2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

DiffuScope: Inferring Post-specific Diffusion Network

Md Rashidul Hasan *, Dheeman Saha*, Farhan Asif Chowdhury*, James Degnan*, Abdullah Mueen*
*University of New Mexico
{mdhasan, dsaha, fasifchowdhury, jamdeg , mueen}@unm.edu

***Abstract*—Post-specific diffusion network elucidates the *who-saw-from-whom* paths of a post on social media. A diffusion network for a specific post can reveal trustworthy and/or incentivized connections among users. Unfortunately, such a network is not observable from available information from social media platforms; hence an inference mechanism is needed.**

In this paper, we propose an algorithm to infer the diffusion network of a post, exploiting temporal, textual, and network modalities. The proposed algorithm identifies the maximum likely diffusion network using a conditional point process. The algorithm can scale up to thousands of shares from a single post and can be implemented as a real-time analytical tool. We analyze inferred diffusion networks and show discernible differences in information diffusion within various user groups (i.e. verified vs. unverified, conservative vs. liberal) and across local communities (political, entrepreneurial, etc.). We discover differences in inferred networks showing disproportionate presence of automated bots, a potential way to measure the true impact of a post.

I. Introduction

A post on social media has become the method of expression for many. A post travels through the social network from the author to his/her friends, followers, and beyond. However, the diffusion path of a post in social media is not readily observable. We consider the problem of real-time tracking of a post on social media by inferring the diffusion network of the post. Such a diffusion network can be useful to identify trustworthy or incentivized connections among users, which, in consequence, is helpful to manage misinformation propagation and understand public sentiment.

In Figure 1, we show an inferred diffusion network of early two hundred retweeters of one of President Donald Trump's tweets on Oct 6, 2020. These retweets were made within two seconds of the original tweet. We observe bot presence in this early diffusion network, especially among the accounts that have successfully influenced other accounts in such short time. We name a few of these accounts that are already suspended by Twitter.

Although social media platforms know *who-saw-from-whom* information for any diffusion, unfortunately, the information is not publicly available. In this paper, we show that post-specific diffusion network can identify diffusion patterns demonstrating differences in information diffusion within and across local communities. We also discover differences in inferred networks of political tweets showing disproportionate presence of automated bots. Therefore, we propose this *novel problem* of inferring post-specific diffusion network from publicly available information on social media.

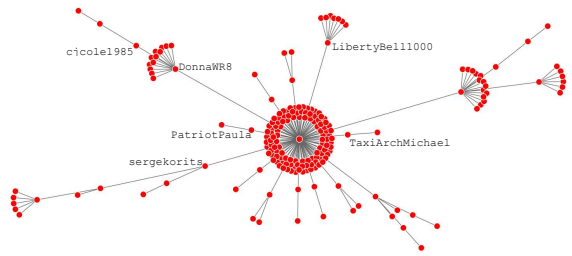

Fig. 1. The diffusion network of one of President Donald Trump's tweets. `@realdonaldtrump` is at the center. Early two hundred retweeters are collected via Twitter API. The diffusion network is inferred by DiffuScope. Seventy accounts receive more than 60% score by Bot-o-meter [4]. The named accounts are already suspended by Twitter.

In this paper, we propose an algorithm to infer the diffusion network of a post from (1) the shares it receives, (2) the social network structure of sharing users, and (3) the history and content of posts in the past. The algorithm uses the temporal point process to model timing of shares, along with the textual similarity of past content and follower distribution of users, to estimate likely diffusion paths on the social network.

We start by motivating the use of temporal, textual, and network information in the inference process in Section II. We set context against related work in Section III. We continue describing the algorithm in Section IV. Section V shows empirical evaluation on synthetic data, and Section VI demonstrates use cases on real data. We make all the figures available in the paper website [1] for interactive and high-resolution viewing.

II. Motivation

Consider four users: A, B, C, and D. Consider a post from A that has been re-posted or shared by B, C, and D. Most social media platforms do not provide *who-saw-from-whom* information. Hence, there can be sixteen different propagation trees for this specific tweet (figure 2). The shares are associated with a timestamp that gives us the time order of re-posts by B, C, and D. In general, on social media, anyone posting earlier cannot be influenced by someone posting later. There can be exceptions, because real-world influence can produce out-of-

ASONAM '21, November 8-11, 2021,Virtual Event, Netherlands

ACM ISBN 978-1-4503-9128-3/21/11
http://dx.doi.org/10.1145/3487351.3490967

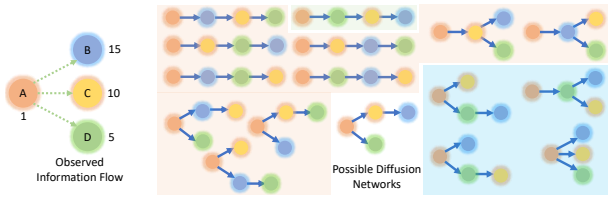

Fig. 2. (left) Observed information from four users. (right) All possible diffusion trees (best viewed in color). Trees in red shade are pruned by temporal information. Trees in blue shade are pruned by textual information. Tree in green shade is pruned by network information, leaving us with the most probable diffusion tree.

order shares on virtual media. However, using the timestamps, one can prune a significant number of possibilities to reduce the search space to a smaller number of options.

If we have textual and network information from these users, in addition to the temporal information, we can exploit them to select the most probable diffusion path from the reduced search space. For example, if B and C are highly similar in their topics of interest, they are more likely to have influenced each other, and hence, we can prune some more examples.

Similarly, if we have network information that B and D do not follow each other, we can exclude another diffusion tree, which leaves us with the most likely diffusion tree. Thus, we infer a post-specific diffusion network by combining multi-modal (i.e. textual, temporal, and network) information. Note that such a diffusion network can include out-of-order influence, and thus, can potentially reveal hidden patterns.

III. Related Work

To the best of our knowledge, inferring post-specific diffusion network is not discussed in the literature. Existing work [6] [10] [14] infer diffusion network of generic concepts across the web, and group together many tweets, blog posts, news articles, etc. to represent the spread of a contagion. In contrast, we consider specific posts on a social network. Post-specific inference is challenging because of the need for real-time application scenarios, while existing work process offline archives of data collected over a duration [10] [12]. Grouping concepts help inference by providing more data per group. Post-specific inference is inherently challenged by sparsity in the data (i.e. the first tweet of an author will not benefit from any historical information) and by dynamic change in influence (based on change in users' interest and connectivity in the network).

Post-specific inference can reveal the roles users play in the diffusion of each post. While contrasting the diffusion network can reveal discernible patterns [16], our work focuses on posts and their authors, as opposed to topics [10] [11] or groups of users [17] or groups of posts [18] [9] .

Information diffusion depends on multiple modalities of user profiles including temporal patterns in engaging with the platform, textual patterns in authorship and readership, and graphical patterns in connecting with other users on the platform. In this work, we combine all of these modalities in inferring the diffusion network. Existing work mostly consider only one mode aggregating the others. For example, [5] models temporal process and does not consider textual content. [6], [17] do not use an explicit social network. [19] [18] [3] exploit textual content and does not consider temporal information. Online analysis of information diffusion on Twitter is proposed in [15]. This online method produces cascades for concepts and suffers from incomplete data. In contrast, we propose to infer diffusion network for as low as tens of shares/retweets.

Our work is inspired by work that model posting behavior as temporal point processes [20] [5]. Work on local influence [18] inference or ego-network inference are user-specific inference algorithms, that inspired us to look at post-specific algorithm.

A. Novelty of DiffuScope

The similarity of DiffuScope to several existing techniques can easily obscure the novelty DiffuScope brings to the literature.

1) DiffuScope infers diffusion of one post. It is a harder problem than inferring aggregate diffusion of concepts, hashtags, topics or web domains. Because the available data for one post is significantly less than the data available for aggregate diffusion. Note that the diffusion networks of two posts from the same user, on the same topic, at different times can be largely different.
2) Most existing work model information diffusion in order to predict future behavior to be able to manipulate, control or exploit information diffusion. DiffuScope is an inference technique to explain how the specific post is being shared; which is a typical data mining task.
3) Diffusion network of recent post(s) create opportunity for timely actions from interested parties.
4) DiffuScope exploits a myriad set of information that includes temporal, textual, posting history and social network of the author and the retweeters. Most existing work consider only a subset of this information. A detailed comparison to existing methods in a tabular format is given in our supporting webpage [1].

IV. DiffuScope

A. Background

On a typical social media, a user follows another user, and thus, they form a directed edge between them on the social network. We define $F_{uv}, 1 \leq u, v \leq n$, as the adjacency matrix representation of the directed network of n users on a social network. We consider unweighted edges, hence, $F_{uv} \in \{0, 1\}$. As we are interested in a single post, without losing generality, we assume that the n users are the ones who shared the post, and we ignore all other users. Although a popular post can receive hundreds of thousands of shares, early diffusion of a post can help forecasting the diffusion of the post in subsequent time. Hence, a diffusion analyzer would focus on the first few thousands of shares, limiting n to a reasonably small number. We require past posts from each of these n users to estimate model parameters for the conditional point process.

We differentiate between a celebrity/popular user and a regular user. We assume that a user can share a post from a celebrity user without explicitly following him/her. Although

one can share anyone's post, it is unlikely to share a post from a non-celebrity user without explicitly following him/her.

The output of our algorithm is a tree rooted at the author of the post. A user on social media may see a post from more than one users, however, we formulate the problem so only one diffusion path (i.e. as opposed to multiple paths) to a user exists. This restriction helps maximizing the likelihood function, and ensures better interpretability of the results. Because the social network dictates information flow on social media, the most likely diffusion network should be a subgraph of the social network. Hence, the problem of inferring diffusion network is reduced to picking *a subgraph of the underlying social network that maximizes the likelihood of the shares being made by the observed users at the observed timestamps.*

B. DiffuScope Model

Suppose a node u_0 posted at time t_0. We call u_0 the *author* node. Let us assume this post is shared by n users in time order, without any co-occurrence. In other words, $t_i < t_j$ whenever $1 \leq i < j \leq n$.

An **external node** represents a celebrity account, which usually follows less number of people than the number of followers it has. Media personnel, politicians, and sportsmen are good examples of external nodes. We assume direct influence from an external node to individual nodes because one can easily see posts from an external node (i.e. `@realdonaldtrump`) by exploring trends, clicking sponsored advertisement, and by searching for the node. In DiffuScope, we assume *any* user can be influenced by an external node.

Internal nodes are users that we are interested in the network. Our goal is to find a diffusion path from the author node u_0 to a sharing node u_n. Note that, u_n *can* be influenced by any combination of nodes in $\{u_1, u_2, ...u_{n-1}\}$ because they all have shared the post before u_n has shared.

The distribution of time between a share and the original post follows a heavy tail which suggests us to model time-to-share with exponential distribution. For any two nodes u and v, the probability of v *saw the post from* u is,

$$p_{uv} \propto exp(-\Delta_{uv})$$

where $\Delta_{uv} = t_v - t_u$, and p_{uv} is the likelihood of a diffusion from u to v. DiffuScope models p_{uv} with three modes of information: temporal, textual and network.

Temporal Information: In our algorithm, we use the Hawkes process [7] [8] with exponential decay to model the time of node v sharing from node u, based on their past sharing history. The Hawkes process is defined as

$$\begin{aligned} \gamma_{uv} &= \int_{-\infty}^{t} \alpha_u e^{-\beta_{uv}(t-u)} dN(u)]dt \\ &= \alpha_u \sum_{\text{all post by } u \text{ shared by } v} exp(-\beta_{uv} \mid t - t_i \mid) \end{aligned} \tag{1}$$

where sharing history is $\forall\, i,\ t_i < t$, α_u is the intensity rate by node u, and β_{uv} is the decay rate of node v given u is the source. Here, γ_{uv} is the rate of the Hawkes process, which we restrict to $\gamma_{uv} \in (\epsilon, 1]$ to directly use as a porbability measure.

Textual Information: We use Jaccard similarity between the bags of words from a pair of users. Let X be the bag of words that node u posted or shared, and Y be the bag of words that node v posted or shared, then Jaccard Similarity is defined as:

$$J_{uv} = \frac{length(X \cap Y)}{length(X \cup Y)} \tag{2}$$

Network Information: We take into account the popularity of a user based on his/her number of followers. We use Rayleigh distribution to model the number of followers. We estimate the influence of a user on a diffusion path using the Rayleigh follower distribution. Let z_u be the number of followers of node u, then

$$\eta_u = \frac{z_u}{\sigma^2} exp(-z_u^2/2\sigma^2), \ \ z_u \geq 0 \tag{3}$$

where σ is scale parameter, and can be estimated by maximum likelihood estimator, $\hat{\sigma}^2 = \frac{1}{2N}\sum_{u=1}^{N} z_u^2$; here N is the total number of nodes in the network, and z_u is the number of followers of the u^{th} node.

We consider the follow-follower network of internal users to formulate the probability of diffusion from u to v, which is p_{uv}. In the absence of any other information, the most likely diffusion path between u and v follows the underlying social network. We define the social network by F_{uv} = 1, if v is a follower of u, 0 otherwise. We incorporate the three kinds of information together to calculate the probability of a potential edge between nodes u and v in the diffusion network.

$$p_{uv} = \begin{cases} \eta_u\, J_{uv}\, exp(-\Delta_{uv})\, \gamma_{uv}, \text{ if } F_{uv} = 1 \\ \eta_u\, J_{uv}\, exp(-\Delta_{uv}) \text{ if } F_{uv} = 0 \text{ and } u \text{ is extenral} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

C. DiffuScope Algorithm

To compute p_{uv}, we need to estimate the parameters γ_{uv}, J_{uv}, and η_u, representing temporal, textual and network information respectively.

Estimating γ_{uv}: We formulate the log-likelihood function, described in Equation 7, in order to optimize the Hawkes process described in Equation 1. The derivation of the log-likelihood function is given in the Appendix of this paper.

$$L = \sum_{i=1}^{n} [\frac{\alpha_u}{\beta_{uv}} (e^{-\beta_{uv}(t_n - t_i)} - 1)] + \sum_{i=1}^{n} log[\alpha_u \sum_{t_i < t_j} e^{-\beta_{uv}(t_i - t_j)}] \tag{5}$$

To find the best values for α_u and β_{uv} that maximize the log-likelihood, we perform a grid search over Uniform distributions with the restriction of $\gamma_{uv} \in (\epsilon, 1]$, where $\epsilon > 0$ is a small numerical value. Algorithm 1 shows the maximization process. The algorithm draws ten U(0,0.1) random numbers as α_u and ten U(0,0.1) random numbers as β_{uv} to form a grid structure in Line 3-4. At each corner of this grid, the likelihood function L is evaluated (Line 5). The algorithm sorts the likelihood values (Line 6) and considers the (α_u,β_{uv}) pairs in descending order of their likelihood (Line 8-9). The algorithm calculates γ_{uv} for each pair of (α_u,β_{uv}) (Line 10) and returns the first $\gamma_{uv} \leq 1$. If the corners of the grid fail

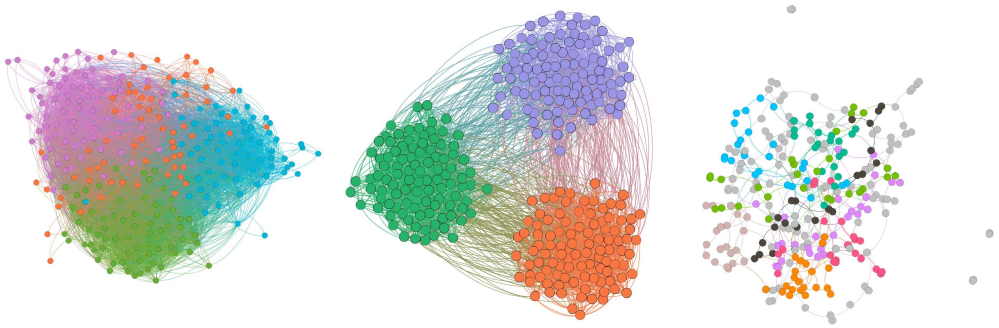

Fig. 3. Three example follow-follower network of different density. Each network consists of roughly 300 nodes. Nodes of same community are colored similar (Modularity based community [13]) (left) A real-world follow-follower network collected from Twitter having density=11.3%, number of communities=4; (mid) A synthetically generated network with density=8%, number of communities=3; (right) Another synthetically generated community with density=1%, number of communities=17.

Algorithm 1 Estimate Gamma

Require: t_i: all shares by node v of a post from node u
Ensure: γ_{uv}
1: **if** v does not share a post from u **then**
2: **return** $\gamma_{uv} \leftarrow \epsilon$
3: **end if**
4: **for** ten α in $U(0, .1)$ **do**
5: **for** ten β in $U(0, .1)$ **do**
6: $L_{\alpha_u \beta_{uv}} = \sum_{i=1}^{n} [\frac{\alpha_u}{\beta_{uv}} (e^{-\beta_{uv}(t_n - t_i)} - 1)] + \sum_{i=1}^{n} log[\alpha_u \sum_{t_i < t_j} e^{-\beta_{uv}(t_i - t_j)}]$
7: **end for**
8: **end for**
9: Sort $L_{\alpha\beta}$ in descending order
10: $\gamma_{uv} \leftarrow 1 + \epsilon$
11: **while** $\gamma_{uv} > 1$ **do**
12: Pick the next likely α_u and β_{uv}
13: $\gamma_{uv} = \alpha_u \sum_{all\ retweet\ i} exp(-\beta_{vu} \mid t - t_i \mid)$
14: **end while**
15: **if** $\gamma_{uv} > 1$ **then**
16: Go to line 3 up to ten times
17: **end if**
18: **return** γ_{uv}

to produce a $\gamma_{uv} \leq 1$, the algorithm generates more grids for up to ten attempts (Line 12). One might worry about non-convergence in ten attempts. In practice, on thousands of posts, Algorithm 1 never tried more than one random grid.

Estimating J_{uv}: We take out stopwords, URLs, emojis from all of the posts, and convert hashtags to words. After cleaning the textual content (including posts, shares, likes, etc.) from both users u and v, we produce the bag of words for each user. We then count the words in both of the bag of words, and calculate the Jaccard similarity by taking the ratio between the number of similar words and the total number of words between two users. If they don't share any similar word, we put a smaller value, ϵ, in order to avoid multiplying by zero probability.

Estimating η_u: We collect the number of followers from the poster and each of the sharing users. We then find the

Algorithm 2 Calculate p_{uv}

Require: u, v : two nodes, t_u and t_v : (re)tweets of u and v, X: bag of words from posts of u,Y: bag of words from posts of v, z: number of followers, N: total number of retweeters, and F_{uv} = 1 if v is a follower of u, 0 otherwise
Ensure: Calculate p_{uv}: Probability of a potential edge from node u to v.
1: $J_{uv} \leftarrow \frac{length(X \cap Y)}{length(X \cup Y)}$
2: $J_{uv} \leftarrow \epsilon$ if u and v do not share any similar words.
3: $\Delta_{uv} = t_v - t_u$
4: $\gamma_{uv} =$ EstimateGamma
5: $\sigma^2 = \frac{1}{2N} \sum_{u=1}^{N} z_u^2$
6: $\eta_u = \frac{z_u}{\sigma^2} exp(-z_u^2/2\sigma^2)$
7: **if** u is followed by v **then**
8: $p_{uv} \leftarrow \eta_u J_{uv} exp(-\Delta_{uv}) F_{uv} \gamma_{uv}$
9: **else if** u is an external node and u is not followed by v **then**
10: $p_{uv} \leftarrow \eta_u J_{uv} exp(-\Delta_{uv})$
11: **return** p_{uv}
12: **else**
13: **return** 0
14: **end if**

Algorithm 3 DiffuScope

Require: Posting history of nodes $1 \leq u \leq n$, Timestamps of all shares of a specific post made by nodes $1 \leq u \leq n$
Ensure: Maximum likely post-specific diffusion tree
1: **for** $1 \leq k \leq n$ **do**
2: **for** $1 \leq i < k$ **do**
3: Calculate p_{ik}
4: **end for**
5: $weight_{ik} \leftarrow$ argmax $p_{ik} \forall\, i < k$
6: **end for**
7: Tree$\leftarrow \max(weight_{ik})$

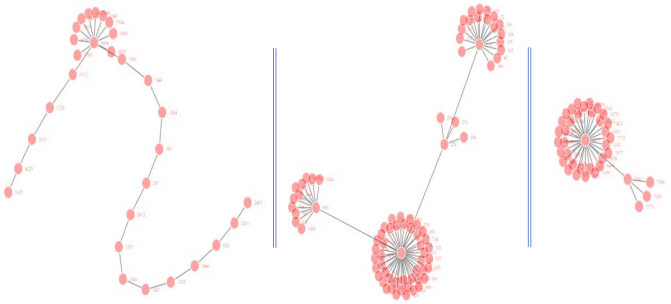

Fig. 4. Examples of synthetic traversals with branching factors B = 0.1, 0.4 and 0.9, respectively from left to right.

probability for each user based on their number of followers using Equation 3.

Once we have η_u, J_{uv}, and γ_{uv}, we combine them into a p_{uv} in Algorithm 2. Recall that p_{uv} is the probability of diffusion from u to v. In Algoirthm 3, we use p_{uv} values to the maximum likely diffusion tree for the given post. Retweet order plays an important role in this process, because a newer post cannot influence an older post. Suppose we have $u_1, u_2, ...u_n$ retweeter from a source node u_0 in order of $t_1, t_2, ...t_n$ respectively. Then at time t_1, u_0 and u_1 will have an edge with p_{01}. But at time t_2, there are two possible edges, either u_0 and u_2 with p_{02} or u_1 and u_2 with p_{12}. In our algorithm, we take the maximum probability between p_{02} and p_{12} and the that edge in the tree, and so on. Therefore, an edge to node k would be from

$$\underset{1 \le i < k}{\operatorname{argmax}}\ p_{ik} \tag{6}$$

After getting all the likely edges, we concatenate them to obtain the most likely diffusion tree.

V. Experimental Evaluation

In this section, we perform experimental evaluation of our proposed method using real and synthetic datasets. We use synthetic datasets to quantitatively evaluate our method's performance and perform sensitivity test to validate our model's robustness in diverse scenarios. We use real datasets to qualitatively evaluate the inferred diffusion networks.

A. Synthetic Datasets

Data on real information diffusion is only collectible, if users log the true influence behind their retweets. Moreover, collecting information from all users on a diffusion path is generally unlikely because of the sampling done at the social platforms. In order to evaluate performance, we develop a synthetic data generator to create synthetic social networks and simulate random post diffusion networks. We generate random sparse graphs containing community structures. The density of edges in the graphs range from 0.5% to 8%.

Example graphs are shown in Figure 3. We simulate random traversals rooted at random nodes of the synthetic network. Depending on an external parameter, the *branching factor* (B), the traversal randomly chooses between *breadth-* and *depth-*first approaches to visit the next set of nodes. The higher the branching factor, the traversal visits immediate followers more often than followers of the followers. We choose six branching factors $\{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$ to simulate diffusion netowrks. We show three example traversals in Figure 4.

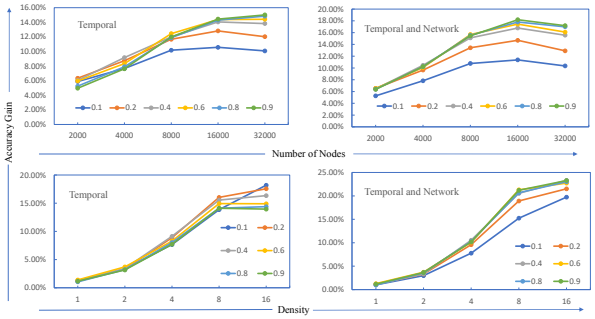

Fig. 5. Left column shows accuracy gain of DiffuScope with only temporal information (Hawkes Process). The right column shows accuracy gain with both temporal and network information (Hawkes and Rayleigh). (top-row) We iteratively double the number of nodes for a fixed density factor of 4. (bottom-row) We iteratively double the density factor for a fixed 4000 nodes in a network. The larger the density factor, the more density the network is. The larger the branching factor, the more nodes have high out-degree in the network.

B. Performance Measure

We propose an edge-based accuracy measure to quantify the correctness of an inferred diffusion network. If an edge in the inferred network is present in the true diffusion network, we count that edge as true-positive (TP). Any spuriously inferred edge will be a false-positive (FP) and any missed edge in the true diffusion network will be a false-negative (FN). We define accuracy as $\frac{TP}{TP+FP}$ for any inferred network, and take the average over all inferred networks for all posts to calculate the overall accuracy on a dataset. The *default accuracy* for a randomly inferred diffusion network depends on the size and shape of the follow-follower network of the participating users. In one extreme, if n users share a post and they are all connected to each other, there are $n-2$ ways to be incorrect for each node, the default accuracy would be $\frac{1}{n-2}\%$. In the other end, if the n users are serially connected in the follow-follower network, the default accuracy is 100% because information can diffuse exactly in one way. We report *accuracy gain* which is the difference between the accuracy of DiffuScope and the default accuracy.

C. Scalability

We generate synthetic datasets by varying the number of nodes in the network. We explore upto $n = 32,000$ nodes by iterative doubling, and generate fixed number (2,000) of posts, each having upto 100 shares. Each post diffuses through a node to either *all* of the followers or *one* of the followers. The branching factor B determines the ratio of the two possibilities. We calculate accuracy of inferred network of each post, and report the accuracy gain. See Figure 5.

We achieve positive gain for all synthetic scenarios. The gain grows with the size and density of the network. The larger the network, the larger and more diverse diffusion trees are. Hence, there are more ways to be wrong for low default accuracy, while DiffuScope holds up the accuracy for

a larger gain. Denser networks have more connections between users, thus, default accuracy suffers on high density networks. In contrast, DiffuScope demonstrates a larger gain for high density network. We see that adding network information with the temporal information increases the gain (from left column to right column) by around 5%.

We see an increase in accuracy gain for increasing branching factor, B. When $B = 0.1$, the diffusion networks have long chains of nodes favoring the default inference, reducing the room for improvement for DiffuScope. However, when $B = 0.9$, the diffusion networks have shorter chains and larger influence spheres, reducing the accuracy of default inference and increasing the gain of DiffuScope.

D. Real Datasets

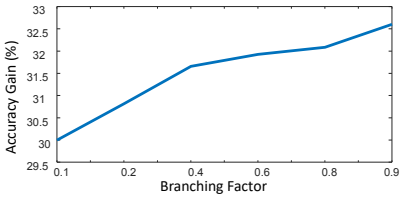

Fig. 6. Accuracy gain with respect to default estimator for various branching factors of seeded diffusions on real social network.

Although our synthetic data contain community structures and are produced with variable density, real world social networks contain rich details including various types of users (e.g. celebrities, bots, experts, trolls, etc.) and activity profile (e.g. consumer, producer, propagator, etc.). In order to evaluate our algorithm on real social network, we collect a small sub-network of Twitter's social network. We focus on the follower-base of a regional news source, *ABQ Journal*. ABQ Journal is a moderately popular regional news media based in Albuquerque, New Mexico, USA, with roughly 88K followers at the time of data collection in February 2020. To form a densely connected user network with a common interest, we filter $8,543$ users who declared Albuquerque (or nearby locations in New Mexico). Afterward, we collect user information (i.e. past tweets and retweets) and follower information for all these users, termed as *ABQ-Journal-Users*. We also collect their most recent posted or retweeted tweets up to the limit of 3200 as set by Twitter Rules. Based on the collected follower information, we form a closed follow-follower network, where each node belongs to the *ABQ-Journal-Users* and all edges between any two *ABQ-Journal-Users* are collected. This small sub-network of Twitter is very dense with an average of 47 edges per node.

Seeded-by-Real-Graph: We simulate diffusion networks on the *ABQ-Journal-Users* network for various branching factors. We test DiffuScope blindly on the simulated diffusions and measure the accuracy gain over the default inference technique. Since, the *ABQ-Journal-Users* network is a dense network, the diffusions for larger branching factors (e.g. 0.9) are generally broader in reach. DiffuScope achieves more than 30% gain over the default inference for such broader diffusion (figure 6).

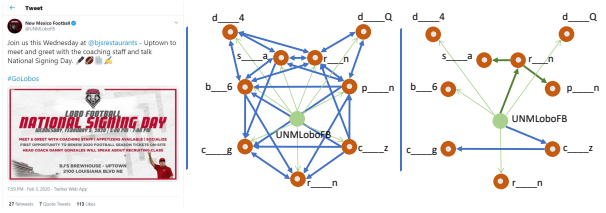

Fig. 7. (left) One example tweet. (middle) The follow-follower network of all users that retweeted the tweet. The author of the tweet is a college football team, UNMLoboFB, shown at the center. (right) The diffusion path inferred for this specific tweet revealing employees of the authoring institute (green path) and past athletes who are currently not affiliated.

VI. Case Studies

A. On ABQ Journal Followers

In Figure 7, we show a tweet on Twitter posted by a verified user, UNMLoboFB, representing a college football team. The tweet is retweeted by nine users whose follow-follower relationships are shown in Figure 7(middle). We infer the most likely diffusion using the retweets and the follow-follower network, shown in Figure 7(right). The diffusion network identifies two unique paths: one through employees of the college (green path), and the other through coaches and players of the team (blue path). The tweet diffused to the rest of the users directly from the author, UNMLoboFB.

Diffusion Within Communities. We use *ABQ-Journal-Users* dataset to investigate how diffusion happens in a densely connected network. Specifically, we answer: is diffusion more likely to occur between users of the same community in comparison to diffusion between random users? First, to identify users of similar interests, we utilize the follow-follower network. We make a reasonable assumption that users with more common connections belong to the same cohort, as used in related studies [2]. In this regard, we use the Modularity based community detection algorithm [13] to identify communities in the follow-follower network. The Modularity based approach minimizes connection across communities and maximizes connection within communities. We recognize a total of 11 communities; however, 99% of users belong to five significant communities.

Based on the most popular topics in the community, we name these communities as (1) Entertainment: users in this community tweet about video, movie, TV, etc.; (2) Education; members of this community are focused on discussing regional school-related topics; (3) Politics: users from this community talk about policing, legislation, budget, etc.; (4) Media: this community is largely comprised of media personalities who share about breaking news; (5) Business: people from this community share business and entrepreneurial topics that promoting local businesses and start-ups. One commonality across all communities is the heavy presence of Covid-19 related issues, which demonstrates the pandemic's pervasiveness and its socio-economic impact on diverse groups of people.

We identify 188 tweets that were retweeted by at least 10 users. Afterward, we use DiffuScope to infer diffusion network for each these retweets that happened within the *ABQ-Journal-Users* network. For each of the diffusion network, we measure how many times a diffusion edge occurs across

distinct communities. We identified 261 diffusion edges, where 61 of them were across communities, resulting in an across community diffusion ratio of 0.24. To contrast this across community diffusion, we identify the number of follow-following connection that occurs across our detected communities. We identify a total of 462, 591 edges in the whole network, where 192, 633 crosses across communities, with a ratio of 0.42. By comparing the two obtained value for across community diffusion and across community connection, we identify that diffusion is more likely to happen within community than across community than the diffusion that could happen with existing follow-follower connections.

B. On Political Tweets

Tweet Diffusion: We develop a software tool to run DiffuScope in real-time. The input to the tool is a user id. The tool repeatedly checks out Twitter API until the user posts a tweet. As soon as the tweet id is available, the tool starts requesting for retweets (i.e. shares) of the specific tweet at a regular interval, until a desired number of retweets are collected or a duration of time is passed. Depending on the speed of diffusion, collected retweets across API requests may have a large amount of repetition. The tool deduplicates the retweets and sorts them in time order before further processing. Attributes of each retweet include retweeter's screen name, id, retweet text, and creation time. The tool is available for download in our supporting webpage [1].

The next step is to collect the history, follower, and following information of each of the retweeters. The Twitter API limits the number of tweets and retweets to 3200 recent objects at each request. On the other hand, there is not any limit on the number of following and followers' information. However, the number of such requests will encounter few waiting periods, each up to 15 minutes. Therefore, to collect history and network data of all retweeters, the tool needs on the order of few hours. Once collected, executing DiffuScope algorithm is much quicker compared to the time needed for data collection. The tool automatically handles suspended users and private users, whose history and network data are unavailable.

Tweets from Competing Campaigns. We use our tweet tracker to collect the earliest two hundred retweeters of a tweet and infer the diffusion network of the tweet. At first, we consider two tweets from two competing political campaigns and search for differences in the inferred diffusion network. In Figure 8, we show one of President Trump's tweets on left, and one of presidential candidate Joseph Biden's tweets in the middle.

The inferred networks from the two tweets show significant visual differences emanating from the differences in the number and size of clusters of users. The first two hundred retweeters are more likely to be influenced by Trump directly than by other sources (i.e. no obvious cluster). In contrast, the first two hundred retweeters of Biden's tweet show two distinct clusters influenced by `@PalmerReport`, a liberal media, and `@davidmweissman`, steering committee member of an anti-Trump organization of conservatives, named The Lincoln Project.

The above findings can largely be attributed to the difference in the number of followers the two candidates have on Twitter (87M vs. 11M). To account for this difference, instead of the first two hundred retweeters, we extract randomly selected two hundred retweets of the Trump's tweet within one minute of the original tweet, which is the time for collecting the first two hundred retweets of Biden's tweet. The inferred network for the Trump's tweets is shown in Figure 8(right). The network shows one distinct cluster centered at `@ThatTrumpGuy`, who promoted a conservative project named `@Project_Veritas`. The account is currently suspended at the time of writing.

C. On (Un)Verified Users

DiffuScope has enabled us to look at tweets of user groups at a greater detail. Consider verified users whose accounts are verified by Twitter, who otherwise do not necessarily have any commonality. We ask if there is any significant difference in how tweets from verified and unverified users propagate. In order to evaluate that we create ten diffusion networks of ten tweets from ten verified users as test set. We collect the same from ten unverified users as the control set.

In Figure 9, we show examples of networks from both test and control set. Visual inspection of the first one hundred retweeters of each tweet shows a dramatic difference in structures of these networks. We evaluate the average diffusion length from the root to the leaves on twenty diffusion networks. The average diffusion length for tweets from verified users is 1.27 with a variance of 0.24 variance. The same from unverified users is 1.79 with a variance of 0.58. We find the distributions are statistically different in a two sample t-test with a p-value of 0.02 at 5% significance level.

VII. Conclusion

This paper develops a technique to infer diffusion network of specific posts on social media. We demonstrate effectiveness of DiffuScope over a variety of datasets, both synthetic and real. We show various diffusion networks in political and news domains along with existence of abusive users on these networks. However, this work is merely one step towards better monitored social media, significant effort must be made to protect human users from inorganic influence.

VIII. Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. CNS-2008910 and a contract with Sandia National Laboratory.

References

[1] Project: DiffuScope. https://sites.google.com/view/diffuscope/home.

[2] F. A. Chowdhury, L. Allen, M. Yousuf, and A. Mueen. On twitter purge: A retrospective analysis of suspended users. In *Companion Proceedings of the Web Conference 2020*, pages 371–378, 2020.

[3] F. A. Chowdhury, D. Saha, M. R. Hasan, K. Saha, and A. Mueen. Examining factors associated with twitter account suspension following the 2020 us presidential election. *arXiv preprint arXiv:2101.09575*, 2021.

[4] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer. BotOrNot: A System to Evaluate Social Bots. In *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, pages 273–274, New York, New York, USA, 2016. ACM Press.

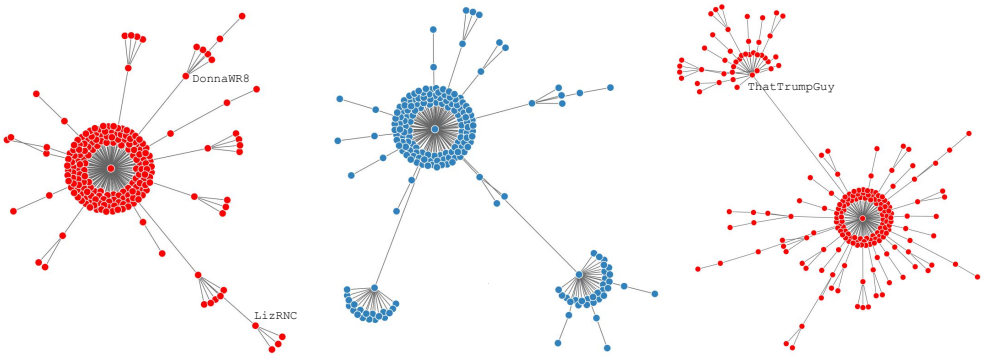

Fig. 8. Three retweet diffusion network for tweet's from (left) Republican Presidential Candidate, (mid) Democratic Presidential Candidate, (right) Republican Presidential Candidate. The earliest 200 retweets were crawled right after both tweets were posted. In both case, the original posters are at the center of the largest stars. Interactive high-resolution and node-labeled graphs are available in paper website.

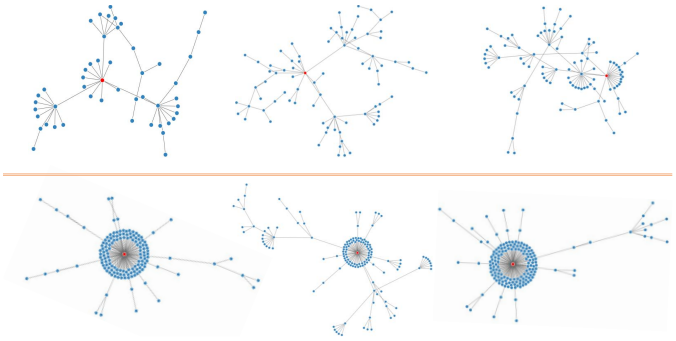

Fig. 9. Examples of diffusion networks of tweets from verified users (bottom) and regular users (top). Source nodes marked in red.

[5] N. Du, L. Song, M. G. Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, pages 3147–3155, 2013.

[6] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.

[7] A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

[8] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[9] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 137–146, New York, NY, USA, 2003. Association for Computing Machinery.

[10] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506, 2009.

[11] S. A. Myers and J. Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *2012 IEEE 12th international conference on data mining*, pages 539–548. IEEE, 2012.

[12] S. A. Myers and J. Leskovec. The bursty dynamics of the twitter information network. In *Proceedings of the 23rd international conference on World wide web*, pages 913–924, 2014.

[13] M. E. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[14] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 695–704, 2011.

[15] I. Taxidou and P. M. Fischer. Online analysis of information diffusion in twitter. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1313–1318, 2014.

[16] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[17] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *2010 IEEE International Conference on Data Mining*, pages 599–608. IEEE, 2010.

[18] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, volume 13, pages 2761–2767, 2013.

[19] J. Zhang, P. S. Yu, and Y. Lv. Organizational chart inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1435–1444, 2015.

[20] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1513–1522, 2015.

APPENDIX

The derivation for log-likelihood of Hawkes process is

$$\log L(t_1, t_2, ...t_n|\alpha_u, \beta uv) = -\int_0^T \gamma_{uv}(t|\alpha_u, \beta uv)dt + \int_0^T log\, \gamma_{uv}(t|\alpha_u, \beta uv)dN(t)$$

where $t_1, t_2, ...t_n$ are the retweeting history by node v from node u. Now, From equation 1, we can write

$$\log L(t_1, t_2, ...t_n|\alpha_u, \beta_{uv}) = -\int_0^T [\int_{-\infty}^t \alpha_u e^{-\beta_{uv}(t-s)} dN(s)]dt + \int_0^T \log[\int_{-\infty}^t \alpha_u e^{-\beta_{uv}t} dN(s)]dt$$

$$= -\int_0^T [\int_s^{t_n} \alpha_u e^{-\beta_{uv}(t-s)} dN(s)]dt + \int_0^T \log[\int_{-\infty}^t \alpha_u e^{-\beta_{uv}t} dN(s)]dt$$

$$= \int_0^T [\frac{\alpha_u}{\beta_{uv}}(e^{-\beta_{uv}(t_n-s)} - 1)]dN(s) + \int_0^T \log[\int_{-\infty}^t \alpha_u e^{-\beta_{uv}t} dN(s)]dt$$

Finally,

$$L = \sum_{i=1}^n [\frac{\alpha_u}{\beta_{uv}}(e^{-\beta_{uv}(t_n - t_i)} - 1)] + \sum_{i=1}^n \log[\alpha_u \sum_{t_i < t_j} e^{-\beta_{uv}(t_i - t_j)}] \quad (7)$$