Data Mining for Online Social Networks

Proseminar Data Mining

Linus Kreitner
Faculty for Informatics
Technical University Munich
Email: linus.kreitner@tum.de

***Abstract*— In this survey we outline the state of art in data mining for big Online Social Networks (OSNs) such as Twitter and Facebook. The paper covers the basics of graph mining and presents several network properties for OSNs. We take a look at multiple algorithms addressing standard problems in network analysis such as community detection and information spreading. Therefore, we consider modern algorithms, for instance Muric et al.'s method to identify a networks most influential nodes by treating the graph as an LTI system and other classic algorithms like Girvan and Newman's community detection strategy. Finally, we will discuss several application areas like event detection and sentiment analysis where data mining shows promising results.**

***Index Terms*— Data Mining, Online Social Network Analysis, Community Detection, Information Spreading, Sampling, Big Data, Survey**

I. Introduction

The interest in Online Social Networks, in the following called 'OSN', has reached a peak in human history. Never before more people were using online services to connect with each other. In April 2017, the US American company Instagram announced that starting from this day, they count more than 800 million monthly active users [1]. The micro-blogging service Twitter recently published statistics claiming that the number of daily active users increased by 10 percent in the first quarter of 2018 compared to the previous quarter [2].

This large number of users coming together in a social network provides a huge amount of data which can be observed. Companies hope being able to use the data for market research, journalists to detect trends and current events all over the world and politicians try to analyse the people's opinion toward a topic in order to obtain their approval. Data mining is used to gather raw data from a network, isolate the needed information and transform it into something more readable.

This survey gives an overview over the different aspects of Data Mining for OSNs and the current state of art about how the analysis is done. Therefore, this paper will include a look upon numerous interesting papers about hot topics from the last years. Specifically, this survey is structured as follows:
In the first chapter we will take a look on how to model OSNs using graphs and techniques to receive the best abstraction of real networks. We will also build a level of terminology to describe specific properties of graphs and to better understand the complex algorithms in the second chapter. After that, the survey will present different application areas, where data mining can be extremely useful, such as intention mining and news gathering. The last chapter shortly sums up what we have discovered and lists the main findings.

II. Modelling

This chapter will present different ways how an OSN can be represented and also explains several technical terms.

A. Terminology

In this section, basic concepts of modelling networks are presented. For a broader view see [3].
A network often is represented as a *Graph* $G = (V, E)$, where for the purpose of this survey the edges E are mostly bidirectional, since most relations in OSNs are symmetrical [4]. The *neighbourhood* of a node is defined as the set of all vertices that can be reached from the node. The *density* of a graph is given by the fraction of the number of edges and the number of vertices. The *local cluster coefficient (CC)* is a measure for the connectivity of a node and is defined in [5] for a directed graph as follows:

$$C_i = \frac{n}{k_i * (k_i - 1)} \tag{1}$$

k_i denotes the number of neighbours of node i and n the number of connections among these neighbours. A real world example for the CC could be the likelihood of one's friends being friends as well. The average CC of all nodes is often referred to as the *global cluster coefficient* and is used for community detection. For the term *closeness* of a node exist several definitions, collected in [3]. Sometimes the closeness factor simply depends on the node's degree, or its neighbours degree, but it could also stand for the number of (shortest) paths through this node, which in this survey will be called *betweennes*.

To compute different algorithms on a network, the graph is also often transformed into an adjacency matrix $(a_{ij}) \in \mathbb{R}^{|V|x|V|}$ where all nodes V_i are matched with each other. a_{ij} denotes a 1 (or the weight in a distance matrix) if the edge between V_i and V_j exists, else 0.

B. Graph Patterns

Since OSNs are services that allow humans to interact with each other, we can use the topology of a network to determine social structures, such as communities or outsiders. Referring to [7], a community can be understood as a subgraph $G_C \subseteq G$ inheriting vertices that are closer connected to each other

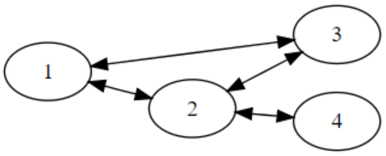
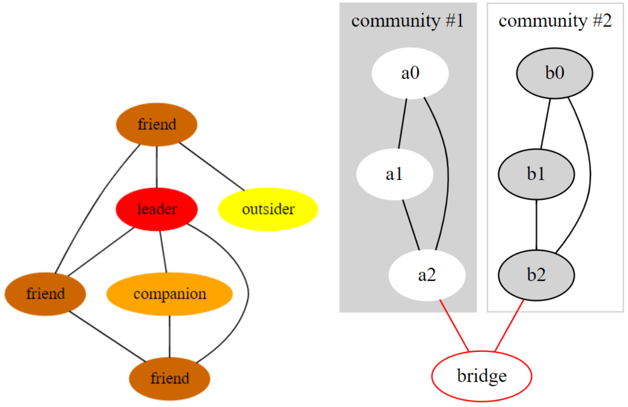


Fig. 1: C_1 to C_4: 1, $\frac{1}{3}$, 1, 0. Graph generated with [6].

than to the rest, for example a group of friends in a university network. An ideal community is called a *clique* meaning that all nodes are connected to all other nodes. The problem now is to identify different communities in the network simply based on the graph and a set of characteristics that a community should fulfil, like CC, betweennes or minimal number of vertices. Such communities can have social structures bound to them [Fig. 2], e.g. hierarchical ones, where every node has a role. Communities can be connected trough a *bridge*, or can overlap, which makes them harder to detect. Specific detection algorithms will be discussed in chapter 2.



(a) Weighted membership inside a community.

(b) two triangle communities connected by a bridge.

Fig. 2: Two possible community characteristics. Graphs generated with [6].

In [5] Watts and Strogatz define the term of so called *Small World Networks*, based on the Small World phenomenon or better known as the *Six Degrees of Separation* [8]. This theory states that every person in the world, on average is connected to all others by only a small amount of edges. Comparing to Watts and Strogatz, a small world network is defined as having (1) a high average cluster coefficient C and yet (2) small characteristic path lengths L, L being the mean shortest path length over all vertices. A small value of L allows the network to rapidly share informations between distant nodes. Random networks often have this property, whereas lattice graphs usually have a high CC.

The authors propose a method to create networks with C close to that of lattice graph and L to that of a random network. The basic idea is to randomly rewire a connection between nodes in a grid network with the probability p, transforming it to a random graph [Fig. 3]. For an interval of p during this procedure, the network will show the characteristics of a small world network with $C \gg C_{rand}$ and $L \approx L_{rand}$.

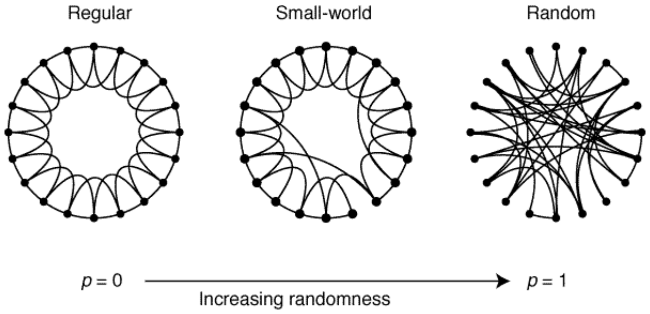


Fig. 3: Random rewiring of a lattice network with probability p [5].

In [9] Telesford et al. describe a network metric ω, which decides whether the given graph is a small world network. Therefore, ω is defined as follows:

$$\omega = \frac{L_{rand}}{L} - \frac{C}{C_{lattice}} \tag{2}$$

This metric shows good results in identifying such networks even in large data samples, however, the computational costs in creating a similar lattice and random graph often is too expensive.

Some research about the *power law distribution, scale free network* and *small word network* properties for real OSNs, such as *Flickr* and *YouTube*, was done by Mislove et al. in [4]. In power law networks the probability for a node being of degree k is proportional to $k^{-\gamma}$, for large k and $\gamma > 1$. Scale free networks are a subset of power law networks, where high degree nodes, so called *hubs*, tends to be connected to other hubs. Their investigation shows that the tested user graphs tend to fulfil the requirements for small world networks, scale free networks and therefore power law networks.

C. Modelling of Large Networks

As mentioned in chapter I, today's OSNs inherit millions of users, leading to a vast network with an even bigger amount of edges. Many algorithms cannot perform on Big Data without using an unreasonably amount of computational time and space. Therefore, several strategies were developed to handle large scale networks.

In [10] the authors outline the pipeline of multimedia big data analysis, i.e. how specific informations can be drawn from a network. First the raw data need to be extracted out of a network, for example by using special API's [11]. After that, the *Data Pre-processing* begins, which according to the source takes about 60% of a data scientist's time. In this process the data is cleaned, which means its inconsistencies are eliminated, missing values are replaced, outliers are identified and in brief, the data quality is enhanced.

The pre-processing is followed by the *Data Reduction*, which is necessary to shrink the amount of data an algorithm has to work with. We will discuss 2 different approaches on how this reduction can be achieved:

The so called *Snowball* algorithm is proposed in [12] and is a method to decrease a graph's number of nodes by sampling. The idea can be summarized as follows: A subset S of k individuals is chosen from the network (usually randomly) and forms the the 0^{th} active stage. In each step, every node in the active stage chooses k individuals, for example his k best friends. Those individuals that never were part of an active stage, but were named by the last active set, form the new active set [Fig. 4].

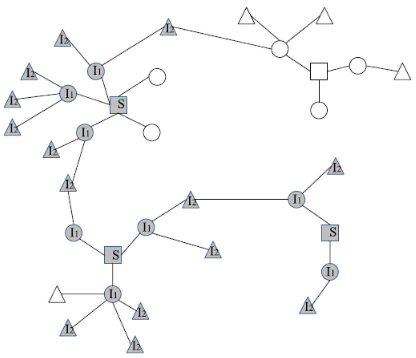


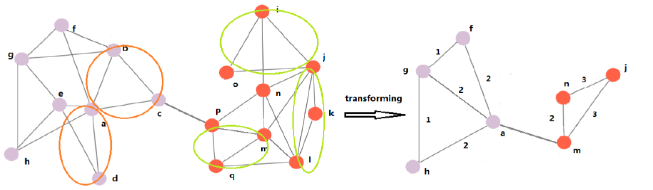
Fig. 4: Snowball-sample after the two iterations $I1$ and $I2$, starting with the initial nodes S (squares). The white nodes are not in the sample [3].

A crucial aspect of this method is the choice of the initial subset S. As we can see in [Fig. 4], a part of the graph was not included in the sample because there was no start node nearby. Snowball sampling is often named a fine method to maintain a graph's connectivity inside the sampling area, but it suffers from "boundary bias" [13].

Another propose is described in [14], where Ying et al. present a new strategy to calculate the spectral clustering for large-scale social networks. The algorithm of spectral clustering will be discussed in chapter III-B, but now we will take a look at the so called *pre-coarsening sampling* of this method. Their idea is to shrink the amount of nodes by replacing each triangles in a graph by one single node. To maintain the informations about the lost edges, weights are attached to every edge, representing the number of real connections to this node (see [Fig. 5]). This method leads to a weighted matrix $\tilde{A}$, inheriting the original graph's cluster topology structure. After coarsening, the matrix samples p rows and q columns into a new matrix $\bar{A} \in \mathbb{R}^{pxq}$ following the probability distribution

$$p_i = \frac{\left|A^{(i)}\right|^2}{\|A\|_F^2}. \tag{3}$$

The authors claim that this pre-processing technique combined with the Nyström method outperforms the state of art spectral clustering algorithms. However, this sampling method may not be the best solution for other algorithms. In [13] several more sampling algorithms are discussed.



(a) Before coarsening

(b) After coarsening

Fig. 5: The transforming of a link relation based network by shrinking triangles. [14]

III. Algorithms and Tools

In this chapter we will take a closer look at a few algorithms which are used to solve specific problems in data mining, such as community detection and the description of information flow inside a network. Therein we will consider a compound of older and newer approaches to outline the state of art in these areas.

A. Information spreading

A common topic of interest in marketing is the question how news spread among people or a product among customers. In the dissemination of information we can use the principle of the *forest fire model* [15] to describe the flow inside a network. A simple description is that starting from an initial set of vertices, each node has a *forward probability p* to light its neighbours, i.e. inform that node, leading rapidly to a completely burning network.

An efficient algorithm for identifying the most influential nodes, meaning vertices that are able to spread informations in a short amount of time, is proposed in [16] by Muric et al. Their approach is based on treating the network as an *LTI (liniar time-invariant) system*, which basically means the response of a system can be obtained by folding the input signal $x(t)$ with a transfer function $h(t)$. Then the (1) input response and (2) the step response of the system are calculated, addressing the idea - referring to the forest fire model - that nodes either can stop burning after a wile, or not, respectively [Fig. 6].

Muric et al. then define the *node imposed response NiR(i)* of node i as

$$NiR(i) = \frac{S_i - S_{min}}{S_{max} - S_{min}}, \tag{4}$$

S_i being the maximum value of the step response for i. This, so the authors claim, is an accurately measure of the nodes spreading power. Furthermore, it is stated that this method outperforms many other strategies, like betweennes and degree centrality. A downside of this method is that the progress requires many informations about the network's topology and characteristics.

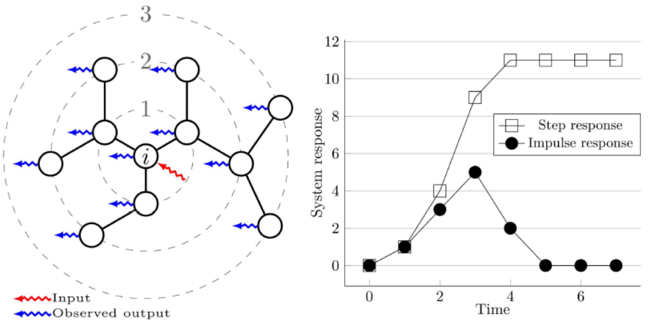


Fig. 6: A graphic visualisation of the input and step response for a small tree graph [16].

However, persuading people is not as easy as informing, thus we have to create new models for the spread. In [17] Kempe et al. discuss several diffusion models and algorithms to simulate the spreading. The *Linear Threshold Model* specifies that each node v is assigned a *threshold* $\Theta_v \in [0,1]$ that determines the number of v's neighbours, which must become active in order to set v on active. Hence a node turns activated if:

$$\sum_{w \in W} b_{v,w} \geq \Theta_v \tag{5}$$

W is v's neighbourhood and $b_{v,w}$ the weight of the edge, i.e. how much a neighbour influences v. $b_{v,w}$ is set to $\frac{c_{v,w}}{deg(v)}$, $c_{v,w}$ being the number of parallel edges between v and w. The *Independent Cascade Model* defines that an active node only can influence each neighbour w with the probability $w_{v,w}$ once, stays active afterwards, but is not allowed to influence others again. $w_{v,w}$ is set to $1-(1-p)^{c_{v,w}}$, with p being either 1% or 10%, respectively.

After that, the *influence* $\sigma(A)$ of a node is defined as "the expected number of active nodes at the end of the progress".

At last, four different algorithms are benchmarked using these different settings, analysing their performance. They competed their newly created greedy method against a (i) heigh degree, (ii) central and (iii) random method to choose what node should be influenced next, where the greedy method and the high degree algorithm showed the best efficiency [Fig. 7].

The authors claim that the *influence maximization problem*, being the challenge to find a k set of nodes with maximum influence, is $\mathcal{NP}$-hard. While their work shows efficient ways on how to spread informations in a network, the aspect on how to choose the initial set of vertices A remains unanswered.

B. Community Detection

This section presents multiple community detection (CD) algorithms and defines a guideline on how to evaluate them. One needs to differ between CD, *graph partitioning (GP)* and *graph clustering (GC)*, since there are often used inaccurate for the same manner. Referring to [7], GP divides a graph into n groups with size k so that the number of edges between the groups is minimum, where n and k are given as input. Many GC algorithms also require the number of expected clusters as input, whereas in CD none of these informations are needed. Due to the large size of OSNs, discussed in section II-C, one can almost never specify the number of clusters or their size in front, hence such algorithms can not be used effectively for OSNs.

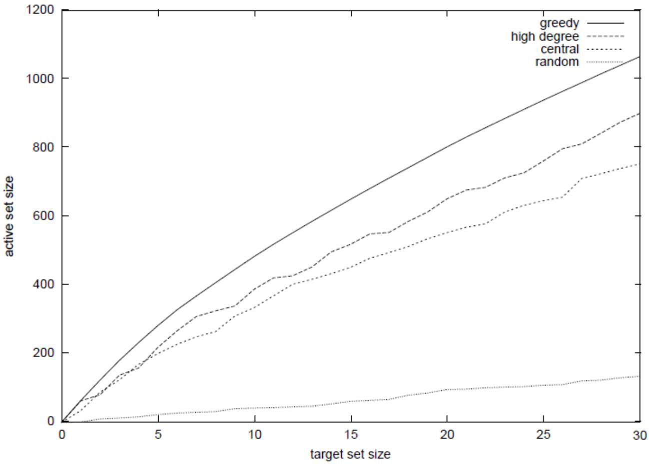


Fig. 7: The results for the linear threshold model show that the greedy method outperforms the other algorithms in persuading people. These results are representative for most other conducted tests [17].

Newman and Girvan present an algorithm in [18] that divides the network into multiple subgraphs by removing specific edges. Their method also provides a metric to calculate how strong the communities are connected and therefore, in how many groups the graph should be divided. Their algorithms works as follows:

(1) Calculate the *edge betweennes* for all edges in the graph. The edge betweennes can be defined in multiple ways, which is discussed in the paper, however, we will use betweennes as the fraction of the number of shortest paths and the number of all paths through an edge. (2) Remove the edge with the highest betweennes. (3) Recalculate the betweennes for all remaining edges and the quality Q of the current separation. (4) Start from step (2) until no edge remains.

The maximum of Q can be considered as being the best division for the network. The authors define this quality metric called *modularity* as follows: Let $E \in \mathbb{R}^{kxk}$ be a symmetric matrix for k communities in the network, where e_{ij} represents one community with the value being calculated by the fraction of all edges connecting community i with community j. For this calculation all edges of the original network are considered, even if they were already removed. The *Trace* $Tr(e) = \sum_i e_{ii}$ represents the fraction of connections inside a community. This leads to a definition of

$$Q = \sum_i (e_{ii} - (\sum_j e_{ij})^2). \tag{6}$$

This strategy shows good results in test graphs, however, since the betweennes and the quality needs to be calculated in each step, this algorithm performs with considerable high computational complexity for large data sets, such as in OSNs. An advanced algorithm based on this first draft is the so called ’Louvain’ method [19], which the authors tested on a web graph with 118 million nodes and more than two billion edges. Its complexity is analysed in [7] in more detail.

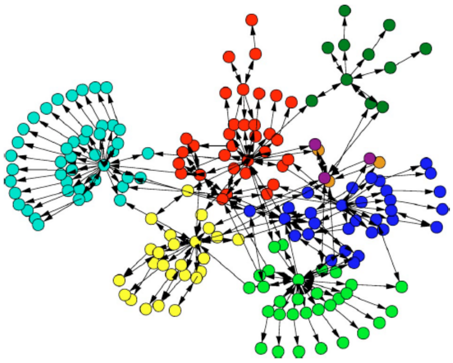


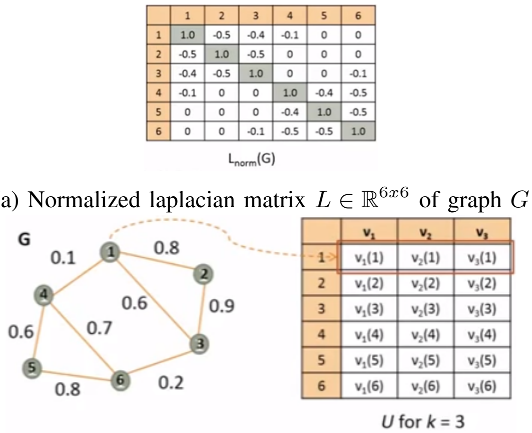
Fig. 8: Optimal division into communities of a website hyperlink graph, found by the Newman and Girvan algorithm using the shortest path betweennes [18].

Although the number of expected communities cannot be specified in general, it is sometimes useful to divide a graph into a fix number of clusters. The *spectral clustering* method is a popular strategy to obtain the best result for this problem. Ng et al. [20] present an algorithm that extracts k clusters from a graph. This is the basic idea:
(1) Consider the affinity matrix ($\hat{=}$ distance matrix) $A \in \mathbb{R}^{nxn}$ for a network with n nodes with a_{ij} being the weight of the edge (i, j) and a diagonal matrix D with $d_{ij} = \sum_{\{j|(i,j)\in E\}} w_{ij}$. (2) We then construct the normalized laplacian matrix $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. The k greatest (different) eigenvectors x of L, belonging to the k greatest eigenvalues, now inherit the information about the k best clusters. (3) Let $X = [x_1 x_2 ... x_k] \in \mathbb{R}^{nxk}$ be the matrix by stacking the eigenvectors. Form the new matrix

$$Y = \frac{X_{ij}}{\sqrt{\sum_j X_{ij}^2}} \tag{7}$$

by normalizing each of X’s rows. (4) Use for example the k-means algorithm to build k clusters by treating every row of Y as a point in $\mathbb{R}^k$. This step makes it much easier to separate the points. (5) Now add each node v_i to cluster j if and only if row i of the matrix corresponds to this cluster.

This algorithm shows satisfying results and has the great advantage that it requires no information on the cluster’s structure. But because of the high computational costs, and the problematic scaling for large networks, scientists have to be very careful on how to use this method.



	1	2	3	4	5	6
1	1.0	-0.5	-0.4	-0.1	0	0
2	-0.5	1.0	-0.5	0	0	0
3	-0.4	-0.5	1.0	0	0	-0.1
4	-0.1	0	0	1.0	-0.4	-0.5
5	0	0	0	-0.4	1.0	-0.5
6	0	0	-0.1	-0.5	-0.5	1.0

$L_{norm}(G)$

(a) Normalized laplacian matrix $L \in \mathbb{R}^{6x6}$ of graph G

	v_1	v_2	v_3
1	$v_1(1)$	$v_2(1)$	$v_3(1)$
2	$v_1(2)$	$v_2(2)$	$v_3(2)$
3	$v_1(3)$	$v_2(3)$	$v_3(3)$
4	$v_1(4)$	$v_2(4)$	$v_3(4)$
5	$v_1(5)$	$v_2(5)$	$v_3(5)$
6	$v_1(6)$	$v_2(6)$	$v_3(6)$

U for $k = 3$

(b) $k = 3$ eigenvectors of L in matrix U. $v_i(l)$ represents how strong node l belongs to cluster i. Ideally, the value has a maximum at one cluster with significant difference to other columns.

Fig. 9: Example for the spectral clustering with 6 nodes [21].

A case we did not consider yet is when communities overlap so that one node may belong to multiple communities. In [22], Galbrun et al. present a new way of identifying overlapping communities in so called *labelled* graphs. A graph G is then defined as $G = (V, E, l)$ with $l : V \mapsto 2^L$ assigning every node a set of labels, i.e. informations about that individual. Their algorithm then identifies k communities with maximum edge density. The authors mapped the *density maximization* problem into a *generalized maximum coverage* problem and proposed three different algorithms, based on a basic greedy scheme. Their *SPECTRAL* algorithms was then able to build 1000 communities in under 13 hours for a data set with nearly 1 million vertices and 3.5 million edges.

A big advantage of their method is the additional information for each community, which indicates why each node happens to be in this specific community. Additional information about each individual and the number of communities may not be available for most OSN graphs, though.

IV. Application Areas

In this chapter we will look at two application areas where data mining is used to solve current problems.

A. Intention Mining

A quite popular field of interest is how people feel about recent events. In [23] the authors conducted a survey on how the Croatian Twitter community felt about the government elections in 2015. Their approach was to find the ’right’ Twitter messages by searching for specific hashtags linked with the election and try to predict their mood based on the text. They created a word cloud of hashtags to see what users mostly talked about and counted how often politicians where named by the Twitter user reference system, hence their popularity. According to the authors, “analyzing Twitter data, for political or any other purpose, certainly is not a costly and time consuming activity", however, they did admit that predicting the result of an election may be not so easy, because of the high number of factor that can influence people. For further research it may be interesting to see how people in social media react to specific election campaign tactics in real time, to see what works and what does not.

Bollen et al. [24] investigated how the mood of Twitter feeds are correlated with the stock market, e.i. the Dow Jones Industrial Average (DJIA) value. For their analysis they used the *OpinionFinder* tool that can distinguish between positive and negative texts, and *Google-Profile*, which measures mood in terms of Calm, Alert, Sure, Vital, Kind and Happy. Their results showed, that their measure can increase predictions about the daily ups and downs of DJIA significantly, leading to a total accuracy of 87.6%.

Finally, we take a look at [25], where Wang et al. propose an algorithm to find *Sentiment Communities*, which are communities that also share the same opinion. They defined graphs of *social networking sites (SNS)* as undirected graphs $G = (V, E, S)$ where s_i is a measure for node i's sentiment popularity. Referring to the in section III-B mentioned *modularity* of Girvan and Newman, the authors added the "objective of maximizing the sentiment consistency" into the optimisation model. Therefore, they transformed the modularity optimisation problem into an *Semidefinite Programming (SDP)* problem where (1) the modularity is maximized and (2) the sentiment difference within a community is minimized. Their tests show that their method provides an effective algorithm to identify such communities with high accuracy.

When thinking about the influence maximization problem from section III-A, informations about sentiment communities can be very helpful to (1) choose the best initial start nodes and (2) choose the right advertisements that are most effective for this group.

B. Event Detection

Whenever a significant event happens in the world, there are very likely to be people that talk about it on social media. Thus OSN analysis can be used to detect upcoming events and predict their popularity. Zhang et al. [26] present a novel method to detect so called *burst events*, such as 'Hurricane Sandy hitting New York' or 'Beijing rainstorm in 2012', and estimate their popularity in the future, e.i. their significance. [Fig. 10] shows a flow chart of their approach.

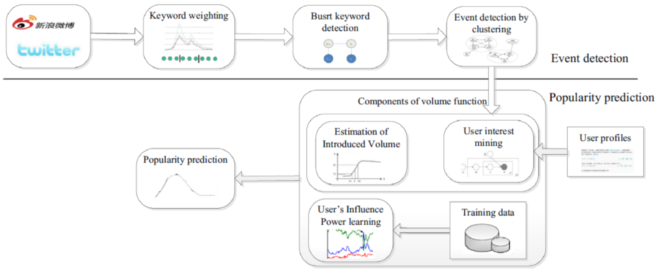


Fig. 10: Flow chart of the approach in [26]. First, the Twitter data is extracted, then events are found using clustering and finally the popularity of the event is measured, taking in account each user's influence

A burst event is represented by a set of *burst words* (words that are frequently used in this context). Therefore, the weight $w_{j,v}$ of the v^{th} word in the micro-blog mc_j is defined as:

$$w_{j,v} = (0.5 + 0.5 * \frac{tf_{j,v}}{tf_j^{max}}) * au(user(mc_j)) \quad (8)$$

$tf_{j,v}$ denotes the *term frequency* of the v^{th} word in mc_j and $au(user(mc_j))$ the *authority* of the author of mc_j. The authority of a user u_i is defined as

$$au(u_i) = \alpha + (1 - \alpha) * \sum_{u_j \in follower(u_i)} \frac{au(u_j)}{|follower(u_i)|} \quad (9)$$

The higher the sum of the weights of micro-blogs in a time interval t (*local weight*) the more likely a word is a burst word. After that, the burst words are clustered and transferred into a directed burst word relation graph. Burst events can then be found by searching for strongly connected subgraphs (see [Fig. 11]).

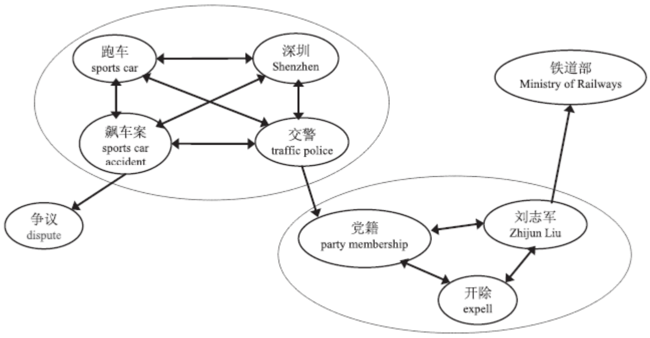


Fig. 11: example for the burst word relation graph [26]

In order to estimate an events popularity (*volume*), the authors present a spread model that shows how an event will be shared among the network in the near future. The current volume at t is represented by the sum of volumes expected to be produced by the 'infected' users, considering the users' influence, their interest in that event and the event's historical popularity.

In [27] six different event detection algorithms are compared. The authors tested these methods on Twitter data streams and automatically generated topics referring to real world events. Most of the algorithms showed reliable results in event detection, however, the tested events were quite significant and it would be interesting to test the methods' ability to detect minor events.

Because we are talking about OSNs it may also be that news or events occurring in the media do not reflect the truth. Shu et al. [28] conducted a survey addressing the detection of *Fake News*. Therefore, they classify fake news and try to detect them by analysing the sources creditability and the objectives truthfulness. They also looked upon several algorithms and related areas, such as *Rumor Classification* or *Clickbait Discovery*. It is stated, that detecting such false information is very important since it disturbs the increasing amount of people using OSNs to inform themselves. Moreover, when using automatic event detection more frequently in the future it is vital to protect the data sets from untrusted sources.

V. Summary and Prospects

This survey gave a brief overlook at how data mining and network analysis is applied on OSNs such as Twitter, Flickr and Facebook. We looked at graph properties and saw that OSN are often small world and scale free networks. We explained that many algorithms can not effectively perform on real social media data sets due to the magnitude of the collected data, thus it is often useful to apply graph sampling methods. In this context we talked about the *Snowball* sampling method and a so called *pre-coarsening* strategy.

We then presented multiple algorithms that are used to solve standard data mining problems, such as community detection and information spreading. For this purpose we analysed new approaches like Muric at al.'s algorithm to identify a networks most influential nodes by treating the graph as an LTI system. We also considered older strategies like Newman and Grivan's community detection algorithm to lay the foundation for newer approaches.

Finally, two application areas were analysed, where these techniques showed to be quite helpful. We saw that we can use community detection algorithms to identify *sentiment communities* and how that could be useful to maximize the influence of new marketing campaigns. We discovered that our society can significantly benefit from event detection algorithms, however, future scientist need to be very careful when analysing user posts because of the threat of *Fake News*.

In upcoming research it would be interesting to see how problems such as finding the initial start nodes for information spreading are addressed, for example by finding sentiment communities. Additionally, due to the increasing amount of Big Data in OSNs, old algorithms need to be adjusted to work on large data sets. Therefore, not only the time complexity needs to be reduced but also the usage of memory space.

References

[1] Instagram LLC, "Our story," 2018. [Online]. Available: https://instagram-press.com/our-story/ [Accessed: 2018-05-01]

[2] Twitter Inc, "Q1_2018_shareholder_letter," 2018. [Online]. Available: https://investor.twitterinc.com/results.cfm [Accessed: 2018-05-01]

[3] D. F. Nettleton, "Data mining of social networks represented as graphs," *Computer Science Review*, vol. 7, pp. 1–34, 2013.

[4] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 29–42. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298311 [Accessed: 2018-06-08]

[5] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[6] "Webgraphviz." [Online]. Available: http://webgraphviz.com/ [Accessed: 10.05.2018]

[7] Papadopoulos, Symeon and Kompatsiaris, Yiannis and Vakali, Athena and Spyridonos, Ploutarchos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[8] J. Travers and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, no. 4, pp. 425–443, 1969.

[9] Q. K. Telesford, K. E. Joyce, S. Hayasaka, J. H. Burdette, and P. J. Laurienti, "The ubiquity of small-world networks," *Brain connectivity*, vol. 1, no. 5, pp. 367–375, 2011.

[10] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, p. 10, 2018.

[11] Batrinca, Bogdan and Treleaven, Philip C., "Social media analytics: a survey of techniques, tools and platforms," *AI & SOCIETY*, vol. 30, no. 1, pp. 89–116, 2015.

[12] Leo A. Goodman, "Snowball sampling," *The Annals of Mathematical Statistics*, vol. 32, no. 1, pp. 148–170, 1961. [Online]. Available: http://www.jstor.org/stable/2237615 [Accessed: 2018-06-08]

[13] N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 2, p. 7, 2014.

[14] Kang, Ying and Yu, Bo and Wang, Weiping and Meng, Dan, "Spectral clustering for large-scale social networks via a pre-coarsening sampling based nyström method," in *Advances in Knowledge Discovery and Data Mining*, Cao, Tru and Lim, Ee-Peng and Zhou, Zhi-Hua and Ho, Tu-Bao and Cheung, David and Motoda, Hiroshi, Ed. Cham: Springer International Publishing, 2015, pp. 106–118.

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05. New York, NY, USA: ACM, 2005, pp. 177–187. [Online]. Available: http://doi.acm.org/10.1145/1081870.1081893 [Accessed: 2018-06-08]

[16] G. Murić, E. Jorswieck, and C. Scheunert, "Using lti dynamics to identify the influential nodes in a network," *PloS one*, vol. 11, no. 12, p. e0168514, 2016.

[17] D. Kempe, J. Kleinberg, and É. 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*, ser. KDD '03. New York, NY, USA: ACM, 2003, pp. 137–146. [Online]. Available: http://doi.acm.org/10.1145/956750.956769 [Accessed: 2018-06-08]

[18] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.026113 [Accessed: 2018-06-08]

[19] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008. [Online]. Available: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008 [Accessed: 2018-06-08]

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS'01. Cambridge, MA, USA: MIT Press, 2001, pp. 849–856. [Online]. Available: https://papers.nips.cc/paper/

2092-on-spectral-clustering-analysis-and-an-algorithm.pdf [Accessed: 2018-05-19]

[21] Omar Sobh, “Spectral clustering 01 - spectral clustering,” 2015. [Online]. Available: https://www.youtube.com/watch?v=zkgm0i77jQ8& [Accessed: 2018-06-08]

[22] Galbrun, Esther and Gionis, Aristides and Tatti, Nikolaj, “Overlapping community detection in labeled graphs,” *Data Mining and Knowledge Discovery*, vol. 28, no. 5, pp. 1586–1610, 2014.

[23] J. Ševa, B. Okreša Ðurić, and M. Schatten, “Visualizing public opinion in croatia based on available social network content,” *European Quarterly of Political Attitudes and Mentalities*, vol. 5, no. 1, pp. 22–35, 2016. [Online]. Available: https://doaj.org/article/8da84edb2c304238a79f164e150bbfb2 [Accessed: 2018-06-08]

[24] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[25] Wang, Dong and Li, Jiexun and Xu, Kaiquan and Wu, Yizhen, “Sentiment community detection: exploring sentiments and relationships in social networks,” *Electronic Commerce Research*, vol. 17, no. 1, pp. 103–132, 2017.

[26] Xiaoming Zhang, Xiaoming Chen, Yan Chen, Senzhang Wang, Zhoujun Li, and Jiali Xia, “Event detection and popularity prediction in microblogging,” *Neurocomputing*, vol. 149, pp. 1469–1480, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231214010893 [Accessed: 2018-06-08]

[27] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, and A. Jaimes, “Sensing trending topics in twitter,” *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1268–1282, 2013.

[28] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake news detection on social media,” *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.