An analysis of time complexity and a discussion on interpretability for two methods of constructing social network graphs

Žan Jonke
zj0527@student.uni-lj.si
Faculty of Computer and
Information Science,
University of Ljubljana
Večna pot 113
1000 Ljubljana, Slovenia
Munich Innovation Labs GmbH
Pettenkoferstr. 24
80336 Munich, Germany

ABSTRACT

Gathering useful information from user interactions on social media is a challenging task but has several important use cases. For example, law enforcement agencies monitor social media for threats to national security, marketers use them for launching marketing campaigns, etc. Since most social media platforms do not provide a standardized way of monitoring their data, most analyses are carried out manually. We aim to expedite this process by constructing social network graphs, where analysts can visually determine what users and contents are important. In this paper we compare two different approaches for constructing such graphs (path-weighted and degree-weighted). We analyze the time complexity of graph construction and discuss the usefulness of their visualization. In order to empirically evaluate both approaches, a method was developed, which stochastically generates data adhering to rules that govern the generation of data on a social media platform. We found that constructing degree-weighted graphs is faster, although the visualization of a path-weighted graph can answer more questions about the dataset.

KEYWORDS

Social media, network analysis, graph construction complexity

ACM Reference Format:
Žan Jonke. 2022. An analysis of time complexity and a discussion on interpretability for two methods of constructing social network graphs. In *Proceedings of Student Computing Research Symposium (SCORES'22).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

SCORES'22, October 6, 2022, Ljubljana, Slovenia

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Social media platforms have grown in popularity over the last two decades, manifesting several new ways of how humans interact with one another. The central idea is that users post contents and other users interact with them. For example, on Facebook [9] users post photos, write texts, create events etc. and other users like, comment or re-post the contents.

Law enforcement agencies monitor social media platforms for extremist groups, which might use them for spreading misinformation, incitement of violence or other forms of threat to national security [8, 14].

The application of social network analysis in marketing can provide marketers with valuable insights for developing communication and branding strategies by building up social capital in social networking sites [1, 5].

How data is collected from such platforms and what information can be extracted is of great utility. Being able to visually consider a network and to investigate it manually can be of great importance to analysts. In order for such a visualization to be beneficial, appropriate parts of the network graph (i.e. important users and content) must visually stand out. Our aim is to provide two ways of doing so and comparing them to one another from a technical perspective i.e. analyzing how much time is needed to construct such graphs and comparing them in terms of interpretability of their visualization.

2 RELATED WORK

Social network graphs can be constructed based on direct or inferred relations, including re-posting, replying or mentioning, through the shared use of hashtags or URLs, reciprocation or minimum levels of interaction activity, or friend/follower connections [4]. Karunasekera et al. [12] constructed networks of Twitter [10] accounts based on re-posts and mentions to discover communities active during the 2017 German election, valuing mentions and re-posts equally. URL sharing behaviour is often studied in the detection and classification of spam and political campaigns [2, 7, 17].

Edwards et al. [6] evaluated several different approaches of extracting social network graphs from datasets, which included linking two people if they were detected at the same event. Nasim et al. [13] introduce an approach of how to detect content polluting bots on Twitter. Their approach was to construct a two-mode user-event network linking two users if they had posted contents on the same day. Nguyen Vo et al. [16] constructed social network graphs which helped them evaluate an algorithm for revealing and detecting malicious re-posting groups. In their approach they considered only re-posts between users and for each pair determined re-post similarity and connected them if it was high enough.

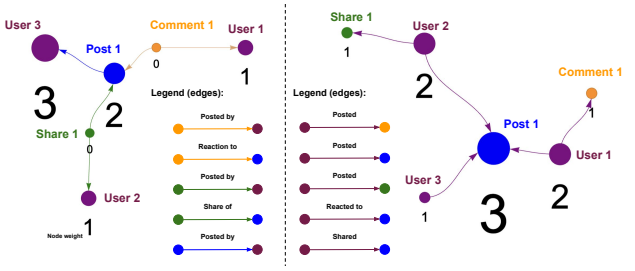

Figure 1: Example of a path-weighted (left) and degree-weighed graph (right) on the same dataset.

3 METHODS

3.1 Constructing graphs

In this section we propose two different methods of constructing discrete graphs given a dataset which can be obtained from a social media platform. The dataset contains entities called users and the content that they generated. Contents can also be reactions to one another i.e. a comment or a share.

Both methods have nodes of classes "user", "content", "comment" and "share". The methods differ in the way how edges are formed between the nodes, in which direction they are oriented and in the way the node weight is calculated.

The first method calculates node weights based on their degree (degree-weighted). We present this method in graph theoretic terms as follows:

Let $G_d(V, E)$ be a directed (degree-weighted) graph. Let $U \subset V$ be the set of users, $C \subset V$ the set of contents, $C_c \subset V$ the set of comments and $C_s \subset V$ the set of shares. Let $u \in U$, $c \in C$, $c_c \in C_c$ and $c_s \in C_s$. Edges $\{(u, c), (u, c_c), (u, c_s)\} \subset E$ only if u is the author of c, c_c or c_s. In this method there are no edges linking directly comments to contents or shares. Such a relation is represented as two edges $\{(u, c), (u, c_c)\} \subset E$ where u is the author of c_c (the same holds for c_s should c_c be a comment on a share). Shares are modelled in the same manner. The weights of edges are equal to 1, however the weights of nodes are equal to the node degrees.

The second method is based on calculating the node weight based on the number of simple paths that lead to that node (path-weighted). We present this method in graph theoretic terms as follows:

Let $G_p(V, E)$ be a directed (path-weighted) graph. Sets and nodes are defined as above. Edges $\{(c, u), (c_c, u), (c_s, u)\} \subset E$ only if u is the author of c, c_c or c_s and $\{(c_c, c), (c_c, c_s), (c_c, c_c), (c_s, c), (c_s, c_s)\} \subset E$ only if c_c is a comment to c, c_s or c_c or if c_s is share of c or c_s. The weights of all edges are equal to 1, but the weights of all nodes are equal to the number of directed simple paths ending in that node.

Cycles in the network would prevent the calculation of node weights from converging. The proposed connection rules guarantee that no cycles exist in the resulting networks. This can be easily seen in the following way: a node in a cycle must have both incoming and outgoing edges. As such U nodes can not be part of a cycle (they only have incoming edges). C nodes may have both incoming (from comments or shares) and outgoing (to the authors) edges. In order for a cycle to be formed there would need to be a directed edge from an author of the content to its comment, this is however not possible since such an edge is not defined. The same proof can be applied to both C_c and C_s nodes.

The degree-weighted graph can be easily understood and offers quick means for analysis, while the path-weighted graph is a bit more complex, albeit offers more in-depth information. In figure 1 we illustrate examples for both methods on the same dataset.

3.2 Generating random data

We want to empirically evaluate the time complexity of construction of both graphs and as such need many datasets with different numbers of contents and comments. Datasets from social media platforms varying in size might however be governed by different social dynamics, which have an effect on the network topology. Sampling from those datasets hence carries the risk of introducing an uncontrolled selection bias. Therefore we propose a way to artificially generate data (only systematically biased by the assumptions being made) in a stochastic manner governed by two parameters:

- the number of contents generated, denoted as N
- the probability that the new content generated is a comment, denoted as p_c

Our approach makes the assumptions that very little content on social media gains very high attention and that most users either post content or react to it but rarely both. To reflect this, each random content node that got generated got assigned a weight, which was sampled from a Pareto distribution [3], which is a power-law probability distribution that is used in description of social, quality control, scientific, geophysical, actuarial, and many other types of observable phenomena. This weight was used for weighed sampling, when assigning user reactions i.e. comment and shares to contents. Each user also got assigned a weight upon creation also sampled from a Pareto distribution, which got used when sampling users for authors of contents. A second weight is generated, which is equal to the inverse of the previous one and is used when sampling users for authors of comments. Using an inverse and weighed sampling manifests unsymmetrical behaviour of users in regards to their posting habits of contents and comments. We observed that most users who post high impact content are more likely to do so more often than others. To reflect this, the weights of all contents got multiplied by the weight of its author (i.e. the user). We also wanted to capture the observation that new users are more likely to enter a discourse, when a popular post was made. As such, when a new random content got generated with a high enough weight, the

probability of a new user spawning in their respective pool, got increased and linearly decayed with each new content generated. We also assumed that content shares are more similar to content than they are to comments and therefore introduced a transformation from content to share, which was dependent only on an individual user and their probability to post a share. For each share, a content was sampled using their respective weights. To take into account nested comments, these can also be sampled when assigning user reactions, however we observed these are not as frequent and as such their weights are sampled from the uniform distribution.

4 RESULTS

4.1 Time complexity of construction

We assume a dictionary representation of a discrete graph. Computing a degree-weighted graph takes $O(|V|)$ time. This can be achieved by iterating over all contents, comments and shares and adding corresponding nodes and edges. Calculating node weights takes $O(|V|)$ time because all that is needed is looping over all nodes and calculating the node degree, which can be done in constant time.

Constructing a path-weighted graph as described above takes $O(|V|)$ time since the same concept of looping over all contents, comments and shares can be used. Now we consider calculating node weights. Calculating one simple directed path between two nodes using depth first search takes $O(|V| + |E|)$ time [15]. Naively doing so for all nodes and all possible paths results in a combinatorial explosion in terms of time complexity. This analysis assumes that lengths of paths are comparable to $|V|$. However in our case this assumption can not be made since long paths require nested comments or shares i.e. comments to comments or shares of shares. For simplicity we exclude such paths from our analysis since we observed those are usually not found very frequently in interactions on social media. We estimate that calculating all simple paths for all nodes to take $O(|V| + |A|)$ time, where A is a list of all ancestors of all nodes. We assume that calculating all ancestors of a node to be constant since we are not considering nested comments or shares. The following observations can be made:

- $|A| \approx |V| + |E|$ since every edge roughly introduces one new ancestor,
- a node might have a large number of ancestors, however each ancestor has at most two outgoing edges and is at most at a distance 2 from the source, meaning that depth first search finds all simple paths between two nodes in constant time.

When $|V|$ gets large enough, most of the time needed for the computation of a path-weighted graph is used for calculating node weights. Therefore calculating a path-weighted graph takes $O(|V|+|E|)$ time.

An empirical evaluation of time complexity can be seen in figure 2. Datasets of different sizes were constructed with N ranging from 0 to 45000 with a step of 500. At every step we generated 30 random datasets using the above described method and measured time of construction for both approaches.

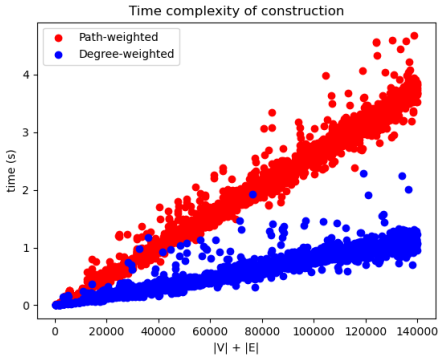

Figure 2: Time complexities of graph construction for both methods. Both depend linearly on $|V| + |E|$.

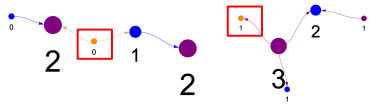

Figure 3: Comparison of interpretability of the path-weighted (left) and degree-weighted (right) graph. It is not possible to determine to what content does the comment belong to in the degree-weighed graph.

4.2 Interpretability

Here we demonstrate a way how one could interpret the node weights and degrees of both types of graphs. In a degree-weighted graph a content's weight reflects the sum of all users that reacted to it, with either a comment or a share. The same interpretation can be made for both comment and share nodes. A user's weight reflects the sum of all their activities. Conversely, in a path-weighted graph a content's weight reflects the sum of all comments and shares written as a reaction to it, hence its "impact". As before, the same interpretation can be made for both comments and shares. A user's weight reflects the sum of the contents written by the user (the "activity") plus the sum of the their impacts (total impact), hence their (relative) importance. Additionally the degrees of nodes in a path-weighted graph can be interpreted as weights of degree-weighted graphs.

As an edge is missing from either a comment or a share to the content in a degree-weighted graph, this results in an ambiguity. Consider figure 3. It shows two visualizations of the same data. For the degree-weighted graph it can not be determined by visualization alone where the comment belongs to, however this is not the case for the path-weighted graph.

4.3 Visualization

Here we show some visualizations of networks using the random data generating model described above. We generated two datasets with values of N equal to 1000 and 2000 and p_c equal to 0.8. In figures 4 and 5 two visualizations of path-weighted and degree-weighted graphs are shown. The visualizations were done using a Javascript library called vis.js where graph physics enable for a

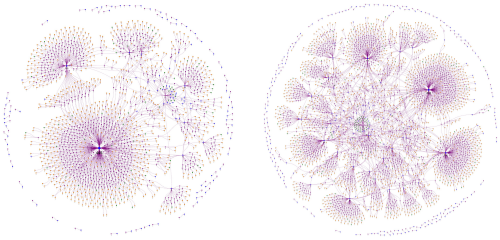

Figure 4: Visualizations of degree-weighted graphs with N equal to 1000 (left) and 2000 (right).

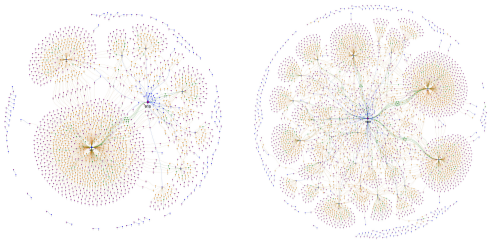

Figure 5: Visualizations of path-weighted graphs with N equal to 1000 (left) and 2000 (right). The same datasets were used as in figure 4.

more flexible visualization. The physics solver ForceAtlas2 was used [11]. The tool allows for zooming and makes even larger networks still manageable to analyze manually. However, these visualizations take a lot of computing power and it becomes very time consuming to render networks which contain more than 5000 nodes. This issue can be mitigated by using alternative libraries with GPU support.

5 DISCUSSION

We have presented and analyzed two different approaches (path-weighted and degree-weighted) for constructing directed graphs from data modelling social media networks. We have shown that in terms of time complexity the degree-weighted graphs are favourable, since they require less time to construct, although the difference only becomes noticeable when the dataset is very large.

The advantages of degree-weighted graphs are that they are fast to construct and their visualizations are easy to interpret regarding questions about the most popular posts and which users post most frequently. The corresponding nodes will have a high weight and therefore visually stand out. A disadvantage of this method is that further analysis is harder to conduct i.e. it is harder to answer questions about the most important users, the impact of individual posts and about the users most people interact with.

The advantage of path-weighted graphs is for analysts to be able to answer the above questions more easily since all relations are unambiguously reflected by graph edges and more meaningful weights are given to content and user nodes. However, the disadvantage of this method is that network graph construction takes a longer time.

Analysing our model for generating random social media data, we find that it lacks modelling of phenomena where users are more likely to react with only a small pool of users (their friends and family), rather than with users who post the most popular content.

As a next step we plan to test the validity of our findings on real social media data and analyse the time complexities of adding new nodes to an existing graph in a realistic monitoring scenario.

ACKNOWLEDGMENTS

We would like to express our very great appreciation to Mathias Uhlenbrock and Oussama Jarrousse for their valuable and constructive suggestions during the planning and development of this research project.

REFERENCES

[1] Ioannis Antoniadis and Anna Charmantzi. 2016. Social network analysis and social capital in marketing: theory and practical implementation. *International Journal of Technology Marketing* 11 (01 2016), 344. https://doi.org/10.1504/IJTMKT.2016.077387

[2] Cheng Cao, James Caverlee, Kyumin Lee, Hancheng Ge, and Jinwook Chung. 2015. Organic or Organized? Exploring URL Sharing Behavior. (2015), 513–522. https://doi.org/10.1145/2806416.2806572

[3] Henry T. Davis and Michael L. Feldstein. 1979. The Generalized Pareto Law as a Model for Progressively Censored Survival Data. *Biometrika* 66, 2 (2022/09/27/1979), 299–306. https://doi.org/10.2307/2335662

[4] Lewis Mitchell Derek Weber, Mehwish Nasim and Lucia Falzon. 2021. Exploring the effect of streamed social media data variations on social network analysis. *CoRR* abs/2103.03424 (2021). arXiv:2103.03424 https://arxiv.org/abs/2103.03424

[5] Shaun Doyle. 2007. The role of social networks in marketing. *Journal of Database Marketing Customer Strategy Management* 15, 1 (2007), 60–64. https://doi.org/10.1057/palgrave.dbm.3250070

[6] Michelle Edwards, Jonathan Tuke, Matthew Roughan, and Lewis Mitchell. 2020. The one comparing narrative social network extraction techniques. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 905–913. https://doi.org/10.1109/ASONAM49781.2020.9381346

[7] Fabio Giglietto, Nicola Righetti, Luca Rossi, and Giada Marino. 2020. It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 Italian elections. *Information, Communication & Society* 23, 6 (2020), 867–891. https://doi.org/10.1080/1369118X.2020.1739732 arXiv:https://doi.org/10.1080/1369118X.2020.1739732

[8] John S. Hollywood, Michael J. D. Vermeer, Dulani Woods, Sean E. Goodison, and Brian A. Jackson. 2018. *Using Social Media and Social Network Analysis in Law Enforcement: Creating a Research Agenda, Including Business Cases, Protections, and Technology Needs.* RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/RR2301

[9] Meta Platforms Inc. 2004. Facebook. https://www.facebook.com. Accessed: 2022-07-29.

[10] Twitter Inc. 2006. Twitter. https://www.twitter.com. Accessed: 2022-07-29.

[11] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. 2014. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLOS ONE* 9, 6 (06 2014), 1–12. https://doi.org/10.1371/journal.pone.0098679

[12] Fred Morstatter, Yunqiu Shao, Aram Galstyan, and Shanika Karunasekera. 2018. From Alt-Right to Alt-Rechts: Twitter Analysis of the 2017 German Federal Election. (04 2018), 621–628. https://doi.org/10.1145/3184558.3188733

[13] Mehwish Nasim, Andrew Nguyen, Nick Lothian, Robert Cope, and Lewis Mitchell. 2018. Real-Time Detection of Content Polluters in Partially Observable Twitter Networks. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1331–1339. https://doi.org/10.1145/3184558.3191574

[14] International Association of Crime Analysts. 2018. Social Network Analysis for Law Enforcement [White paper]. (2018).

[15] Robert Sedgewick. 2001. *Algorithms in C, Part 5: Graph Algorithms, Third Edition* (third ed.). Addison-Wesley Professional.

[16] Nguyen Vo, Kyumin Lee, Cheng Cao, Thanh Tran, and Hongkyu Choi. 2017. Revealing and Detecting Malicious Retweeter Groups *(ASONAM '17)*. Association for Computing Machinery, New York, NY, USA, 363–368. https://doi.org/10.1145/3110025.3110068

[17] Tingmin Wu, Sheng Wen, Yang Xiang, and Wanlei Zhou. 2018. Twitter spam detection: Survey of new approaches and comparative study. *Computers Security* 76 (2018), 265–284. https://doi.org/10.1016/j.cose.2017.11.013