Evolving Centralities in Temporal Graphs: A Twitter Network Analysis

3 authors:

Fabiola Souza Fernandes Pereira
Universidade Federal de Uberlândia (UFU)

8 PUBLICATIONS   **10** CITATIONS

SEE PROFILE

Sandra de Amo
Universidade Federal de Uberlândia (UFU)

76 PUBLICATIONS   **395** CITATIONS

SEE PROFILE

João Gama
University of Porto

356 PUBLICATIONS   **5,351** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   Weightless Neural Systems View project

Project   Streaming Social Network Analysis and Visualization View project

Evolving Centralities in Temporal Graphs: a Twitter Network Analysis

Fabíola S. F. Pereira, Sandra de Amo
School of Computer Science
Federal University of Uberlândia, Brazil
{*fabiola.pereira, deamo*}*@ufu.br*

João Gama
LIAAD, INESC TEC
University of Porto, Portugal
jgama@fep.up.pt

Abstract—**In online social media systems users are not only posting, consuming, and sharing content, but also creating new and destroying existing connections in the underlying social network. This behavior lead us to investigate how user structural position reacts with the evolution of the underlying social network structure. While centrality metrics have been studied in the past, much less is known about their temporal behaviors and processing, mainly when analyzing not just networks snapshots, but interval graphs. Here, we study Twitter follower/followee network and how users centralities evolve over time. Our analysis is founded on temporal graphs theory. First, we model Twitter as a temporal network and revisit the concept of shortest path considering the time dimension. We show how to compute closeness and betweenness centralities using fastest paths. Then, we propose a baseline algorithm for mining streams of temporal networks. The task is to find all pairs fastest paths inside an observation window. We find that Twitter users are fairly dynamic and from one moment to the next, they can assume (or leave) central roles in the network.**

1. Introduction

Twitter is one the most dynamic networks in the world, where users are always posting and sharing content, besides creating new and destroying existing connections in the underlying network. A key problem in online social networks analysis is that although real social networks are fairly dynamic, traditional approaches are essentially static, i.e., nodes and edges are analyzed as being persistent over time [1].

When considering the additional dimension - time, social networks can be analyzed as *temporal networks*, where the times when edges are active are explicit elements of representation [2]. A classical example of temporal network application is on disease contagion through physical proximity. Usually, the spreading of pathogenic organisms occurs through contact between two individuals. Thus, a temporal network seems to be the best way to represent this scenario. Specially, social networks, our topic of interest, can also be represented as temporal networks, in the sense that they are ubiquitous and complex on their interactions [3].

The structural properties of a complex network usually reveal important information about its dynamics and func-

tion [4]. This is particularly true if we take into account the relationship between the position occupied by a node and the role played by this node for the evolution of a dynamic process. For instance, not all nodes have the same impact on the spreading of a rumor over a network: intuitively, the nodes having a higher number of neighbors should contribute much more to the spreading than nodes having few connections. However, if we perform a deeper analysis, we observe that not just the number of edges is important to identify good spreaders. In fact, nodes mediating a large number of shortest paths are indeed those that contribute the most to the spreading of information over a network. The identification of nodes that play a central role, i.e., nodes having high centrality, has been a quite active research field in complex network theory. Here we review two standard centrality measures – closeness and betweenness – and their extension to the case of time-varying graphs.

Finally, analyzing temporal networks implies on processing high velocity applications, where the evolve nature of data creates complexity for data mining algorithms. A lot of stream network mining algorithms have emerged from this scenario [5], [6], [7], [8]. While, some algorithms process streams of networks considering just static nodes and edges (snapshots networks) [5], others adopt approximation and sampling techniques to track evolving structural properties behaviors [6], [9].

In this work we analyze Twitter as a social temporal network. This is a pioneer quantitative study over Twitter follower/followee network and its evolving aspect. Our focus is on centralities metrics *closeness* and *betweenness*. We address two challenges: (i) modeling Twitter network using temporal graphs structures and (ii) processing these structures as graph streams. The goal, in the end, is to show that it is important to consider the sequence of contacts when analyzing evolving networks. This implies in a different meaning and richer analysis than considering just networks snapshots over time. Furthermore, we aim at quantifying how dynamic a real network, as Twitter, may be from users relationship viewpoint.

First, in Section 2 we propose a formalism to represent Twitter as a temporal graph. Once represented, we formalize path problems over temporal graphs, as well as revisit centrality metrics considering the new aspects of temporal paths presented so far. In the next section, we detail

the Twitter dataset used for analysis. It is an unprecedent dataset, containing the timestamps of when users start/finish follow/unfollow relationships. In Section 4 we propose a baseline algorithm for mining streams of temporal networks. In Section 5 we present the analysis of evolving centrality behaviors. In the end, we discuss about proposed approaches and applications and Section 7 concludes the paper.

2. Social Temporal Networks

In this section, we introduce the main concepts related to temporal networks and discuss how centrality metrics can have different behaviors if we consider the dimension time on the calculation of *paths*. Our main goal here is to represent social networks, specially Twitter, founded on temporal graphs theory [2].

According to [2], temporal networks can be divided into two classes corresponding to the two types of representations: *contact sequences* and *interval graphs*. While in contact sequences, the edges are active over a set of times, in interval graphs, they are active over a set of intervals. In this paper, we are interested in representing Twitter social network as an interval graph.

Formally, let $G = (V, E)$ be a directed temporal graph, where V is the set of nodes and E is the set of edges of G. A directed edge $e \in E$ is a quadruple $(u, v, t_{init}, t_{end})$, where $u, v \in V$, the direction is from u to v, $t_{init} < t_{end}$ and the interval $[t_{init}, t_{end}]$ corresponds to the existence period of e in G.

For example, the temporal information of Twitter has the following meaning: the nodes are users and an edge $(u, v, t_{init}, t_{end})$ indicates that v starts following u at t_{init} and unfollows u at t_{end+1} (v follows u during $[t_{init}, t_{end}]$). As we are dealing with a real dynamic social network, an user can follow/unfollow another user at any time. This is the most interesting aspect that we are investigating: how *following* relationships on Twitter evolve over time and impact on users behaviors?

There are three global variables that must be defined for we start reasoning with temporal graphs: W, R and T. $W = [n, N]$ is the window time of observation of G, for $n < N$. R is the retention time of nodes, i.e., the time between information arrival in the node and the instant from which it can be forwarded. T is the edge traversal time. In our Twitter temporal graph representation, we adopted $R = 1$ day and $T = 0$, as tweets are published instantaneously and the average interaction time for posts is one day [7]. Notice that in our context, T and R are defined as global variables, but for flight graphs, for example, each edge (flight traversal) and node (airports) must be defined with their respective traversal and retention times [7]. Figure 1 is an example of Twitter as a temporal network – specifically, an interval graph.

2.1. Temporal Paths

The definitions here are grounded on the works [4], [7], [10] with soft adaptations. The topological structure of static
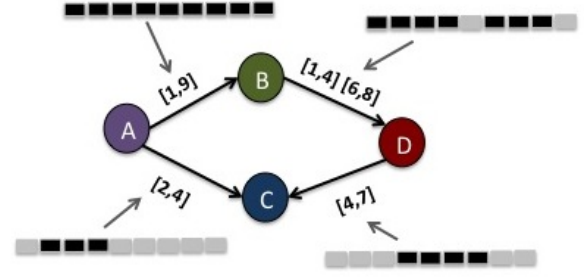

Figure 1. Twitter as an interval graph.

networks can be characterized by an abundance of measures [1]. When the additional dimension of time is included in the network picture, many of these measures need rethinking. So that the concept of geodesic distance cannot be limited to the number of *hops* separating two nodes but should also take into account the temporal ordering of links.

A *temporal path* $P_{u,v}$ in a temporal graph G is a sequence $P_{u,v} =< (v_1, v_2, t_1), (v_2, v_3, t_2), ..., (v_{k-1}, v_k, t_{k-1}) >$, where

- $(v_i, v_{i+1}, t_{init}, t_{end}) \in E$ is the i-th temporal edge on $P_{u,v}$,
- $1 \le i \le k$,
- $t_i + R + T \le t_{i+1}$,
- $t_{init} \le t_i \le t_{end}$,
- $n \le t_1$ and $t_{k-1} \le N$,
- $u = v_1$ and $v = v_k$.

The *temporal length* or *duration* $d_{P_{u,v}}$ of a temporal path $P_{u,v}$, is the number of snapshots from t_1 to t_{k-1}, i.e., $d_{P_{u,v}} = t_{k-1} - t_1$. Finally, the *temporal shortest-path* or *fastest path* is the minimum temporal length between two nodes u and v, defined as $min(d_{P_{u,v}})$.

Example 1. Considering the temporal network of Figure 1 and the parameters $W = [1, 9]$, $T = 0$ and $R = 1$, we can cite some examples of temporal paths:

Temporal Path	Duration	Is fastest path?
$P_{A,D} =< (A, B, 1), (B, D, 2) >$	1	yes
$P_{A,D} =< (A, B, 2), (B, D, 6) >$	4	no
$P_{B,C} =< (B, D, 1), (D, C, 4) >$	3	no
$P_{A,C} =< (A, C, 2) >$	0	yes

TABLE 1. EXAMPLES OF TEMPORAL PATHS

2.2. Temporal Centralities

Once defined temporal paths concepts, we need to revisit two important centrality metrics: *closeness* and *betweenness*. As nodes represent users in our context, we chose these local metrics to conduct our analysis in this paper.

The *closeness centrality* of a node is used to measure how close it is from the others nodes in the graph. For example, people with high closeness in a social network are in an excellent position to monitor the information flow –

they have the best visibility into what is happening in the network. The closeness centrality of a node v in G is defined as:

$$closeness(v) = \sum_{u \in V \setminus \{v\}} \frac{1}{min(d_{P_{v,u}})} \tag{1}$$

where $min(d_{P_{v,u}})$ is the *duration* of the *fastest-path* from v to u. If there is not any path $P_{v,u}$ then the summation term is 0. If $min(d_{P_{v,u}}) = 0$ (when traversal time $T = 0$), then the summation term is 2.

The *betweenness centrality* of a focal node is the fraction of fastest paths passing through it. In a social network, people with high betweenness have great influence over what flows, and not in the network:

$$betweenness(v) = \sum_{v \neq j \neq k} \frac{w_v(j,k)}{w(j,k)} \tag{2}$$

where $w_v(j,k)$ is the number of fastest paths between j and k that pass through v and $w(j,k)$ is the total amount of fastest paths between j and k.

3. Twitter Dataset

Although there are many Twitter datasets available in literature [10], [11], [12], we need one containing the information of *when* relationships start and end in the network. Only then we will have a complete temporal network of follower-followee relationships. Moreover, we chose Twitter due to rich information available that allow us to correlate with users behaviors, for example. Datasets like [13] are limited on structural information. To the best of our knowledge, there is no available Twitter dataset with temporal network structure.

We developed TCraw[1], our Twitter data collector. The architecture is composed by two crawlers that use Twitter Rest APIs[2] to get data. The first one, Data Crawler is responsible to collect our observation network. This is the name used to refer to nodes and edges that we choose to track. The observation network crawling is done according to the following steps:

1) Choose s users seeds on Twitter. This is the level 0 ($current\_level = 0$).
2) While $current\_level < MAX\_LEVEL$ do
   a) For each node n in $current\_level$, get m followers of n
   b) Set $current\_level = current\_level + 1$

Once stored entire observation network in file format, the Update Crawler is started. Its function is to update structural node information based on a given time interval U. For example, for $U = 24hs$, the temporal network is built with one day granularity. Since structural information is not available in Twitter (Twitter API does not provide historical information about when a user starts/end following other),

1. Available at: http://lsi.facom.ufu.br/~fabiola/temporal-networks
2. https://dev.twitter.com/rest/public

our dataset only makes sense from the moment Update Crawler is started. So, the observation network built from Data Crawler is our initial state.

Time-changing Characteristics of Data. Table 2 details statistics of our dataset[3]. It is important to notice that the dataset does not track nodes evolution, just edges activation/inactivation for the 144975 nodes. The total number of temporal edges has been computed considering that two edges of the type $(u, v, t_{init}, t_{end})$ and $(u, v, t_{init'}, t_{end'})$ are different.

Observation window W	[08/28/2015, 12/15/2015]
Update window (granularity) U	1 day
Max fanout (m)	10000
`MAX_LEVEL`	1
# nodes	144975
# edges in first day	837961
# total temporal edges	1222118
Avg # new follows/day	3492
Avg # unfollows/day	3657
# seeds (s)	27
# themes	9 (3 seeds each)
Themes related to seeds	politics, sport, news, religion, music, humor, fashion, health, TV

TABLE 2. TWITTER DATASET STATISTICS

As illustrated in Figures 2 and 3, the dataset is fairly dynamic, specially on October 2015. These observations endorse that the dataset has a time-changing characteristic, when considering edges.
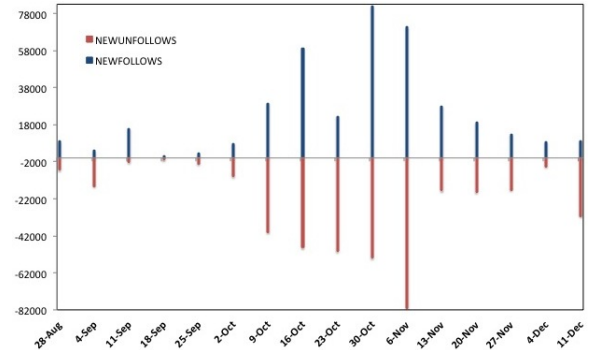

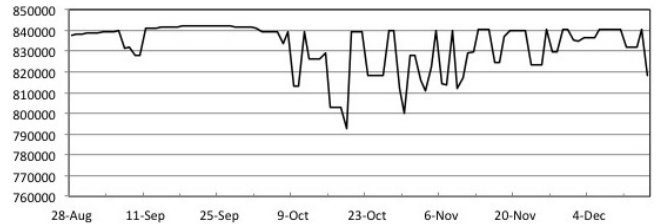Figure 2. Number of follows/unfollows aggregated per week

Figure 3. Number of active edges

3. Available at: http://lsi.facom.ufu.br/~fabiola/temporal-networks

Limitations. The strategy we have used for collecting the observation network is a limitation in our dataset. The idea of starting from seeds celebrities resulted in a network extremely unbalanced, with weak connections between users that are not seeds. Another limitation is the difficulty on tracking updates daily in our network. Twitter API has hard restriction policies, which resulted in a relatively small Twitter sample. The longer nodes or edges are considered, the higher should be the granularity (update window).

Even with these limitations, in what follows we show how interesting is the analysis of evolving centralities in our dataset. It is certainly a time-changing network structured data.

4. Network Evolution

How to analyze a temporal network? Which algorithms can be applied for this task? In this section, we present a baseline algorithm for mining Twitter temporal graph from evolving centralities perspective. The idea is to analyze temporal edges as streams.

4.1. Stream Representation of an Interval Graph

An interval graph can be represented as data stream. Intuitively, a stream is a sequence of all edges in G that come in order of the time each edge is created. When analyzing interval graphs, the existence interval of an edge $e = (u, v, t_{init}, t_{end})$ means that at any time inside this interval $[t_{init}, t_{end}]$, there may be a communication between the nodes u and v. For that reason, we first define the stream representation of e as a set of contacts.

Definition 1 (Contact). An edge $e = (u, v, t_{init}, t_{end})$ can be represented by a set of contacts $C_e = \{(u, v, t_{init}), (u, v, t_{init+1}), ..., (u, v, t_{end})\}$. A contact $c_{e,t} \in C_e$ is a triple (u, v, t) indicating that there may be communication between u and v at t.

The data stream representation of an interval graph is a sequence of contacts from all edges in G, ordered by the time of each possible contact. For example, if G has the following edges:

$\{(v_1, v_2, 4, 6), (v_1, v_3, 1, 3), (v_2, v_4, 5, 6), (v_4, v_5, 6, 8)\}$,

the corresponding data stream is:

$\{(v_1, v_3, 1), (v_1, v_3, 2), (v_1, v_3, 3), (v_1, v_2, 4), (v_2, v_4, 5),$
$(v_1, v_2, 5), (v_1, v_2, 6), (v_2, v_4, 6), (v_4, v_5, 6), (v_4, v_5, 7),$
$(v_4, v_5, 8)\}$.

This definition is different from the one adopted in [7]. In our proposal we convert the edges represented by means of time-varying intervals into set of contacts. The data stream representation is a sequence of all possible contacts from all edges in G that come in order of each contact time. On the other hand, in [7], the stream is a sequence of all edges in G, ordered according to their starting time.

We argue that this conversion is necessary for the analysis considering the meaning that we want to represent: the information in Twitter can be diffused as a result of a sequence of contacts among users.

4.2. Temporal Centrality Analysis

To compute the exact closeness or betweenness centrality, we need first to compute all pairs fastest paths, which is too expensive for a large graph. The Algorithm 1 is a baseline algorithm we propose for this task.

Algorithm 1 All pairs fastest paths detection

Input: A temporal graph $G = (V, E)$ in its data stream representation, $W = [n, N], R, T$
Output: For each node $v \in V$, a set L_v containing all fastest paths from v

1: **for** each incoming contact $c = (u, v, t)$
2:     **if** $t \geq n$ and $t \leq N$ **then**
3:         *// $P_{u,v}$ is a temporal path*
4:         $P_{u,v} \leftarrow \{c\}$
5:         *// L_u is a set of all temporal paths from u, initially $\emptyset$*
6:         $L_u \leftarrow L_u \cup \{P_{u,v}\}$
7:         **for** each $x \in V$
8:             **for** each $P_{x,z} = < (x, v_1, t_1), ..., (v_k, z, t_k) > \in L_x$
9:                 **if** $z = u$ **then**
10:                     **if** $t \geq t_k + R + T$ **then**
11:                         $P_{x,v} \leftarrow P_{x,z} \cup \{c\}$
12:                         $L_x \leftarrow L_x \cup \{P_{x,v}\}$
13: *// Removing cycle paths and paths that are not the fastest ones*
14: **for** each $x \in V$
15:     **for** each $P_{x,z} \in L_x$
16:         **if** $x = z$ **then**
17:             $L_x \leftarrow L_x - \{P_{x,z}\}$
18:         **else**
19:             **for** each $P'_{x,z} \in L_x$
20:                 **if** $d_{P_{x,z}} > d_{P'_{x,z}}$ **then**
21:                     $L_x \leftarrow L_x - \{P_{x,z}\}$
22: **return** L_x for each $x \in V$

Remarking on complexity analysis of Algorithm 1, we have that for each incoming contact, the computational cost to update the values is $O(V \times P)$, for P being the average number of paths between two nodes. In the end, there is the additional cost of removing cycles and paths that are not the fastest ones, corresponding to $O(V \times P)$ (from line 14). If we consider the total number of elements in the stream as C, we have a final complexity of $O(C \times 2(V \times P))$.

It is important to mention that the algorithm has a high spatial cost. To be able to calculate the exactly (not approximated) values of fastest paths, our algorithm keeps all paths stored while processing the stream. In what follows, we explore the aspects of memory size vs. stream size vs. high velocity processing temporal paths.

5. Experimental Analysis

All the experiments were performed with 32GB of main memory available. The values for retention and traversal time are $R = 1$ day and $T = 0$, respectively. The Twitter dataset granularity is of 1 day.

5.1. Varying Time Intervals

For computing the all pairs fastest paths, the input observation window W can affect several aspects, for cite:

overall running time, stream size and duration of fastest paths. We define different intervals to perform the analysis. As Twitter dataset has been collected from Aug 28th, 2015 to Dec 15th, 2015, in table 3 we summarize the adopted observation windows.

Period	# of intervals	Values
Weekly (WE)	15	$WE_1 = [09/01, 09/07], WE_2 = [09/08, 09/14], ..., WE_{15} = [12/08, 12/14]$
Fortnightly (FO)	7	$FO_1 = [09/01, 09/15], FO_2 = [09/16, 09/30], ..., FO_7 = [11/30, 12/14]$
Monthly (MO)	3	$MO_1 = [09/01, 09/30], MO_2 = [10/01, 10/31], MO_3 = [11/01, 11/30]$
Total (TO)	1	$TO = [08/28, 12/15]$

TABLE 3. OBSERVATION WINDOWS

We measure the execution time of Algorithm 1 for different observation windows. Figure 4(a) illustrates these results. The values for WE, FO and MO correspond to the average execution time in each group of windows. We can observe an exponential behavior compatible with the increasing number of incoming contacts C (Figure 4(b)). Our algorithm is dependent on the size of the observation window.

In Figure 4(c) we show how different the fastest paths values can be just varying observation windows. This endorses the time-varying aspect of Twitter network.
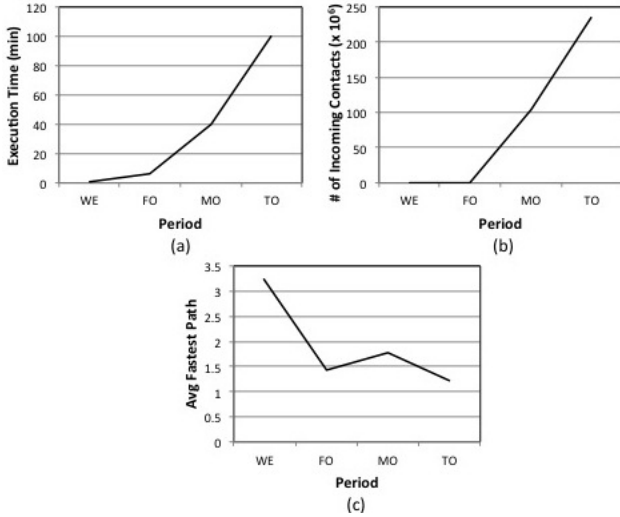

Figure 4. Results varying the size of observation window when running all pairs fastest paths algorithm

5.2. Evolving Centralities

The closeness centrality can be easily calculated from the return of Algorithm 1. With all pairs fastest paths and

their respective duration, the $closeness(v)$ is a straight sum of these values (see Eq. 1). In Figure 5(a) we can see the closeness value averaged across all users for different observation windows. The observation during short or large amount of time does not influence on closeness. These values depend mainly of the network behavior: as Twitter network is diversified and extremely dynamic, nodes' closeness vary accordingly.

Another interesting analysis is illustrated in Figure 5(b). Three users u_1, u_2 and u_3 were randomly selected and their closeness analyzed over fortnightly intervals. Remark that these users are not seeds. The graph shows that users are always changing their closeness.

As well as closeness, the betweenness centrality $betweenness(v)$ is a straight calculus from the fastest paths returned by Algorithm 1 (see Eq. 2). Furthermore, the nodes have the same behavior in varying their centrality values. In Figure 5(a) we have the betweenness averaged across all users for different intervals. And in 5(c) the variation for users u_1, u_2 and u_3.

Finally, we rank all nodes according to their centrality values (first positions for higher centralities). For this analysis we consider a sequence of incremental observation windows of the type $I_1 = [day1, day15], I_2 = [day16, day30], ..., I_7 = [day91, day105]$. The values in Table 4 suggest that nodes centralities are fairly dynamic and from one observation to the next, the node may have become more or less important.
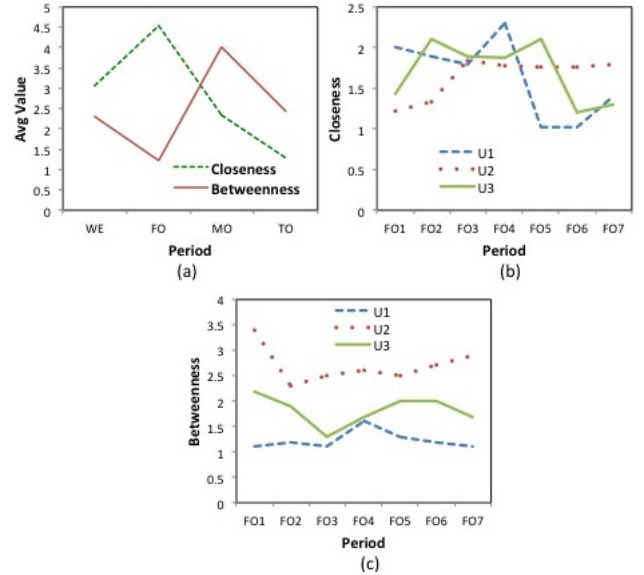

Figure 5. Evolving centralities observations

6. Discussion

We consider the strategy proposed in this paper for mining all pairs fastest paths as a pseudo-stream mining. The Algorithm 1 is extremely sensitive to the size of the stream, despite processing data as stream. Its high velocity

User	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	544	530	533	530	544	580	600
U_2	653	644	600	589	592	615	617
U_3	122	123	100	224	220	235	249

TABLE 4. RANKING POSITIONS VARIATIONS FOR CUMULATIVE OBSERVATION WINDOW

processing depends on the size of available memory, which technically does not fully characterize the stream mining problem [14]. On the other hand, our proposal is a first look in temporal networks over a real dataset from evolving centralities perspective. Despite of some works investigate the evolving centrality problem, they use approximation and sampling methods [5], [7], [9], [15]. To best of our knowledge, there is not a solution for exactly centrality values tracking, specially for evolving *betweenness*.

Applications. Analyzing evolving centralities in networks can be applied in diverse real applications. We highlight here two of them. First, the problem of contagion (information, influence and disease) where the interest is in observing not exactly people getting infected, but who infected them [16]. As contagion is a dynamic process, temporal networks can be applied in this problem and tracking evolving centralities can help in increasing the sales in marketing field, for example.

A second emerging application is on the analysis of user preferences and behaviors [17]. The analysis of evolving centralities can reveal patterns of influence and communications in social networks. For instance, these patterns help to understand how users' preferences evolve over time for more accurate recommendation systems.

7. Conclusion

Our purpose in this paper was to analyze Twitter from evolving network structure viewpoint. We have modeled Twitter as a temporal network and revisited the concept of shortest path considering the time dimension. We have shown how to compute closeness and betweenness centralities using fastest paths. We have developed an algorithm for mining streams of temporal networks and have used it to perform analysis over Twitter follower/followee network. Our findings have shown that analyzing Twitter as a temporal graph, models the behavior of real applications and is different from just considering static analysis.

Acknowledgments

References

[1] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[2] P. Holme and J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.

[3] P. Holme, "Analyzing temporal networks in social media," *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1922–1933, 2014.

[4] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora, *Temporal Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Graph Metrics for Temporal Networks, pp. 15–40.

[5] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, "Temporal distance metrics for social network analysis," in *Proceedings of the 2nd ACM Workshop on Online Social Networks*, ser. WOSN '09, 2009, pp. 31–36.

[6] D. Eppstein and J. Wang, "Fast approximation of centrality," *SODA*, pp. 228–229, 2001.

[7] H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, and Y. Xu, "Path problems in temporal graphs," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 721–732, 2014.

[8] C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, 2014.

[9] S. Tabassum and J. Gama, "Sampling massive streaming call graphs," in *Proceedings of the 2016 ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: ACM, 2016, pp. 923–928.

[10] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, "Characterising temporal distance and reachability in mobile and online social setworks," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 118–124, 2010.

[11] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[12] H. Kwak, H. Chun, and S. Moon, "Fragile online relationship: A first look at unfollow dynamics in twitter," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1091–1100. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979104

[13] C. Cattuto, M. Quaggiotto, A. Panisson, and A. Averbuch, "Time-varying social networks in a graph database: a neo4j use case," in *First International Workshop on Graph Data Management Experiences and Systems*. ACM, 2013, p. 11.

[14] J. Gama, *Knowledge Discovery from Data Streams*, 1st ed. Chapman & Hall/CRC, 2010.

[15] N. Santoro, W. Quattrociocchi, P. Flocchini, A. Casteigts, and F. Amblard, "Time-varying graphs and social network analysis: Temporal indicators and metrics," *CoRR*, vol. abs/1102.0629, 2011. [Online]. Available: http://arxiv.org/abs/1102.0629

[16] J. Yang and J. Leskovec, "Modeling information diffusion in implicit networks," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ser. ICDM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 599–608.

[17] J. Li, A. Ritter, and D. Jurafsky, "Inferring user preferences by probabilistic logical reasoning over social networks," *CoRR*, vol. abs/1411.2679, 2014. [Online]. Available: http://arxiv.org/abs/1411.2679