NAME: MOHAMMAD REZA ARDESTANI DATE: NOV 20, 2022

This Is Damn Slick! Estimating the Impact of Tweets on Open-Source Project Popularity & New Contributors

Hongbo Fang, Hemank Lamba, James Herbsleb, Bogdan Vasilescu

PROBLEM(S) ADDRESSED

This research is conducted as to whether Twitter can effectively help in promoting Open-Source Software (OSS) projects on GitHub, one of the widely used platforms for hosting OSS projects. For this purpose, three main questions are formulated. Firstly, how much can a tweet result in projects' popularity (measured by the number of new GitHub stars) and attracting new contributors (people who did not have any contribution in the project repository and started contributing shortly after being impacted by related tweets)? Secondly, what are the characteristics of impactful tweets? Thirdly, what are the attributes of new contributors?

MOTIVATION

Gaining sufficient adoption and contribution, which has remained a challenge, underpins OSS projects' sustainability and success. Previous papers mostly study in-network awareness-raising mechanisms, namely the relation between GitHub page and badges maintenance with GitHub audience engagements or being featured by the original hosting platform. We can, however, use social media platforms to surpass this limitation. Social media, especially Twitter, which is highly used by developers, allows us to reach a larger audience beyond our close connections with fewer barriers to participation and with high viewership. Although social media activity has been shown to be effective in promoting other contexts, like the increase of academic paper citations, the potential positive effect on OSS sustainability has not been explored yet.

PROPOSED SOLUTION

First, we discuss methods being used for each of the three questions in the following two paragraphs, then we analyze all five phases of the project pipeline in a fine-grained way.

Research question (RQ) 1 and RQ2 use the difference-in-difference (DID) method, a widely used technic in experimental research for measuring a causal effect. We might see an increase in repository stars and related tweets about that project, but we are only interested to count those cases where the increase of tweets caused an increase in stars, not the inverse case. We are also interested to exclude the effect of confounding factors, namely, in-person events or advertisements on other platforms. DID, by using a regression estimator, help us to accurately estimate the causal effect in this experiment that we cannot observe or control all confounding factors.

A mix of quantitative and qualitative methods is used to address RQ3. First, we manually appreciate the difference between newly joined developers and those who did not decide to contribute to the project while both were exposed to the project's tweet. Secondly, we numerically measure past twitter as well as GitHub activities and characteristics of newly attracted contributors.

Collecting a sample of nearly 70k cross-linked GitHub and Twitter users is the first step of the preprocessing phase in our pipeline. Further, we explore the user's tweets to extract all tweets containing GitHub artifacts to create a tweet dataset. Next, a set of filters are used to de-noise the dataset. We then use a six-month window time and discard all tweets outside this period. Now, we are left with 10k tweets containing almost 7k unique repositories. Finally, we use the name of collected repositories and collect other tweets, not tweeted by our initial cross-linked users, in the mentioned time window. A different heuristic for further cross-linking users is used on all users, however.

The second phase is aggregating tweets into bursts. It is impractical to measure the effect of single tweets while other similar tweets are posted closely after our single tweet. That is why we collect tweets about a specific repository into bursts that have an in-burst time gap less than X, a hyper-parameter, and an out-burst distance to other similar tweets more than Y time. For choosing the best value for the aforementioned hyper-parameters, sensitivity analysis is used.

The treatment group and control group, which are integral parts of experimental studies, are prepared in the third phase by the stacked DID design. Mentioning a particular GitHub repository in a burst is considered an intervention. This design helps us implicitly control confounding factors and computational efficiency. Propensity score matching, described later, is also used in sampling control groups.

In the fourth phase, to answer RQ1 and RQ2, we record various characteristics of the burst (time, mentioned repository, author, or # of likes), various characteristics of the repository, and the value of our two dependent variables (# of stars, # of contributors). Before estimation we do a set of standardization, e.g., removing outliers and log-transformation of skewed distribution. Then we use the following estimator regression function:

$$Outcome\ variable = treatment\ estimate * bool\ pre\_treatment * bool\ treatment * specific\ characteristics + same\_repository\ bias * bool\ pre\_treatment * same\ repository\ effect + initial\ difference * bool\ pre\_treatment + base - line\ change\ over\ time * bool\ treatment + time\_cohort\ bias + repository\_cohort\ bias.$$ More Formally:

$$O_{itc} = B_0 * p_{tc} * t_{ic} * X_{ic} + B_1 * p_{tc} * S_{ic} + B_2 * p_{tc} + B_3 * t_{ic} + \delta_{ic} + \alpha_{ic} \; ; \; t: time, \; c: cohort, \; i: i^{th} repository.$$

We use this regression for both RQ1 and RQ2 with this difference that is only used for RQ2.

In the last phase, we study new contributors plausibly attracted by tweets to answer RQ3. In a qualitative and iterative and thematic manner, we analyze the relation between new contributors to the author of the tweets and the repository mentioned in the tweet. We also distinguish between *likely attracted* and *attracted* developers. Next, we use three regressions to quantitatively answer RW3. One logistic regression is used to measure the association between developers' GitHub status to whether they become attracted after seeing tweets or not. Linear regression is used for estimating the association between short-term activity (30-day commit counts) to their past Twitter and GitHub interaction and collaboration. Finally, we use a Cox proportional hazards survival regression to measure the relation between past Twitter and GitHub interaction and collaboration with the risk of disengagement from the project.

Evaluation & Results

Regarding RQ1 and RQ2, tweeting about an OSS GitHub project is statistically significant for the increase of the number of stars (nearly 7% increase), its effect on attracting new committers is small (nearly 2%), however. Burst duration also has a positive effect on stars gained.

Concerning RQ3, we qualitatively found three themes across the sample; close tweet's author and new collaborator relation (32%), weak ties (32%), and new contributors who are interested and actively follow the author (36%). Estimated logistic regression shows that having previous GitHub collaboration, being new to the GitHub platform, and the ratio of original tweets over retweets have a positive correlation with being attracted. Moreover, previous GitHub commits have no effect, and being an active Twitter user has a negative effect. Our linear regression model shows both past GitHub collaboration and Twitter interaction are statistically significant for the number of 30-day commits of new contributors. Finally, our last regression model shows past GitHub collaboration lowers the risk of disengagement. Almost all results are provided with their standard error rate in tables to state their degree of correctness.

Other founding and implications are; getting featured by GitHub requires a project to be already popular unlike Twitter, more specific rather than generic tweets can be more attractive to new committers, repositories with a stronger community are more successful at attracting new committers, and Twitter can be closely integrated with code hosting platforms. Lastly, missing data problems and noisy cross-linked user collection might make the carefully conducted result less reproducible.

Reflections on Learning

- DID was indeed an interesting new method I learned. By knowing it we can no longer get tricked by common fallacies. Many people, for example, associate their recovery with the meditation that they did while they don't realize they might have got well from their illness because it simply had finished its natural period. DID is helpful in many social and experimental contexts.
- Propensity Score Matching was another interesting statistical method. I did not know we can artificially generate control groups with this technic.
- Cox Hazards Survival Regression; a tool for survival analysis. This method computes the effect of the hazard, also known as the risk of failure, considering that the participator has survived until a specific time.
- As depicted in the appendix, they use human evaluation to find the extent to which their heuristic (home page links are promotional and issue pages are for technical purposes) works. Manual annotation of an expert proved their heuristic to be 77% accurate. I learned to enlist the help of human evaluation in my work when I am setting hyper-parameters or using heuristics.

Research Ideas or Directions

- Damn Slick?! This pseudo-catchy phrase is neither cool nor relevant to the paper's content. Somebody should research why some people, in pursuit of attention, put fancy titles on their papers and do not notice this doesn't work.
- As the RQ2 results show, it is hard to attract new committers. One hypothesis to test is whether dividing the repository into different areas and trying to target Twitter users with similar expertise is significantly effective or not. By targeting certain experts with germane content we can help the audience easily and quickly find a place for the OSS project where they can be of immediate help thereby getting more appealed to contribute to the project.
- Another potential future work is crawling LinkedIn accounts for targeting experts to compare Twitter and LinkedIn in their effectiveness at advocating for the OSS projects.