Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 98-106
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

The Dynamic Prediction Model of Number of Participants in Software Crowd Sourcing Collaboration Development Project

Yu-Tang Zheng, Sun-Jen Huang*, Te-Hsin Peng

Department of Information Management, National Taiwan University of Science and Technology, Taiwan

Email: smilepigly@gmail.com, *huang@cs.ntust.edu.tw, sal55960310@gmail.com

How to cite this paper: Zheng, Y.-T., Huang, S.-J. and Peng, T.-H. (2018) The Dynamic Prediction Model of Number of Participants in Software Crowd Sourcing Collaboration Development Project. *Journal of Computer and Communications*, **6**, 98-106.
https://doi.org/10.4236/jcc.2018.612010

Received: November 2, 2018
Accepted: December 23, 2018
Published: December 26, 2018

Abstract

Many online platforms providing crowd with opportunities to participate in software development projects have been existed for a while. Meanwhile, many enterprises are using crowd source to collaboratively develop their software via these platforms in recent years. However, some software development projects in these platforms hardly attract users to join. Therefore, these project owners need a way to effectively predict the number of participants in their projects and accordingly well plan their software and project specifications, such as the program language and the size of the documentation, in order to attract more individuals to participant in the projects. Compared with the past prediction models, our proposed model can dynamically add the factors, such as number of participants in the initial stage of the project, within the project life cycle and make the adjustment to the prediction model. The proposed model was also verified by using cross validation method. The results show that: 1) The models with the factor "the number of user participation" is more accurate than the model without it. 2) The factors of crowd dimension are more influential on the prediction accuracy than those of software project and owner dimensions. It is suggested that the project owners not only just consider those factors of the software project dimension in the initial stage of the project life cycle but also those factors of crowd and interaction dimensions in the late stage to attract more participants in their projects.

Keywords

Prediction Model, Software Crowd Sourcing Collaboration Development, Open Source

1. Introduction

According to Kalliamvakou [1], nearly 33% of the collaborative platforms have no users involved in the development projects. Therefore, it's important for project owners to know whether their projects are attractive to users, and whether their specifications are developed for most users. In addition, if they can well predict the number of users who are interested in participating in their projects in advance, they can well plan their development activities.

In the past, scholars indicated that the higher the number of project's participants is, the higher probability of being collaboratively developed is [2]. Therefore, it is obvious that it is difficult to correctly predict the number of participants in the software development collaboration projects only based on those factors of software and project dimensions before the projects are put into the platform. This study considers that an ideal software project prediction system should be able to dynamically adjust the predict results based on the data within the software development life cycle.

Based on the above research motivation, we expect to propose a dynamic prediction model for the number of participants in the collaborative software development, and to explore the impact of the factors of crowd dimension on the degree of attention to the project.

2. Literature Review

2.1. Software Crowd Sourcing Collaboration Development

On the software crowd sourcing collaboration development platform, users can easily upload local software projects to the Internet, and download the projects interested to them and save them into their own project library for further participation on project. It provides the ability to easily develop software projects collaboratively, including allowing users to track other users, compose organizations, track the dynamics of software libraries, and modify software code, make comments, etc.

Mining software repositories abbreviated as MSR. It refers to the behavior of searching for software library or code data [3]. The research data uses the GHTorrent data set provided by MSR officially. The data source is accessed through the Github API into a new data set (**Figure 1**).

The definition of the software library refers to all the log files saved during the process of software evolution. The files include: the changes of Metadata (such as user and developer ID or time stamps), a record of the differences between versions (such as the change log. branches and tags between versions) and project bug tracking system.

There are 34 events in Github. In the software collaborative development process, whether the commit submitted by everyone can be adopted is decided by the project owner. Users can propose issue or ideas in the discussion area.

If users see a favorite project and want to contribute, he can copy to his local end repository. If the user wants the project owner to be merged, the user can pull

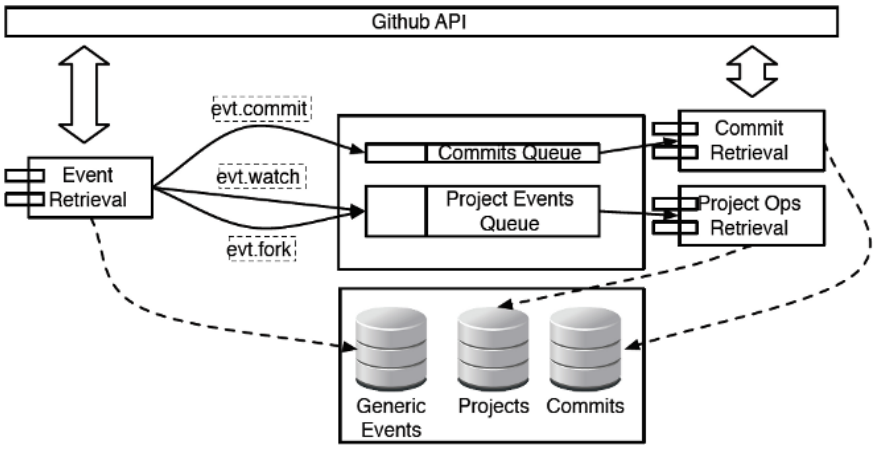

Figure 1. The system architecture of GHTorrent access to Github.

request. When the project is developed to a certain extent, the version can be released and become a more formal software product.

On the user side, a user can follow his/her favorite users and pay attention to the project development they created. The Github also provides users with the function to form an organization event without limiting the number of members.

2.2. Software Crowd Sourcing Collaboration Development Project Prediction

Many scholars use the API method to obtain the real data on the platform for further research. For the research on the number of people paying attention to the project, the main data currently provided on the Github platform include the number of users paying attention to the project and the number of users fork the project.

Some scholars believe that the quantity of attention represents the user's interest in the project [4] [5] [6]; some scholars believe that the number of fork represents the user's interest in the project and wants to contribute [5] [7]. However, some scholars have also found that there are too many copies bot without further participation action. Therefore, there are many different opinions on predicting whether a particular software collaborative development project is attractive to the crowd.

A popular software collaborative development projects in this research is defined as one which attracts a certain number of users on the platform. A project which is predicted as attractive to crowd means that this project will have a higher probability to put into action through collaborative development activities.

3. Research Method

3.1. Research Process

Stepwise Regression Procedure is adopted in the predictive model construction.

Using cross-validation, the data was divided into ten equal parts, 9/10 training data, and 1/10 test data which were used to verify the final model. The research process includes the six steps as follows.

Step 1: Grab the Github data through a third-party API and build a history database.

Step 2: Perform multi-layer grouping through the K-means algorithm until the group features are obvious.

Step 3: Include the influence factors using clustering result obtained in Step 2.

Step 4: Evaluate the impact of each factor on the number of participants among groups.

Step 5: Construct the predict models using the influence factors and clustering groups obtained from the above steps.

Step 6: Verify and further compare the prediction accuracy of two models using the MMRE and Pred (0.25) metrics.

3.2. The Impact Factors of the Number of Participants

On the Github, 34 attribute factors are provided and can be divided into three dimensions which are software project, owner and crowd. However, this research found that the project owner has an important degree of influence in the early stage of the project, so the software project dimension is divided into project owner dimension and software project dimension. The project will change with the development life cycle. Therefore, the research variables are divided into fixed factors and uncertain factors according to the time characteristics.

3.3. Dimension Design

We have defined the dimensions of factors affecting the number of participants in the software development collaboration project. The main dimensions include the following three ones:

- Software Project Dimension

The software project dimension is the basic information of the project and the changes in its development process. This study draws four factors as research variables including Fix_doc, Fix_language, Fix_developer and Dy_release.

- Project Owner Dimension

The project owner refers to the user who created the project. The feature is that the initial stage of the project has an impact on the growth of the number of participants. We set the research variables including Fix_type, Fix_follower and Fix_following.

- Crowd Dimension

When the software project is developed through the collaborative crowd development platform, users on the platform can join the project development at any time. Whether it affects the number of participants after the crowd participation is an important issue of this study. The factors we set for this dimension include Dy_commit, Dy_issue and Dy_fork.

4. Model Construction and Verification

4.1. Data Collection

The object of this study is the artificial intelligence software project on the Github. The collected data are the projects created from January 1, 2015 to January 31, 2016, and the development information for each project during the one-year period. The project filter conditions are provided by Github's artificial intelligence related labels, and the project is an originally pure software project. The project samples were filtered out the projects with zero attention, and the final sample dataset was 1096.

4.2. Multi-Layered Data Grouping

Data grouping is to classify similar things. Variables in the same group may have unequal differences. There're two types of data grouping. The first one is to use the number of participants increased each week to do classification; the second one is to monitor the growth trend of the number of weekly participants. The characteristics between the separated groups are the same. However, the first method is not suitable because the amplitude of weekly curve is too dramatic. In the end, the study adopts the second method, divided into five groups with obvious characteristics.

4.2.1. Grouping Method

Through data conversion, we first scale the growth trend of each project so that we can compare the relative trends and then calculate the difference between each data.

Data standardization:

$$Z = (x - \mu)/\sigma$$

σ is population's standard deviation; x is raw data to be normalized; μ is population's mean

Calculate the difference between vectors:

$$D_i = Z_{i+1} - Z_i$$

Clara algorithm can deal with the larger dataset. Internally, it is achieved by considering a fixed sample size subset so that time and storage requirements become linear at n instead of quadratic.

The RCl arasyn tax as below was used to group the data in this study:
clara(*x, k, metric* = "*Euclidean*", *stand* = FALSE, *samples* = 5,
sampsize = *min*(*n*, 40 + 2 * k), *trace* = 0, *medoids.x* = TRUE,
keep.data = medoids.x, *rngR* = FALSE, *pamLike* = FALSE, *correct.d* = TRUE).

4.2.2. Grouping Results

The sample data was multi-layered in this study. The first grouping results were four groups. Group one had 1003 projects, group two had 27, three had 57, and four had 8. Since the number and characteristics of group one are not focused enough, group one is divided into three groups. After the analysis, we merged

the group three of the second group with the group two of the first group.

The results were five groups; group one was 257 projects, group two 690, group three 84, group four 57, group five 8. The one, two and three groups grew the number of participants gradually, but the growth rate was different. Group four of participants stopped growing after a few weeks, and group five suddenly increased in the last five weeks. In order to shorten the content, this study only presents the experimental result of groups one and two.

4.3. Experimental Results

Following the above grouping results, we do model construction and verification for the number of participants in the software collaboration project. The number of samples in group 5 is too small, and only data analysis can be conducted. The other four groups divide the data into training data and test data. Through the design of the research model, the factors affecting the number of participants for each group were found, and then the multiple regression analysis was used to construct the prediction model.

The prediction model is divided into Model I and Model II. Model I is a predictive model that hasn't been added to the uncertainty factor, and only uses in the initial project. Model II uses the information that the user participates in the development to make adjustments on the model.

The prediction results are compared with the actual data of the project. The MRE and the average MMRE of each group are calculated. The smaller the number is, the smaller margin of error in the prediction result is. Since one-tenth of the samples were randomly selected as test data, the standard deviation was calculated to avoid the influence of outliers. The study calculates the prediction with an accuracy of plus or minus 25% as an acceptable error range [8].

4.3.1. Group One—Stable Growth Prediction Model

The result of prediction model in Group one is shown in **Table 1**. He group had 257 samples. In the model I, Fix_doc and Fix_developer in a half year had a significant impact on the number of participants; In a year, the Fix_follower of the project owners began to increase. However, Fix_developer, Fix_follower and the affective of participants have an impact on model II. Group one built the number of participants and added the uncertainty factors in Model II. The model explanatory power (R^2) is 0.821 in 26 weeks and 0.813 in a year, which is higher than model I.

According to the evaluation and verification summary table of group one attention number prediction model (**Table 2**) can see that group one had 22 test data, the experimental result show that the semi-annual model I the Pred (0.25) is 45%, MMRE is 0.9, Model II the Pred (0.25) is 77%, MMRE is 0.5; One year model I the Pred (0.25) is 68%, MMRE is 0.5, Model II the Pred (0.25) is 81%, MMRE is 0.36. It can be seen that the prediction model II has higher accuracy rates than the model I in both the Precision and MMRE metrics.

Table 1. The result of prediction model in group one.

Model	Weeks	R^2	Prediction equation
I	26	0.091	3.731 + (−0.006) * Fix_docT + 0.242 * Fix_developer T
	52	0.382	−7.316 + (−0.002) * Fix_docT + 0.159 * Fix_developerT + 0.584 * Fix_followerT
II	26	0.821	−2.755 + 1.39 * Fix_developerT + (−0.149) * Fix_followerT + 0.094 * Dy_commitT + (−0.092) * Dy_issueT + 0.916 * Dy_forkT
	52	0.813	−2.412 + 0.254 * Fix_followerT + (0.104) * Dy_issueT + 0.803 * Dy_forkT

Table 2. Group one evaluation and verification summary table.

Weeks	Model	Prediction (±0.25)	MMRE
26	I	45%	0.9 (σ = 4.6)
	II	77%	0.5 (σ = 3.3)
52	I	68%	0.5 (σ = 1.4)
	II	81%	0.36 (σ = 0.8)

4.3.2. Group Two—Rapid Growth Prediction Model

The result of prediction model in Group two is shown in **Table 3**. Group two had 690 samples. Only Fix_developer has significant impact on model I, R^2 in 26 weeks and a year is 0.22 and 0.2. In model II, the effect of fixed factors is Fix_developer, Fix_follower and the user's uncertainty factors. The number of participants in the predictive model for 26 weeks and a year R^2 were 0.97 and 0.98.

Group two had 85 test data. Among the test results, the correct number of participants in a half year is 41 in Model I, and 52 in Model II. The correct number of participants in a year is 42 in Model I, and 53 in Model II. According to the evaluation and verification summary table of group two attention number prediction model (**Table 4**), the results show that the model II has a prediction accuracy of 61% in a half year and 62% in a year, which is 20% more accurate than the model I.

5. Conclusions

The experimental results show that the MMRE and Pred (0.25) of Model II are better than Model I, and the prediction result of one year is better than half a year. It can be seen that the more data the user accesses, the more accurate the prediction model is. That is, we can improve the accuracy with more participant data.

The impact of the software project dimension on the number of participants is mostly affected by Fix_developer. Fix_language only affects in half a year in group five, but the group five's participants is increased in the late development. Its influence on the project participants is low, which is consistent with the conclusions of the past literature. Fix_doc is only affected in groups one and five in Model I, and has no effect in Model II. In addition, Dy_release has no impact on

Table 3. The result of prediction model in group two.

Model	Weeks	R^2	Prediction equation
I	26	0.22	1.469 + 0.349 * Fix_developerT
	52	0.20	1.274 + 0.346 * Fix_developerT
II	26	0.97	2.053 + 0.017 * Fix_developerT + (−0.169) * Dy_commitT + 0.86 * Dy_issueT + 0.27 * Dy_forkT
	52	0.98	0.933 + 0.14 * Fix_developerT + 0.14 * Fix_followerT + 0.04 * Dy_commitT + 0.954 * Dy_forkT

Table 4. Group two evaluation and verification summary table.

Weeks	Model	Prediction (±0.25)	MMRE
26	I	48%	0.6 (σ = 0.7)
	II	61%	0.5 (σ = 1.3)
52	I	49%	0.4 (σ = 0.8)
	II	62%	0. 3 (σ = 0.6)

each group, so this study believes that users won't care if the project is being developed.

Fix_follower in the project owner dimension is the most influential. For Model II in a year, groups one, two and four have influence; for Model I in a year, groups one and four have influence. Therefore, the project owner needs to interact with users on the platform to enhance the number of participants. Fix_following isn't affected in the Models I and II, which is contrary to the results of past literature. This study believes that we can not only track users, but also build reputation on its platform through participation in the development life cycle.

The impact of the crowd dimension is most influential in Dy_fork, which is consistent with the past literature. The results also show that the crowd dimension has a high degree of influence on the collaborative development process. If the attracted users are willing to interactive in the project, they will have a positive impact on the number of participants. The study also found that users don't care whether the project has been developed, but worry about whether the project is continuously developed or maintained, so all factors in the crowd dimension have an impact on the number of participants.

Acknowledgements

This research was supported by the Ministry of Science and Technology (MOST) of Taiwan under the contract 106-2410-H-011-009.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M. and Damian, D. (2014) The Promises and Perils of Mining GitHub. Proceedings of the 11*th Working Conference on Mining Software Repositories*, 92-101. https://doi.org/10.1145/2597073.2597074

[2] Nielek, R., Jarczyk, O., Pawlak, K., Bukowski, L., Bartusiak, R. and Wierzbicki, A. (2016) Choose a Job You Love: Predicting Choices of Github Developers. *IEEE/WIC/ACM International Conference on Web Intelligence*, 200-207. https://doi.org/10.1109/WI.2016.0037

[3] Pletea, D., Vasilescu, B. and Serebrenik, A. (2014) Security and Emotion: Sentiment Analysis of Security Discussions on Github. Proceedings of the 11*th Working Conference on Mining Software Repositories*, 348-351. https://doi.org/10.1145/2597073.2597117

[4] Borges, H., Hora, A. and Valente, M.T. (2016) Understanding the Factors That Impact the Popularity of GitHub Repositories. 2016 *IEEE International Conference on Software Maintenance and Evolution*, Raleigh, 2-7 October 2016, 334-344. https://doi.org/10.1109/ICSME.2016.31

[5] Izquierdo, J.C., Cosentino, V. and Cabot, J. (2015) Attracting Contributions to Your GitHub Project. *The Journal of Object Technology*.

[6] Weber, S. and Luo, J. (2014) What Makes an Open Source Code Popular on Git Hub? 2014 *IEEE International Conference on Data Mining Workshop* (*ICDMW*), Shenzhen, 14 December 2014, 851-855. https://doi.org/10.1109/ICDMW.2014.55

[7] Chen, F., Li, L., Jiang, J. and Zhang, L. (2014) Predicting the Number of Forks for Open Source Software Project. Proceedings of the 2014 3*rd International Workshop on Evidential Assessment of Software Technologies*, 40-47. https://doi.org/10.1145/2627508.2627515

[8] Yu, Y., Wang, H., Yin, G. and Ling, C.X. (2014) Reviewer Recommender of pull-requests in Github. *IEEE International Conference on Software Maintenance and Evolution*, 609-612. https://doi.org/10.1109/ICSME.2014.107