Data Mining Project History in Open Source Software Communities

Yongqin Gao
University of Notre Dame
ygao1@nd.edu

Yingping Huang
University of Notre Dame
yhuang3@nd.edu

Greg Madey
University of Notre Dame
gmadey@nd.edu

Abstract

Understanding the Open Source Software (OSS) movement came into focus for many researchers due to the recent fast expansion of OSS communities. SourceForge, which is the data source of this research, is one of the biggest OSS communities. While most of the existing research about OSS communities is focused on the community itself, our research is focused on one of the components in the community – the project. With a full database dump from SourceForge, we are able to represent the project history using monthly statistics. Our goal is to find the critical statistical features for the history of the individual projects. Thus based on these features we can generate the rules to predict the future of a project. In this paper, we present the method in three steps. First, we prepare the data for data mining, which include data formatting and feature extraction. Then, we use the non-negative matrix factorization (NMF) method to select independent significant features. In particular, we reduce the size of the feature set from 63 to 10. By using the feature selection method, we can improve the performance of the following step dramatically without sacrificing accuracy. Finally, we use data mining techniques (clustering and summarization) to find the rules and relative features. This research uses the Oracle Data Mining toolkit. By using the method described in this paper, we are able to predict the future of a project with up to 78% confidence.

Contact:
Yongqin Gao
Dept. of Computer Sciences & Engineering
University of Notre Dame
Notre Dame, IN 46556

Tel: 1-574-631-7596
Fax: 1-574-631-9260
Email: ygao1@nd.edu

Key Words: Open Source Software, Data Mining, Clustering, Feature Selection

Data Mining Project History in Open Source Software Communities

Y. Gao, Y. Huang and G. Madey

Open Source Software (OSS) development [Drummond, 1999] is a classic example and prototype of collaborative social networks. Open source ("Open source" is a certification mark owned by the Open Source Initiative http://www.opensource.org) software, usually created by volunteer programmers dispersed worldwide, now competes with that developed by commercial software firms. This is due in part because closed-source proprietary software has associated risks to users. These risks may be summarized as systems taking too long to develop, costing too much and not working very well when delivered. Open source software offers an opportunity to surmount those disadvantages of proprietary software. SourceForge (http://sourceforge.net) is one of the most famous OSS hosting web sites, which offer features like bug tracking, project management, forum service, mailing list distribution, CVS and more. It has more than eighty thousands developers and fifty thousands projects as of spring 2003. It is a good sample of the OSS community to study the underlying mechanisms of OSS communities.

The rest of the paper is structured as follows. Section 2 presents the problem definition and objectives. Section 3 defines the state-of-art of related research. Section 4 discusses the data preparation. Section 5 delves into the feature extraction and data mining, along with some discussion. Finally section 6 is the conclusion.

Problem Definition and Objectives

Although there is more and more research about OSS movement, most of it is focused on the community itself, which we present in the following section. In this paper, we focused on the history of the individual projects. By inspecting the project history, we are able to discover the critical features that determine the attraction of the projects and the rules to judge the success of a project, and eventually predict the future of a project based on this information. This research into project history of Open Source Software can also help us understand other collaborative network and improve other kinds of software development.

We use data mining techniques to investigate the project history. In specific, the techniques include data clustering and data summarization. Our research can be divided into three steps:

1. Data preparation: Since we can not apply the data mining method on the data stored in the database directly, we have to process the data before the next step. The two main tasks in data preparation is data formatting and feature extraction.
2. Feature selection: Also the data mining techniques we are using are all feature-based. But too many features can downgrade the performance dramatically. Fortunately many of the features are dependent or insignificant. So we will use feature selection to select the independent and most significant features in this step.
3. Data mining: We will use the clustering methods to generate clusters of the projects based on features of their histories. Then summarization will be used to generate the rules.

Related Works

Research about Open Source Software is growing rapidly recently. There are four major perspectives.

1. Social science: Recent research include Chan & Lee [Chan & Lee, 2004] and Hemetsberger [Hemetsberger, 2004]. Their paper focus on the economy and business research about Open Source Software.
2. Software engineering: Recent research include Scacchi [Scacchi, 2003] and Elliott [Elliott & Scacchi, 2004]. Their research is mainly the evolution of Open Source Software and its influence to traditional software development.
3. Topology analysis: Recent research include Gao [Gao & Madey, 2003] and Xu [Xu & Gao, 2003]. Their study focuses on understanding the evolution of open source software community from the network perspective.
4. Data mining: Recent research includes Chawla [Chawla, 2003]. This paper is about mining the open source software data using ARN (Association Rules Network).

Data Preparation for Data Mining

The original table of the project history includes descriptive and statistical attributes for every month and every project which stand for one record. There are totally 24 attributes in every record of the table. We selected 21

statistical attributes in this research, which are listed and explained in Table 1. The total number of records in the table is 1098777.

Attribute	Description
Developers	Number of core developers in the group
Downloads	Number of downloads
Site_views	Number of views of the website for the group
Subdomain_views	Number of views of the subdomain for the group
Page_views	Number of views of pages for the group
File_releases	Number of file releases for the group
Msg_posted	Number of the messages posted in the group forum
Bug_opened	Number of bugs opened for development
Bug_closed	Number of bugs closed for development
Support_opened	Number of developing jobs opened for support (feature request)
Support_closed	Number of developing jobs closed for support (feature request)
Patches_opened	Number of jobs opened for patch developing
Patches_closed	Number of jobs closed for patch developing
Artifacts_opened	Number of artifacts opened
Artifacts_closed	Number of artifacts closed
Tasks_opened	Number of tasks assigned to developers
Tasks_closed	Number of tasks finished by developers
Help_requests	Number of help requested by users
CVS_checkouts	Number of CVS checkout activities
CVS_commits	Number of CVS commit activities
CVS_adds	Number of CVS add activities

Table 1: Project history statistical attributes

We extracted the first 6 monthly values of every attribute for every project which has at least six months history from the table. Specifically, we collected 6 values of every attribute for every project. This 6-months time series data of every attribute for every project is the history we will investigate.

In time series analysis, there are two general aspects – trend and seasonality. The former represents a general systematic linear or nonlinear component that changes over time and does not repeat within the time range. The latter represents the pattern that repeats itself in systematic intervals over time. In our research, the time series have just 6 values, so we will just investigate the trend aspect of the time series. Since we used data mining in this research and data mining is feature based analysis, we need to generate features to describe the trends of the time series. In this situation, any single value in the time series is not proper as a feature to describe the trend. So we extracted features to describe the trend instead of using the 6 monthly values of any attribute as the features. We used three features to describe the trends of the time series data for every attribute in every project. These features are the targets of the following data mining. The equations to calculate these features are listed as follows:

$$v_{k,i} = V_{k,i} / V_{k,1}$$

$$F_{k,1} = V_{k,1}$$

$$F_{k,2} = (v_{k,2} - v_{k,1}) + (v_{k,3} - v_{k,2}) + (v_{k,4} - v_{k,3}) + (v_{k,5} - v_{k,4}) + (v_{k,6} - v_{k,5})$$

$$F_{k,3} = \frac{1}{\sqrt{\sum_{i=1}^{6} v_{k,i}^{\ 2} - \frac{(\sum_{j=1}^{6} v_{k,j})^2}{6}}}$$

Where $k = 1..21$ represents different attributes, $i = 1..6$ and $j = 1..6$ represent different months, $V_{k,i}$ is the original value of attribute k in the ith month and $v_{k,i}$ are the normalized ith monthly value of attribute k in given project. The baseline of the normalization is the value of attribute k in the first month, which is $V_{k,1}$.

So there are 3 features for every attribute in every project. Feature 1 is the starting point of the time series, the original value of the first month. Feature 2 is the accumulative differences between normalized adjacent months. And feature 3 is Pearson's correlation coefficient between the normalized time series data and X coordinate. Thus there are totally 63 features for every project.

Feature selection and data mining

After data preparation, we use the NMF (Non-Negative Matrix Factorization) method to select independent and significant features. By using feature selection, we actually reduced the size of the feature set from 63 to 10, which significantly improved the program performance without sacrificing the accuracy. The most significant attributes include file_releases, developers, help_requests and task activities.

Finally we can start the data mining procedures. First, we will use the clustering method to group the projects by the similarity of their histories. The clustering method we used is K-MEAN. The result is a cluster tree and there are 10 leaf clusters, which represent the fine-grained clusters of the projects. The size of these leaf clusters are listed in Table 2.

Cluster ID	Size
1	3201
2	31212
3	51
4	1
5	2
6	14212
7	2
8	5
9	1
10	22
Total	55723

Table 2: Cluster distribution

From these clusters, we can summarize the representative feature sets of different project cluster and generate the rules to describe these clusters. Here are two of the rules:

1. if $F_{file\_releases,2}$ in [1.050, 20.040] and $F_{task\_closed,2}$ in [1.238, 211.590] and $F_{help\_requests,2}$ in [1.385, 9.908] then 6
2. if $F_{file\_releases,2}$ in [20.040, 39.030] and $F_{task\_closed,2}$ in [1.238, 211.590] and $F_{help\_requests,2}$ in [35.477, 44] then 2

Where 2, 6 are cluster ids as in Table 2.

To evaluate the correctness of these rules, we also manually clustered part of the data to 4 classes (FAIL, TOP10, TOP500 and NORMAL). FAIL represents the projects finally failed (number of developers is 0) at the end of the sixth month; TOP10 and TOP500 represent the top 10 and top 500 projects at the end of the sixth month; NORMAL represents the other projects. And we evaluate the rules by comparing the automatic clusters to the manual ones.

We got the supports and confidences for these rules by evaluation. Following the previous sample rules, the support for rule 1 is 0.995 and confidence is 0.78; and the support for rule 2 is 0.112 and confidence is 1.

After evaluation, one interesting discovery is that all the rules for TOP10 and TOP500 all have very small support (smaller than 0.001). This is because the actually projects belongs to these classes are very few, for example there are just 10 projects belongs to TOP10 while the total project number are over 50,000. So this method is not good for discovering 'success' projects due to its limited popularity. We should use an outliner detection method to

investigate these projects. But we can also find that this method can predict the ‘failed’ projects by the project history with good confidence.

References

[Drummond, 1999] G. Drummond, 1999, "Open Source Software and Documents: A Literature and Online Resource Review" *http://www.omar.org/opensource/litreviewa*.

[Chan & Lee, 2004] Chan, Tzu-Ying & Jen-Fang Lee, 2004, "A Comparative, Study of Online Communities Involvement in Product Innovation and Development" *National Cheng Chi University working paper, Taiwan*.

[Hemetsberger, 2004] Hemetsberger, Andrea, 2004, "Fostering Cooperation on The Internet: Social Exchange Processes in Innovative Virtual Consumer Communities" *University of Innsbruck.*

[Scacchi, 2003] Walt Scacchi, 2003, "When is Free/Open Source Software development faster, better and cheaper than software engineering" *Working paper, Institute for Software Research, UC Irvine*.

[Elliott & Scacchi, 2004] Margaret S. Elliott, Walt Scacchi, 2003, "Free Software Development: Cooperation and Conflict in a Virtual Organzational Cluture" *Free/Open Source Software Development, Idea Publishing*.

[Gao & Madey, 2003] Yongqin Gao & Greg Madey, 2003, "Analysis and Modeling of the Open Source Software Community" *NAACSOS*, Pittsburgh, 2003.

[Xu & Gao, 2003] Jin Xu & Yongqin Gao, 2003, "A Docking Experiment: Swarm and Repast for Social Network Modeling", *Agent,* Chicago, 2003

[Chawla, 2003] Sanjay Chawla, 2003, "Mining Open Source Software (OSS) Data Using Association Rules Network" *University of Sydney*, NSW, Australia, 2003