Unbiased estimator for the number of GitHub active users using random subset sampling.

Unbiased estimator of the number of GitHub user

This paper will be discussed about random sampling and an unbiased estimator for estimating the number of GitHub active users.

Aukkawut, Ammartayakun

Worcester Polytechnic Institute, aammartayakun@wpi.edu

GitHub is a website for programmers, developers, and researchers in which their user-id generation is start with 1 and increase by 1 for each user created. However, if some users delete their accounts, the ids will be removed from the id space and not being used by new users. This paper will estimate the number of active users on GitHub with the modified version of random prefix sampling method suggested by *Zhou et al.* or random subset sampling. Then, evaluation on the estimator shows that for small subset size (l = 2500) on validation set of size (n = 500000) provides the overestimated result but with the subset size (l = 5000) on validation set shows that this is the good amount for estimating number of GitHub users.

CCS CONCEPTS • Online Information Services

Additional Keywords and Phrases: Sampling, Unbiased estimator, GitHub

1 INTRODUCTION

GitHub is an online Git provider which is widely being used by programmers, developers, and researchers as there are over 56 million users on GitHub in 2020 [2]. Although the uses of GitHub are not only limited to sharing the repository, but also for hosting websites and other services in which it would require a lot of resources to maintain and sustain the system. Moreover, the information from each user like how many repositories does they have or number of active users by time would help GitHub and other online Git provider to estimate their resources that are needed to put into the system and elevate users' experiences.

The problem is that there are a lot of users decided to delete their accounts or create multiple accounts in which it would be hard to estimate the number of active users on GitHub. Let say for example, we have user with id `1`and user with id `2`. Then, if user with id `2` decided to remove their account, the new user will be assigning with an id of `3` instead of reusing the old id. Hence, just look at the last id is not enough to estimate the number of users.

With GitHub API, or REST API, the sample of active users account from starting id `i` to id `j` can be done and that will lead to the estimation of active users across the entire user spaces on GitHub in which the detail will be discussed on the next section.

2 RANDOM SUBSETS SAMPLING WITH FIXED SIZE SUBSET

In this section, the properties of GitHub user-id space in which will leads to the method of using fixed-size subsets for random subsets sampling will be discussed.

2.1 GitHub user-id space

GitHub API id generation algorithm is the natural number running from the first id 1 to the latest id `n` with an increment of 1. The problem is that when users removed their accounts, those ids will not be able to use by new users. In another word, if users removed their accounts, those ids will be removed from the space.

2.2 The estimator and proof of unbiasedness

Let Ω be a user-id space with the size of |Ω|. According to [1], they suggest that dividing the space into the fixed size subset ω_i with the equal length l for $i \in 1,2,\dots,\left\lceil\frac{|\Omega|}{l}\right\rceil$ and sample $m < \left\lceil\frac{|\Omega|}{l}\right\rceil$ of subsets. Then, the probability of sample the particular ω_j is $p_l = \frac{1}{\left\lceil\frac{|\Omega|}{l}\right\rceil}$. Let say the random variable X_i refers to the number of active users in ω_i. Hence, the estimator is

$$\widehat{N} = \frac{\mathbb{E}[X]_{sample}}{p_l} = \frac{1}{mp_l}\sum_{i=1}^{m} X_i \quad (1)$$

Proof *Let say we randomly pick m subset in which contains id q inside it. Then, the number of valid ids $q_i \in \omega_i$ can be represent as random variable X_i in which $X_i \sim Binom(|\Omega|, p_l)$. It can be represented this way because it can be represented as the summation of indicator function of which are series of independent Bernoulli's trials.*

Hence,

$$\mathbb{E}[X_i] = |\Omega|p_l$$

But

$$\mathbb{E}[X]_{sample} = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} X_i\right] = |\Omega|p_l$$

Hence,

$$\widehat{N} = \frac{1}{mp_l}\sum_{i=1}^{m} X_i$$

and,

$$\mathbb{E}\left[\widehat{N}\right] = \mathbb{E}\left[\frac{1}{mp_l}\sum_{i=1}^{m} X_i\right] = |\Omega|$$

So, this is unbiased.

2.3 Methodology

Due to the limitation of GitHub API, the request output is capped to 100 per requests and the request call is capped to 5000 requests per hours. The ALGORITHM 1 was used to gather the validation set of size 500000 (start from id = 1 to id = 500000) and calculate the ground truth for the validation set. Practically, we do not want to have high sample size as it will take a lot of time to accomplish the task given the constraint. So, as tested, the sample of size 2500 would take over 30 hours to accomplish (on average, 33 minutes for 100 samples with the size of 2500) in which this does not included delay time caused by the API call limit and internet connection speed. Hence, only sample with size 2500 is the only test I did.

To calculate the non-constant term of estimation, the algorithm can be summarized as in ALGORITHM 1.

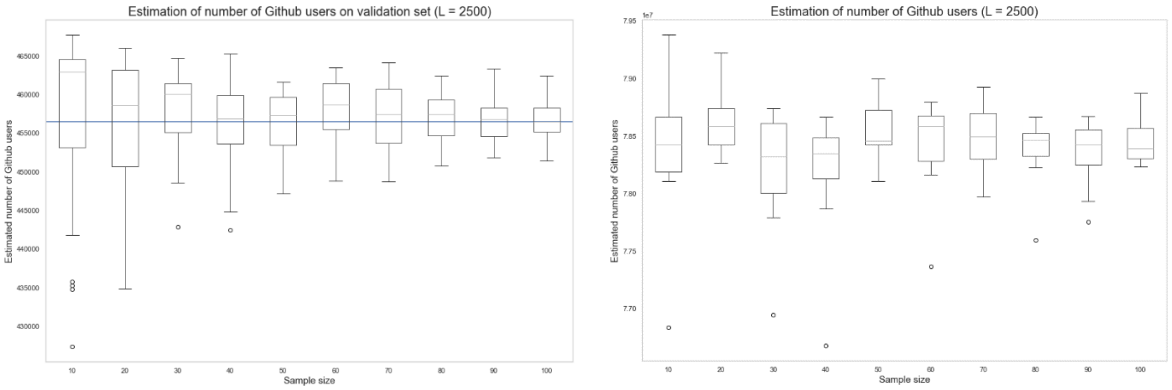
ALGORITHM 1: Calculate X_i

```
Define exhaustive_collection(int starting_id, int end_id):
    id_Array = []
    While True:
        response = call API for searching USERS with id = starting_id to starting_id + 100
        if response id < end_id:
                Append response id to id_Array
                End if
        Else
                break
                End else
    End while
    Return id_Array
End Define
Define calculate_sum_X(int m, int l):
        subset_id_Array = Random m integers from 1 to latest user id / l
        int i = 0
        For q in subset_id_Array:
                X[i] = length of exhaustive_collection((l-1)*q, l*q)
                i = i + 1
        End For
        Return sum(X)
End Define
```

3 RESULTS AND DISCUSSION

In this section, the correctness of the method will be discussed.

3.1 Number of Active GitHub users



a b

Figure 1: The boxplot in a) shows the estimation of number of GitHub users on validation set with subset size of 2500 with ground truth in blue line, and b) shows the estimation of number of GitHub users on actual data with subset size of 2500.

According to Figure 1a, the test has been performed on the validation set of size 500000 with ground truth of 456500. The boxplot shows that the estimator is acceptable (within interquartile range of all sample size). Although it seems to overestimate but with another experiment with subset size of 5000 (see Figure 2), the estimator seems to perform well. This shows that there might be some interplay between sample size and the estimation.

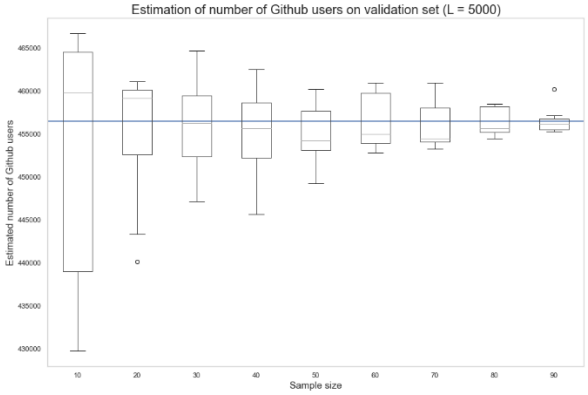


Figure 2: The boxplot shows the estimation of number of GitHub users on validation set with subset size of 5000 with ground truth in blue line.

According to Figure 1b, the plot suggests that number of active users on GitHub is around 78 million to 79 million accounts. Due to the likely to overestimate on low subset size, the result might not convincing. More testing might be ideal to evaluate this estimator and it would be interesting if different subset size and more experiments were considered. Lastly, the evaluation of estimator, this includes confidence interval and the proper effective theoretical sample size for both GitHub and other online services will be useful for the future work.

ACKNOWLEDGMENTS

This work is the part of DS 504 Big Data Analytics class at Worcester Polytechnic Institute and was helped by professor Yanhua Li's suggestions in class.

REFERENCES

[1] Jia Zhou, Yanhua Li, Vijay Kumar Adhikari, and Zhi-Li Zhang. 2011. Counting YouTube videos via random prefix sampling. In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference (IMC '11). Association for Computing Machinery, New York, NY, USA, 371–380. DOI:https://doi.org/10.1145/2068816.2068851

[2] The State of the Octoverse. *GitHub*, 2021. https://octoverse.github.com/.