Detecting Events in Evolving Social Networks through Node Centrality Analysis

Fabiola S. F. Pereira[1], Sandra de Amo[1], and João Gama[2]

[1] Federal University of Uberlândia
{fabiola.pereira,deamo}@ufu.br
[2] University of Porto, LIAAD INESC TEC, Porto, Portugal
jgama@fep.up.pt

Abstract. Social networks have an evolving characteristic because of continuous interaction between users. Existing event detection tasks do not consider the analysis under a user-centric perspective. In this paper we propose to detect node centrality events, that is the task of finding events based on the position and roles of the nodes. We present a naive algorithm for detecting such events in network streams. Moreover, we apply our proposal in a case study, showing how node centrality events can be used for tracking user preferences changes.

1 Introduction

Social networks streams are dynamic networks that have a fast rate of edge arrival [1]. The analysis of such networks is especially challenging, because it needs to be performed with an online approach, under the one-pass constraint of data streams.

When considering the evolving characteristic of such networks, changes as the impact on communities or the impact on network structural parameters such as node degrees, needs to be analyzed. These changes are stated as event detection problems. Detecting events in evolving networks can be investigated under different perspectives: anomaly detection [3], burst detection [6], concept drift [14] or topic evolving in social streams [7].

Generally, these perspectives are related with the whole graph structure evolving behavior, as in anomaly and burst detection or related with the evolving nature of the content being discussed in the network (concept drift and topic evolving analysis). In this paper we focus on the analysis of nodes positions evolution – a user-centric perspective. At a high level, our goal is to identify the behavior of nodes evolution in the network. For so, we define *node centrality event detection*: the task of finding events based on the position of the nodes.

Our proposal is to maintain nodes centrality values summarizing topological information as real numbers, which allows us to leverage the changes in nodes roles. We perform analytical evolution analysis, always considering data streams constraints [9].

In order to illustrate how our problem can be applied in evolving social networks, we meet node centrality event detection with dynamics of user preferences. Preferences dynamics refer to the way a user evolves his or her preferences over time [13]. Looking for node centrality events in a Twitter interaction network, we were able to detect changes in the user preferences, thus just considering the topology of network.

Summarizing, this work makes the following contributions: (i) proposal of a technique for detecting events in network streams; (ii) empirical observations of the proposed technique over a Twitter dataset and (iii) a case study describing how the proposed technique can be applied for tracking user preferences dynamics in social networks.

2 Related Work

We highlight some related work in the directions of graph stream processing and event detection in networks. Our proposal is innovative when considering *event detection* from a *node-centric perspective* in a *stream processing* environment.

Processing graphs as streams is an incoming problem. The work [4] is one of the most complete work when considering data mining in evolving graph streams. The focus, however, is on mining closed graphs, not on event detection. In [8] a framework for processing graphs as streams is proposed for the link prediction task. This framework considers the cumulative grown of the graph, not addressing the space saving feature [9].

The most studied events in evolving networks are anomalies and bursts [7]. Anomaly detection refers to the discovery of rare occurrences in datasets. The most representative work in anomaly detection for dynamic graphs is [10]. It addresses the problem considering a time sequence of graphs (graph sequences). The focus is on faults occurring in the application layer of Web-based systems. First, they extract activity vectors from the principal eigenvector of dependency matrix. Next, via singular value decomposition, it is possible to find a typical activity pattern (in $t-1$) and the current activity vector (t). In the end, the angular variable between the vectors defines the anomaly metric. The network processing is through snapshots, not in a streaming fashion. Moreover, this Eigen Behavior based Event Detection (EBED) method is orthogonal to ours as it detects events in a global perspective of the network, while ours is node-centric.

Burst events are generally related to topic evolving detection and tracking [7]. These works are looking for events like hot buzz words, what are users' sentiments about a product release or how is a specific topic evolving. In [6] the goal is to track interest profiles in real time by detecting bursts in Twitter's social media stream in real time using linear regression. These approaches are orthogonal to ours because are focused on the content of the network (texts, topics) not in the topology evolution analysis. The work [2] incorporates network structure in event discovery over purely content-based methods. Each text message is associated with at least a pair of actors in the social network. The events detected are also related with topics evolving. Finally, in [15] the authors consider the problem of mining activity networks to identify interesting events, such as a big concert in a city, or a trending keyword in a user community in a social network. The algorithms are founded in geo-spatial event detection information. Any stream processing strategy is addressed.

3 Detecting Events in Network Streams

The problem of detecting events in evolving networks can be investigate under different forms: anomaly detection [3], burst detection [6], concept drift [14]. Our focus is on *event detection from nodes centralities*. We formalize the problem as the task of detecting changes in nodes centralities values over time, according to some centrality function. As example of centrality function we can cite: betweenness, closeness, degree, eigenvector etc. The foundation of our method is the technique of change point detection [16].

Definition 1 (Network Stream). *A network stream is a continuous and temporal sequence of edges $S = e_1, ..., e_n, ...,$ such that each edge $e_i = (u, v, t)$ corresponds to an interaction from node u to node v on time t. We define V the set of all nodes that arrived in the stream since the beginning of the observation. E_t is the set of edges that arrived on t.*

Definition 2 (Node Event). *Let us define a node centrality function $c_t : V \to \mathbb{R}$ that assigns a nonnegative value $c_t(s)$ for each node $s \in V$ on time t. This centrality function is computed considering the network $G = (V, E_t)$ and can be calculated according to any node centrality metric (betweenness, degree, closeness, PageRank, ...). A node event $\mathcal{E}_{s,c,t}$ is defined as*

$$\mathcal{E}_{s,c,t} = \begin{cases} 1, & |\bar{c}_{t-1}(s) - c_t(s)| > \theta \\ 0, & otherwise \end{cases} \tag{1}$$

where

- *$s \in V$,*
- *t is the time of the analysis,*
- *$\theta \in \mathbb{R}$ is a threshold value,*
- *c is the centrality function and*
- *$\bar{c}_t = \frac{1}{|W|} * \sum_{i=t-|W|+1}^{t} c_i(s)$, for W being the sliding window ($\bar{c}_t$ is the average of all centrality values of s inside W).*

The intuition of above definitions is: we detect an event for node s at certain time t if the centrality value of s had a high variation in relation to its past centrality values. In Algorithm 1 we present a sketch of the node centrality event detection task. We adopt the sliding window stream processing strategy.

The most important task is to compute the centrality function $c_t(s)$ (line 8). This specific function is not computed in streaming fashion, as we consider just the edges of the current instant t. So, classic batch algorithms can be applied in this task [17]. In fact, the computation of node centrality in streaming environment is an open challenge for many centrality functions [11]. The computational cost and scalability of this algorithm is proportional to the window size and the time t granularity.

Algorithm 1 Sketch for detecting events through node centralities

```
Input: Sliding window $W$, threshold $\theta$, centrality function $c$, network stream $S$
Output: A vector $\mathcal{E}$ containing the events detected for all nodes in $V$ at any time $t$
$V \leftarrow \emptyset$
for all time instant $t$ do
    $E_t \leftarrow \emptyset$
    for all $e_i = (u, v, t)$ arriving in the stream do
        $E_t \leftarrow E_t \cup \{e_i\}$
        $V \leftarrow V \cup \{u, v\}$
    Add $G = (V, E_t)$ in $W$
    Compute $c_t(s)$ for all $s \in V$, considering $G = (V, E_t)$
    for all $s \in V$ do
        if $|\bar{c}_{t-1}(s) - c_t(s)|> \theta$ then
            $\mathcal{E}_{c,t}[s] = 1$
        else
            $\mathcal{E}_{c,t}[s] = 0$
    Slides $W$
```

Though naive, this event detection approach is able to report changes in the network without complex text analysis, just observing the topology and nodes position. As we will show in the following, detecting such events provide meaningful evidences of network evolution. However, we are not able to gain insight about what kind of event we are detecting (drift, burst, anomaly).

4 The Evolving Social Network

4.1 Dataset

Folha de São Paulo (or Folha, for short) is one of the most influential newspapers in Brazil. Taking advantage of the fact that Twitter is widespread in the country, we performed our analysis over the news domain in Twitter social network. We collected a large body of tweets from Folha over the course of 3 weeks, starting in June 24, 2016. Our data collection strategy was as follows.

First, we used Twitter's streaming API to collect all tweets related to the newspaper (user @folha). Thus, our dataset consist of tweets about the news tweeted by Folha newspaper, the retweets and all inherent information mentioning these news. Next, we built the following interaction network: nodes are Twitter users. One edge from user u_1 to u_2 means that u_2 retweed on t some text originally posted by u_1, i.e. edges represent the information flow. The edges are temporal and just exist in the moment of the interaction (time t), then they disappear. Fig. 1 illustrates the evolving aspect of our network. The topics represented by colored edges were obtained using LDA. In Section 6 we detail this process.

In all, we collected 200806 tweets, 78944 nodes (users) and 108133 distinct edges considering the 3 weeks of observation period. An important characteristic of our network is that it has a low average path length. This is consequence of the fact that in Twitter a retweet always comes from the original post, not mattering from where the user read that post – from the user who originally posted it or from an intermediate user who already retweeted it. On average, the path length is 1.033.

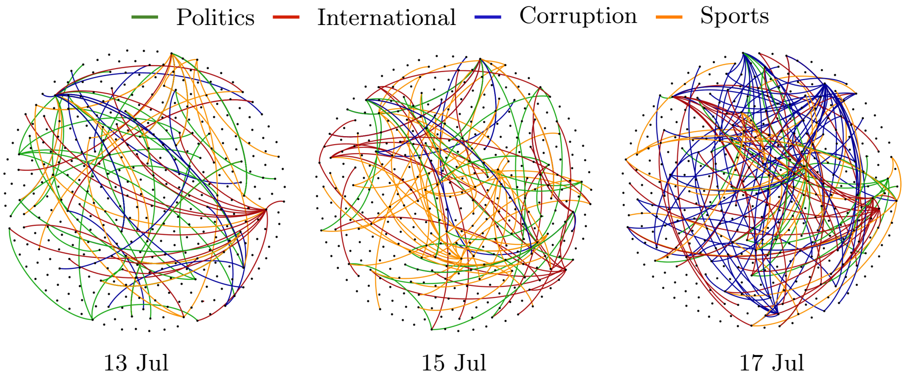


Fig. 1. Snapshots of samples of the evolving interaction network. Nodes are Twitter users. One tie from user u_1 to u_2 means that u_2 retweed at t some text originally posted by u_1. The colors represent topics that users are talking about at t. The samples were built by filtering nodes with degree between 50-22000 and edges representing the 4 most popular topics. Each snapshot corresponds to 1 day time-interval. This figure highlights the *edges* evolving aspect. Nodes are not evolving for better visualization.

4.2 Network Semantics

We analyzed the evolving behavior of each node in the network considering the closeness centrality measure [17]. *Closeness* is related to the visibility of a node in the network. It is the capacity of a node the reach the others in a fast way. Thus, a high closeness value means a good information spreading capacity.

In our context, we identify three types of user: *consumers*, *producers* and *consumers&producers*. *Consumers* are the users who most often just retweet, not publishing any new content. Generally, they have low closeness values. *Producers* are always publishing popular tweets and have a medium closeness value. Finally, the *consumers&producers* have a high activity in the network, tweeting and retweeting all the time. These users have the highest closeness values.

As example, let us consider the scenario illustrated in Fig. 2. Events can occur with any type of user, meaning that their usual role changed at that moment. User 3 has a typical *consumer* behavior until time $t6$. Just retweeting or even with no activity in the network. From time $t7$ user 3 presents a different behavior, which can be a persistent change or an ephemeral behavior. Thus, an event occur around $t7$ and $t8$. User 1 is clearly a *producer* from $t1$ to $t3$. And users 2 and 4 are *consumers&producers* during the whole observation period.

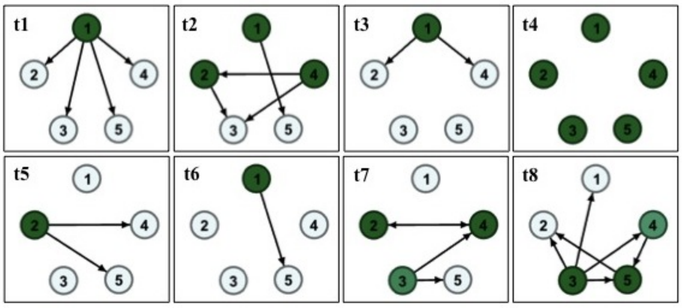


Fig. 2. Example of evolving behavior with closeness centrality. The darker nodes, the greater the centrality.

5 Empirical Analysis: Detecting Events with Closeness Centrality

We analyzed the evolving behavior of each node in the network considering closeness centrality measure. In Fig. 3 we show the centrality evolving behavior for three different users, one of each type (*consumer*, *producer* and *consumer&producer*). It is possible to distinguish that, generally, the types of users are related with their closeness centrality.

5.1 Influence of Parameters Setting

The balance between window size $|W|$ and threshold θ determines what we call of *nature of the event* that we are detecting.

θ adjusts the intensity of the events, varying from smooth to drastic events. In Fig. 4 we present an analysis for user $u4$, with θ assuming $0.1, 0.2$ and 0.5 and $|W|= 4$. We chose $u4$ due to its high activity level in the network and $|W|= 4$ as an intermediate value according to our observation period.

As expected, when considering smooth variations more events were detected. Drastic events indicate that the user changed drastically his role in the network. Around day 15 the events indicate that $u4$ leaves a central position as a *consumer&producer* to assume a *consumer* role.

Now, analyzing the impact of the window size $|W|$, in Fig. 5 there are the events detected for $|W|$ assuming 2, 4 and 10, $\theta = 0.2$ and user $u4$. Varying $|W|$ means that we are considering the recent past for low values (short-term events) or a big historic for high values (long-term events). As our dataset is relatively short, in these experiments the window size variation did not result in interesting findings. Short-term events are not interesting in our context due the high variation of centrality values in the network. Considering a mid-term period ($|W|= 4$) reflected better the evolving user role.

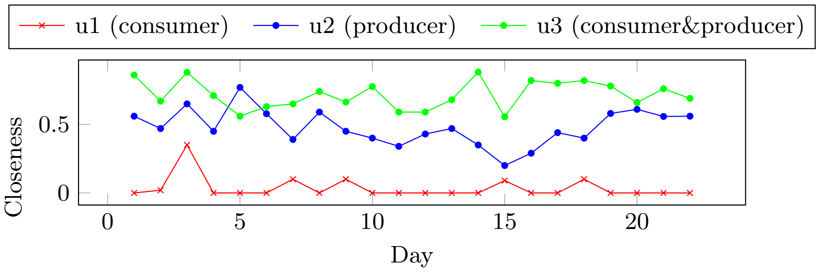


Fig. 3. Closeness evolving for three different types of user.

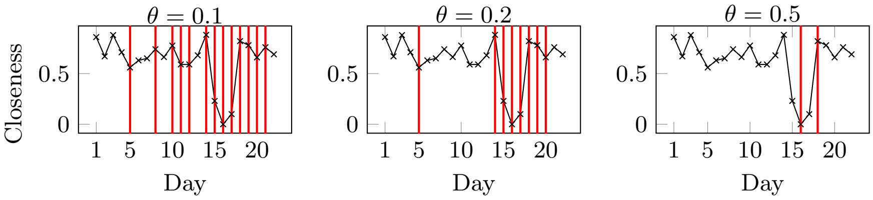


Fig. 4. Impact of θ (intensity of the events) for a high activity user. Detected events are highlighted in red lines.

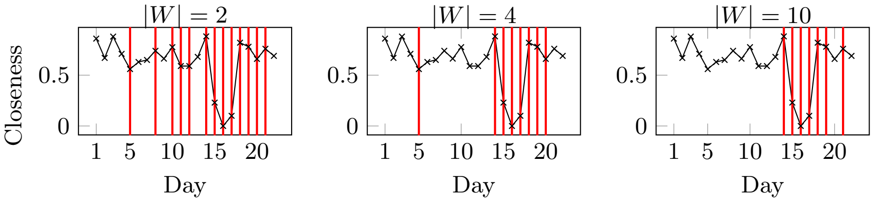


Fig. 5. Impact of $|W|$ (window size) for a high activity user. Detected events are highlighted in red lines.

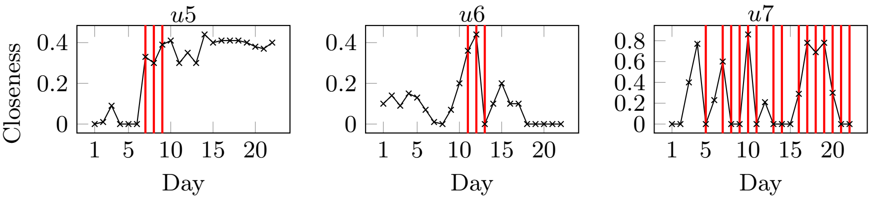


Fig. 6. Detected events highlighted in red lines for three different users.

5.2 Detected Events Analysis

From the previous analysis, we consider $\theta = 0.2$ and $|W|= 4$ as the default values. Thus, here we are interested in smooth mid-term events. In Fig. 6 we present the events detected for three users $u5$, $u6$ and $u7$. For users $u5$ and $u6$ it is possible to distinguish that the sequence of detected events reflects the moment they change their roles. In case of user $u5$ the change is permanent and for $u6$ the change is just ephemeral. However, in the case of $u7$, a lot of events were detected sequentially, reflecting a behavior of intermittent activities in the network. In fact, a weak point in our event detection method is this: just observing the events, we are not able to distinguish if the events are persistent, if they reflect a burst in the network or if they are ephemeral.

6 Case study: what detected node centrality events could mean?

Besides just reflecting changes in the network topology, we can interpret the events detected by analyzing the meaning of the centrality metric used so far. In our case study, nodes with high *closeness* are influential users that posted about topics with high interest. If a user u starts to occupy (or leaves) a central position this could mean that (i) his activity is high (low) and (ii) his activity is (not) interesting for the other nodes. This behavior could indicate a preference change for u, i.e., he probably changed his interests and is posting about different topics. We now investigate to what extent we could establish the correlation between: events with closeness centrality (network topology) and preference change behavior (network content).

6.1 Preliminaries

User preference is a specific type of opinion, that establishes an order relation between two objects. For example, when a user says: “I prefer sports than education”, we clearly identify his preference to sports subjects over education ones. These preference order relations (or preferences, for short) respects the irreflexive and transitive properties. When analyzing user preferences over time, we can discovery interesting patterns of users’ behavior.

In this case study, we are interested in detecting what are the moments that users changed their preferences. For example, let us suppose that on day 24, user A prefers to read/post/share on his social network news about *sport*, but between *celebrity* and *religion* topics, he is in the mood of *celebrities.* On the following days, A's preferences practically do not change, just appearing a preference of *celebrity* over *corruption.* However, on day 30, A's presented a preference change, as *corruption* became preferred over *celebrity.* So, we have detected a preference-change.

These definitions were formally proposed in [13]. The detection of preference-change events is based on the existence of inconsistencies in the temporal preferences of a user. These inconsistencies appear if the resultant composition of user preferences does not hold the irreflexive property.

6.2 Extracting preferences

In order to discover what users are talking about in the network, we performed topic modeling with LDA algorithm [5] considering all tweets of the entire observation period. We got a total of 15 topics, that then were manually grouped in 7 more general topics, as detailed in Table 1.

The 7 general topics are the domain of preferences. The intuition in this preference mining process is: if user u tweets (or retweets) about *politics* on time t, then u has more interest in *politics* over the remaining topics in that moment. Thus, we mined preferences of the type: $politics \succ_u^t celebrity$, $politics \succ_u^t sports$ and so on. We also considered a weight based on the number of tweets posted in the same time (our time granularity is of 1 day. So, one user can post many tweets on t). If the user posts three times about *corruption* and two times about *sports* on t, then we can establish a preference order between *corruption* and *sports* ($corruption >_u^t sports$ and *sports* is preferred over the remaining topics). It is worth to mention that this preference mining process did not consider the network as a stream. We extracted the preferences of all users based on the global configuration of the topics during the whole period of observation.

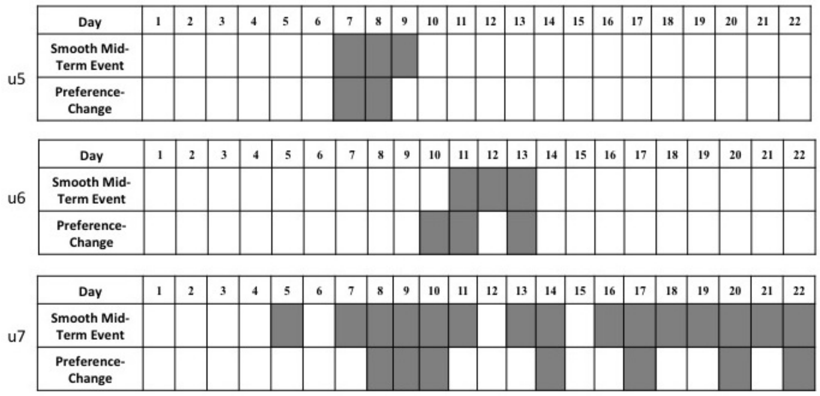
Topic	**Issues**
Politics	Impeachment, Eduardo Cunha, Temer, Coup, Dilma
International	Terrorism, Brexit
Corruption	Sergio Moro, Lava Jato, Paulo Bernardo
Sports	Olympic Games, Eurocup
Security	Rio de Janeiro
Education	Science without Borders
Celebrity	Luana Piovani

Table 1. Topics about Brazilian news extracted from Twitter dataset. Issues are the keywords that guided each topic creation.

6.3 Preference-Change Events

Remarking on Figure 7, the preference change events are highlighted for each user. These events were found following the strategy describe in Section 6.1. Basically, they describe the respective users changing their interests over time about the 7 topics of Brazilian news.

For the three users of our case study, we found that changes in preferences occurs around the same time of closeness centrality events. According to [12] there are many factors that influence on preferences dynamics. In this case study, we illustrate that applying closeness centrality event detection can be a good strategy for mapping user preferences changes just observing the evolving topology of the network, without any content information. We did not try to understand why users are changing their preferences. We just found the events that indicate these changes.



	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
u5	Smooth Mid-Term Event							■	■	■													
	Preference-Change							■	■														

	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
u6	Smooth Mid-Term Event											■	■	■									
	Preference-Change										■	■		■									

	Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
u7	Smooth Mid-Term Event					■		■	■	■	■	■		■	■		■	■	■	■	■	■	■
	Preference-Change								■	■	■				■			■			■		■

Fig. 7. Evolving behavior of three users considering the closeness centrality. Detected events/preferences changepoints are highlighted.

7 Conclusion

In this paper we introduced the notion of event detection based on node centrality. We proposed an algorithm to detect these events in streams of social networks. Empirical observations in an interaction network, built from the Twitter dataset of Brazilian news, were performed to validate our proposal. We also presented a case study showing how node centrality events can be applied for tracking user preferences changes.

A lot of work remains to be done. We intend to (i) explore a large range of centrality metrics; (ii) propose a more robust algorithm for node centrality event detection, considering other factors and (iii) compare our algorithms with state-of-the-art event detection streaming algorithms.

Acknowledgments This work was supported by the research project "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact / NORTE-01-0145-FEDER-000020", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF) and by European Commission through the project MAESTRA (Grant number ICT-2013-612944). We also would like to thank the Brazilian Research Agencies CAPES, FAPEMIG and CNPq for partially supporting this work.

References

1. Aggarwal, C., Subbian, K.: Evolutionary network analysis: a survey. ACM Computing Surveys 47(1), 10–36 (2014)
2. Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: 12th SIAM International Conference on Data Mining, USA. pp. 624–635 (2012)
3. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Mining and Knowledge Discovery 29(3), 626–688 (2015)
4. Bifet, A., Holmes, G., Pfahringer, B., Gavaldà, R.: Mining frequent closed graphs on evolving data streams. In: 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 591–599. KDD '11 (2011)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (Mar 2003)
6. Buntain, C., Lin, J.: Burst detection in social media streams for tracking interest profiles in real time. In: 39th International ACM SIGIR conference (2016)
7. Cordeiro, M., Gama, J.: Online Social Networks Event Detection: A Survey, pp. 1–41. Springer International Publishing, Cham (2016)
8. Fairbanks, J., Ediger, D., McColl, R., Bader, D.A., Gilbert, E.: A statistical framework for streaming graph analysis. In: IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining. pp. 341–347. ASONAM '13 (2013)
9. Gama, J.: Knowledge Discovery from Data Streams. Chapman & Hall/CRC (2010)
10. IDÉ, T., KASHIMA, H.: Eigenspace-based anomaly detection in computer systems. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 440–449. KDD '04 (2004)
11. Kourtellis, N., Morales, G.D.F., Bonchi, F.: Scalable online betweenness centrality in evolving graphs. In: 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016. pp. 1580–1581 (2016)
12. Liu, F.: Reasoning about Preference Dynamics. Spring (2011)
13. Pereira, F., de Amo, S., Gama, J.: On Using Temporal Networks to Analyze User Preferences Dynamics (2016)
14. Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F.: Anomaly detection in dynamic networks: a survey. Wiley Interdisciplinary Reviews: Computational Statistics 7(3), 223–247 (2015)
15. Rozenshtein, P., Anagnostopoulos, A., Gionis, A., Tatti, N.: Event detection in activity networks. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1176–1185. KDD '14 (2014)
16. Wei, W., Carley, K.M.: Measuring temporal patterns in dynamic social networks. ACM Transactions on Knowledge Discovery from Data (TKDD) 10(1), 9 (2015)
17. Zafarani, R., Abbasi, M.A., Liu, H.: Social Media Mining: An Introduction. Cambridge University Press, New York, NY, USA (2014)