] & \dots & \mathrm{E}\left[X[0]\,X[q-1]\right] \\ \mathrm{E}\left[X[1]\,X[0]\right] & \dots & \mathrm{E}\left[X[1]\,X[q-1]\right] \\ \vdots & \ddots & \vdots \\ \mathrm{E}\left[X[q-1]\,X[0]\right] & \dots & \mathrm{E}\left[X^2[q-1]\right] \end{bmatrix} \\ &= \begin{bmatrix} r_0 & r_1 & \dots & r_{q-1} \\ r_1 & r_0 & \dots & r_{q-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{q-1} & r_{q-2} & \dots & r_0 \end{bmatrix}\end{aligned} \tag{18}$$

In this case the Wiener filter is a step one linear predictor of the random process X. Therefore Y is a left shifted by one version of X, hence giving to $\mathbf{r}_{xy}$ the special form of equation (19).

$$\begin{aligned}\mathbf{r}_{xy} &= \begin{bmatrix}\mathrm{E}\left[X[0]\,X[1]\right] & \dots & \mathrm{E}\left[X[0]\,X[q]\right]\end{bmatrix}^T \\ &= \begin{bmatrix} r_1 & \dots & r_q \end{bmatrix}^T\end{aligned} \tag{19}$$

In (18) and (19) r_k is the k-th autocorrelation coefficient of X. Although both alternatives have access to the entire paths, the DTW is more flexible. Nonetheless, in both cases the different nature of the two benchmark graphs is evident, indicating the ability of algorithm 1 to discover communication dynamics.

Once the n_c channels are estimated, the channel matrix $\mathbf{G}_c \in \{0,1\}^{|V|\times n_c}$ is constructed by placing each channel in a column. Observe that $\mathbf{G}_c$ is essentially a compressed version of the original adjacency matrix $\mathbf{G}$ based on a communication, namely a functional, criterion. The real weights necessary to reconstruct $\mathbf{G}$ from $\mathbf{G}_c$ can be obtained by the LS problem of equation (20).

$$\mathbf{W}_c \triangleq \min_{\mathbf{W}} \left\{\|\mathbf{G} - \mathbf{G}_c\mathbf{W}\|_2\right\}, \quad \mathbf{W} \in \mathbb{R}^{n_c\times|V|} \tag{20}$$

The number of channels n_c' for which the root mean square (rms) value of the elements of $\mathbf{W}$ is minimum is taken as the best value of n_c. The rationale behind this is that a lower rms value corresponds to less weight fluctuation, indicating a smoother distribution of channel weights. From the entries of table 5 it can be seen that the DTW variant outperforms the one based on the Wiener filter.

4.3 Functionality Metrics

In this subsection certain graph functionality metrics are used to indicate that for both benchmark graphs the corresponding $\mathbf{G}_c$ maintains a considerable part of $\mathbf{G}$ with any remaining activity being attributed to graph reconstruction error, remote communities, and latent hashtag channels. These metrics are the hashtag mutation distribution, the distribution of steps to first mutation, and the distribution of vertex hashtag sets of orders one to four. For any two discrete distributions p and p^* The Kullback-Leibler divergence computes the entropy of p relative to the reference distribution of p^*. In this work the former is the distribution regarding a feature of $\mathbf{G}_c$ and the latter the corresponding of $\mathbf{G}$.

$$\left\langle p \,||\, p^* \right\rangle \triangleq \sum_k p_k \log\left(\frac{p_k}{p_k^*}\right) \tag{21}$$

To compute the divergence of the first three of the above distributions, the maximum likelihood (MLI) estimators of the models (2), (3), and (5) were used. For the set distributions the empirical ones were used as no model was available. Given n_y i.i.d. observations $\mathbf{y}$ which depend on n_a unknown deterministic parameters stored in $\mathbf{a}$ the likelihood function $l(\cdot)$ is defined as in (22):

$$l(\mathbf{a};\mathbf{y}) \triangleq \ln f_Y(\mathbf{y};\mathbf{a}) \;=\; \ln\prod_{k=1}^{n_y} f_Y(\mathbf{y}[k];\mathbf{a}) \;=\; \sum_{k=1}^{n_y} \ln f_Y(\mathbf{y}[k];\mathbf{a}) \tag{22}$$

The values $\mathbf{a}_{MLI}$ maximizing (22) are the MLI estimators for the parameters of f_Y. Since the above models belong to the broad family of exponential distributions, it suffices to compute the values zeroing the likelihood Jacobian of equation (23).

$$\nabla_{\mathbf{a}} l(\mathbf{a};\mathbf{y}) \triangleq \begin{bmatrix} \dfrac{\partial l(\mathbf{a};\mathbf{y})}{\partial a[0]} & \dfrac{\partial l(\mathbf{a};\mathbf{y})}{\partial a[1]} & \dots & \dfrac{\partial l(\mathbf{a};\mathbf{y})}{\partial a[n_a-1]} \end{bmatrix}^T \tag{23}$$

From the entries of table 5 it follows that the graph resulting from the application of algorithm 1 contains a considerable part of the original functionality for both benchmark graphs. Again the DTW variant achieves better results. Moreover, the Poisson approximation is close to the binomial model as in both graphs p_0 is very low. Additionally, as the order grows the divergence for the vertex profile set distribution decreases, which can be attributed to the fact that some hashtag combinations are far more common and by capturing them in $\mathbf{G}_c$ the dynamics of $\mathbf{G}$ are also preserved. Note that J_b, J_p, and J_g in table 5 denote respectively the distribution for models (2), (3), and (5), whereas J_k the k-th order vertex hashtag profile distribution.

4.4 From Functionality To Structure

The preceding analysis indicated how in both scenarios a major component of the functionality of the $\mathbf{G}$ is preserved in $\mathbf{G}_c$. This still leaves the question of what can be told about the structure of $\mathbf{G}$. Intuitively speaking, since structure contributes to communication, then indirectly a part of it should also be preserved, provided that the compression ratio is high enough and that salient communication patterns are linked to structural ones. For instance, a verified account routinely retweeting hashtags is more likely to be retained.

Table 5: Results of the experiments.

	Sec.	1821	1821	1821	1821	US2020	US2020	US2020	US2020
n_c^*	4.2	34721	28313	21940	19821	41061	38303	34814	28345
J_b	4.3	13.2256	12.1742	11.9904	9.7834	15.5631	13.8912	11.0763	10.3325
J_p	4.3	14.3117	12.9921	12.0048	10.0082	16.6777	15.5660	15.0779	11.7883
J_g	4.3	8.2552	7.9967	7.1142	6.6336	10.2563	9.1147	8.7223	8.2026
J_1	4.3	16.9981	15.0025	13.0862	11.5514	17.9616	15.5445	14.4000	11.9251
J_2	4.3	15.7462	14.3871	12.7913	10.0389	16.3006	15.2114	13.0640	11.4819
J_3	4.3	13.9816	11.3334	9.5626	7.9245	14.6012	13.3810	12.3104	10.5718
J_4	4.3	11.6634	10.7614	8.4777	7.1359	13.7400	12.3334	10.1608	9.1361
I_r	4.4	0.5912	0.6521	0.6818	0.7433	0.5309	0.5776	0.6092	0.6765
V_r	4.4	0.3334	0.2845	0.2214	0.1567	0.3501	0.3231	0.2671	0.2189

In order to examine whether this holds true, in both cases the resulting $\mathbf{G}_c$ served as a starting point for a low complexity and stochastic scale free graph generation model. If systematically a graph $\mathbf{G}_r$ obtained from this model with the same number of vertices is similar to $\mathbf{G}$, then, since the model itself cannot significantly alter the patterns inherent in $\mathbf{G}_c$, the latter must be close to $\mathbf{G}$.

The scale free graph generation model utilized in this work relies on a preferential attachment mechanism where at each time step either of the following two actions is performed:

- With probability equal to the current inverse density of $\mathbf{G}_r$ select two vertices with probability proportional to their inbound degree and connect them.
- Otherwise, insert a new vertex and connect it with an existing one with probability proportional to the inbound degree of the latter.

The preferential attachment mechanism computes a raw score $K(v)$ for every vertex v which is subsequently normalized. For each inbound neighbor u and their respective u' of v the (24). This is repeated for the outbound neighbors and the harmonic mean, which is immune to zero score values, of the two is computed.

$$K(v) \triangleq \frac{\deg(u)}{\sum_{u' \to u} \deg(u')} \tag{24}$$

The similarity metric between $\mathbf{G}$ and $\mathbf{G}_r$ is the correlation of (25), namely the ratio of the number of common edges to the total number of edges of $\mathbf{G}$.

$$E_r \triangleq \frac{1}{|E|} \left| e \mid e \in G \wedge e \in G_r \right| \tag{25}$$

Since $\mathbf{G}_r$ is stochastically generated, it makes sense to compute the average of (25). Assuming (24) is an ergodic process, then its stochastic average equals its realization one. Otherwise, the former can still reasonably approximate the latter over a large number of N_r realizations each with its own coefficient $E_r[k]$.

$$I_r \triangleq \mathrm{E}\left[E_r\right] \approx \frac{1}{N_r} \sum_{k=1}^{N_r} E_r[k], \quad 0 \leq I_r \leq 1 \tag{26}$$

The variance of E_r, or its approximated value, can reveal how reliable I_r is or in other words how close it is to the true value of the correlation coefficient.

$$V_r \triangleq \mathrm{Var}\left[E_r\right] \approx \frac{1}{N_r - 1} \sum_{k=1}^{N_r} (E_r[k] - I_v)^2 \tag{27}$$

From table 5 it follows that the reconstruction error is systematically low as indicated by the relatively high correlation value and the low variance.

5 CONCLUSIONS AND FUTURE WORK

The focus of this conference paper is a Twitter graph decomposition algorithm based on diffusion channels. The latter are estimated by monitoring the possibly non-linear paths of hashtags as they are posted and retweeted and then clustering these paths. The more hashtags or mutations thereof a given channel has, the stronger its effect should be. The two inspirations behind the proposed methodology are the meme theory, stating that complex ideas can be broken down to elementary ones termed *memes* in approximately the same way genome consists of genes, and the Hilbert-Huang spectrum, which decomposes a given signal based on simpler ones reflecting intrinsic properties of the original. When applied to two graphs extracted from political Twitter, it gave encouraging results as the channels capture major parts of graph functions, hinting therefore at a general meme decomposition. Concerning the limitations of the proposed methodology, the main one is that hashtag mutation probability is a global property and cannot be known in advance. Consequently, if it is estimated while hashtag paths are created, then branch backtracks may be necessary.

This work can be extended in a number of ways. First and foremost, experiments with more social graphs with different properties are in order. Also cross language memes can be harvested from non-English speaking Twitter accounts. Moreover, the starting point for each path can be selected in many ways depending on the context. Finally, hashtag evolution appears to be more Lamarckian than Darwinian in the sense that mutations come from use, namely from treating them as strings, but this warrants further investigation.

ACKNOWLEDGMENTS

This conference paper is part of Project 451, a long term research initiative with a primary objective of developing novel, scalable, numerically stable, and interpretable tensor analytics

REFERENCES

[1] Thulfiqar Hussein Altahmazi. 2020. Collective pragmatic acting in networked spaces: The case of# activism in Arabic and English Twitter discourse. *Lingua* 239 (2020).

[2] Reema Aswani, Arpan Kumar Kar, and P Vigneswara Ilavarasan. 2018. Detection of spammers in Twitter marketing: A hybrid approach using social media

analytics and bio inspired computing. *Information Systems Frontiers* 20, 3 (2018), 515–530.
[3] Adam Badawy, Emilio Ferrara, and Kristina Lerman. 2018. Analyzing the digital traces of political manipulation: The 2016 Russian interference Twitter campaign. In *ASONAM*. IEEE, 258–265.
[4] Susan J Blackmore. 2000. *The meme machine* (1st ed.). Oxford Paperbacks.
[5] Riccardo Cantini, Fabrizio Marozzo, Giovanni Bruno, and Paolo Trunfio. 2021. Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs. *TKDD* 16, 2 (2021), 1–26.
[6] Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. 2021. Beltrami flow and neural diffusion on graphs. *Advances in Neural Information Processing Systems* 34 (2021).
[7] Richard Dawkins. 2016. *The extended selfish gene*. Oxford University Press.
[8] Drakopoulos Drakopoulos, Konstantinos C. Giotopoulos, Ioanna Giannoukou, and Spyros Sioutas. 2020. Unsupervised Discovery Of Semantically Aware Communities With Tensor Kruskal Decomposition: A Case Study In Twitter. In *SMAP*. IEEE. https://doi.org/10.1109/SMAP49528.2020.9248469
[9] Georgios Drakopoulos, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. 2020. A graph neural network for assessing the affective coherence of Twitter graphs. In *IEEE Big Data*. IEEE, 3618–3627. https://doi.org/10.1109/BigData50022.2020.9378492
[10] Georgios Drakopoulos, Ioanna Giannoukou, Phivos Mylonas, and Spyros Sioutas. 2020. On Tensor Distances for Self Organizing Maps: Clustering Cognitive Tasks. In *DEXA (Lecture Notes in Computer Science, Vol. 12392)*. Springer, 195–210. https://doi.org/10.1007/978-3-030-59051-2_13
[11] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Lazaros Iliadis. 2021. Transform-based graph topology similarity metrics. *NCAA* 33, 23 (2021), 16363–16375. https://doi.org/10.1007/s00521-021-06235-9
[12] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Spyros Sioutas. 2021. A graph neural network for fuzzy Twitter graphs. In *CIKM companion volume*, Gao Cong and Maya Ramanath (Eds.), Vol. 3052. CEUR-WS.org.
[13] Georgios Drakopoulos, Eleanna Kafeza, Phivos Mylonas, and Spyros Sioutas. 2021. Approximate high dimensional graph mining with matrix polar factorization: A Twitter application. In *IEEE Big Data*. IEEE, 4441–4449. https://doi.org/10.1109/BigData52589.2021.9671926
[14] Sarah Elsharkawy, Ghada Hassan, Tarek Nabhan, and Mohamed Roushdy. 2019. Modelling meme adoption pattern on online social networks. *Web Intelligence* 17, 3 (2019), 243–258. https://doi.org/10.3233/web-190416
[15] David K Hammond, Yaniv Gur, and Chris R Johnson. 2013. Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. In *IEEE Global Conference on Signal and Information Processing*. IEEE, 419–422.
[16] Yudong Han, Lei Zhu, Zhiyong Cheng, Jingjing Li, and Xiaobai Liu. 2018. Discrete optimal graph clustering. *IEEE Transactions on cybernetics* 50, 4 (2018), 1697–1710.
[17] Saike He, Xiaolong Zheng, and Daniel Zeng. 2016. A model-free scheme for meme ranking in social media. *Decision Support Systems* 81 (2016), 1–11. https://doi.org/10.1016/j.dss.2015.10.002
[18] Bo Jiang, Doudou Lin, Jin Tang, and Bin Luo. 2019. Data representation and learning with graph diffusion-embedding networks. In *CVPR*. 10414–10423.
[19] Peiguang Jing, Yuting Su, Zhengnan Li, and Liqiang Nie. 2021. Learning robust affinity graph representation for multi-view clustering. *Information Sciences* 544 (2021), 155–167.
[20] Qingchao Kong, Wenji Mao, Guandan Chen, and Daniel Zeng. 2018. Exploring trends and patterns of popularity stage evolution in social media. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, 10 (2018), 3817–3827.
[21] Stavros Kontopoulos and Georgios Drakopoulos. 2014. A space efficient scheme for graph representation. In *ICTAI*. IEEE, 299–303. https://doi.org/10.1109/ICTAI.2014.52
[22] Michael Marountas, Georgios Drakopoulos, Phivos Mylonas, and Spyros Sioutas. 2021. Recommending database architectures for social queries: A Twitter case study. In *AIAI*. Springer. https://doi.org/10.1007/978-3-030-79150-6_56
[23] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. 2019. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine* 36, 3 (2019), 16–43.
[24] Jari Miettinen, Sergiy A Vorobyov, and Esa Ollila. 2018. Graph error effect in graph signal processing. In *ICASSP*. IEEE, 4164–4168.
[25] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE* 106, 5 (2018), 808–828.
[26] Bastien Pasdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat. 2017. Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE Transactions on Signal and Information Processing over Networks* 4, 3 (2017), 481–496.
[27] Hiroki Sato, Itsuki Doi, Yasuhiro Hashimoto, Mizuki Oka, and Takashi Ikegami. 2020. Selection and accelerated divergence in hashtag evolution on a social network service. In *Artificial Life Conference*. MIT Press, 535–540.
[28] William Schultz, Tess Avitabile, and Alyson Cabral. 2019. Tunable consistency in MongoDB. *PVLDB* 12, 12 (2019), 2071–2081.
[29] Santiago Segarra, Sundeep Prabhakar Chepuri, Antonio G Marques, and Geert Leus. 2018. Statistical graph signal processing: Stationarity and spectral estimation. *Cooperative and Graph Signal Processing* (2018), 325–347.
[30] Xiaocai Shan, Shoudong Huo, Lichao Yang, Jun Cao, Jiaru Zou, Liangyu Chen, Ptolemaios Georgios Sarrigiannis, and Yifan Zhao. 2021. A revised Hilbert-Huang transformation to track non-stationary association of electroencephalography signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), 841–851.
[31] Krzysztof Stepaniuk and Katarzyna Jarosz. 2021. Persuasive linguistic tricks in social media marketing communication – The memetic approach. *PLoS one* 16, 7 (2021).
[32] Arthur D Szlam, Mauro Maggioni, and Ronald R Coifman. 2008. Regularization on graphs with function-adapted diffusion processes. *JMLR* 9, 8 (2008).
[33] Hongteng Xu, Dixin Luo, and Lawrence Carin. 2019. Scalable Gromov-Wasserstein learning for graph partitioning and matching. *NIPS* 32 (2019), 3052–3062.
[34] Fan Yang, Yanan Qiao, Shan Wang, Cheng Huang, and Xiao Wang. 2021. Blockchain and multi-agent system for meme discovery and prediction in social network. *KBS* 229 (2021).
[35] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. 2019. Deep clustering by Gaussian mixture variational autoencoders with graph embedding. In *ICCV*. 6440–6449.
[36] Ming Yin, Shengli Xie, Zongze Wu, Yun Zhang, and Junbin Gao. 2018. Subspace clustering via learning an adaptive low-rank graph. *IEEE Transactions on Image Processing* 27, 8 (2018), 3716–3728.
[37] Daniel Yue Zhang, Jose Badilla, Yang Zhang, and Dong Wang. 2018. Towards reliable missing truth discovery in online social media sensing applications. In *ASONAM*. 143–150. https://doi.org/10.1109/ASONAM.2018.8508655
[38] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. 2018. Multiview consensus graph clustering. *IEEE Transactions on Image Processing* 28, 3 (2018), 1261–1270.
[39] Jingyi Zheng, Mingli Liang, Sujata Sinha, Linqiang Ge, Wei Yu, Arne Ekstrom, and Fushing Hsieh. 2021. Time-frequency analysis of scalp EEG with Hilbert-Huang transform and deep learning. *IEEE Journal of biomedical and health informatics* (2021).