Open Source Projects Analyse Framework Based on Data Mining

LiBo
(Beijing University of Posts and Telecommunications)

Abstract: Open source projects are always a hot topic in the computer industry these years, more and more companies or individual programmers have joined the group to make contribution to the open source projects, especially in the open source communities such as python, Mozilla, etc. However, as there are more and more open source projects which have the similar functions, it becomes difficult for users to recognize which one is best. Ospaf(open source projects analyse framework) is a kind of project which helps to evaluate the open source projects. Ospaf based on big data and uses data mining way to evaluate each open source projects..

Key words: Artificial Intellegence,Open Source Projects Evaluation, Statistics, Machine Learning, Big data

0 Introduction

With the development of calculating capacity of computers. People could analyze a great amount of data in the shorter time. At the same time, many online businesses or services have collected many data based on people's on-line activities. In that case, using machine learning method to analyse big data is becoming an efficient way to evaluate the certain things or guide people's behavior. Ospaf collects data from social coding groups such as github、blackducks、google-code. From the basic datas we have got such as the number of stars, the time sequence, the commites, we extract certain traits[1]. Then after managing the collected data by denoising and normalization we get the target data. Then we use common machine learning algorithms like logical regression and k-means to get the final math model. Thus, we use this model to evaluate the open source projects in order to get the rank of these projects.

In this paper, the target data is getting from Github-api and BlackDucks, we will show how to deal with these data and extract traits. What's more, how to use machine learning method to get the final math model will be introduced. Moreover, we carefully crafted the methodology to ensure scientific validity and rigor when dealing with small data sizes.

The contributions of this paper are summerized as follow:

1. The way to get the traits from data which comes from github or other open source projects platform.
2. How to find relationship between these statistics, and extract the features by analyzing these statistics.
3. The paper will show how to use machine learning algorithms to get the math mode, which will be used to evaluate these open source projects
4. We use some textual analysis way to evaluate the open source projects..
5. Get the score of each open source project.

1 Previous Works

We need to prepare the dataset for analyzing. However, there are a lot of works to do. First, we should get the original dataset. Second, we need to deal with these original data by denoising[2] and normalization. Third, as we want to use the machine-learning algorithm logical regression, it is a kind of supervised learning method, so we should divide the data into two groups, one is the

Brief author introduction:LiBo(1990),Male,Master,Data mining. E-mail: garvinli@garvinli.com

positive group and the other one is negative group.

1.1 Get original data

When we execute Github-api in order to get the data of a certain open source project. It returns a dataset which is in a Json format[3]. We use python standard library Json to get the statistics which we need in our research. The data which comes from Github-api is showing below,

```
GarvinLideMacBook-Pro:~ Garvin$ curl -G https://api.github.com/orgs/octokit/repos
[
  {
    "id": 417862,
    "name": "octokit.rb",
    "full_name": "octokit/octokit.rb",
    "owner": {
      "login": "octokit",
      "id": 3430433,
      "avatar_url": "https://avatars.githubusercontent.com/u/3430433?v=3",
      "gravatar_id": "",
      "url": "https://api.github.com/users/octokit",
      "html_url": "https://github.com/octokit",
      "followers_url": "https://api.github.com/users/octokit/followers",
      "following_url": "https://api.github.com/users/octokit/following{/other_user}",
      "gists_url": "https://api.github.com/users/octokit/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/octokit/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/octokit/subscriptions",
      "organizations_url": "https://api.github.com/users/octokit/orgs",
      "repos_url": "https://api.github.com/users/octokit/repos",
      "events_url": "https://api.github.com/users/octokit/events{/privacy}",
      "received_events_url": "https://api.github.com/users/octokit/received_events",
      "type": "Organization",
      "site_admin": false
    },
```

Fig.1 github-api data

1.2 Extract traits from original data

First, we have some original data, such as the stars, forks, issues, etc. We get these traits from Github-api directly. Second, when we combine some traits, they will reflect different meaning, for example, when we divide the number of stars with the length of time which the project has been build, it will reflect the growth rate of the trait-star[4]. Third, because there is much noise in the dataset, we use the normal distribution principle to denoise. What's more, different kinds of traits have different units. Normalization is used to resolve this problem.

1.3 Get the label set

Label set is used to divide data set into two groups, one of these groups is the mature ones and the other group is the immature ones. The final target dataset is as following(This is just a sample):

Tab.1 target dataset

Name	Size	Star	Watch	Label
gsa-prototype	252	6	6	0
votigoto	116	4	4	0
Twitter	27096	2939	2939	1
Insoshi	5380	2	2	0
Rubycon	1096	3	3	0
Hhvm	321662	10170	10170	1
vegmadison	160	1	1	0

We can easily get the list of mature ones from Github's explore or open source foundations.

2 Analytics Methods

We fill all these traits and label in the matrix. Every column of this matrix is a kind of trait, the last column is the label set. Every row of this matrix is the statistics of an open source project.

2.1 Logical Regression

Many educational research problems call for the analysis and prediction of a dichotomous-outcome: whether a student will succeed in college, whether a child should be classified as learning disabled (LD), whether a teenager is prone to engage in risky behaviors, and so on. Traditionally, these research questions were addressed by either ordinary least squares (OLS) regression or linear discriminant function analysis[5]. Both techniques were subsequently found to be less than ideal for handling dichotomous outcomes due to their strict statistical assumptions, i.e., linearity, normality, and continuity for OLS regression and multivariate normality with equal variances and covariances for discriminant analysis[6].

In this paper we use this regression method to classify these open source projects. The following is the introduction of how we use logical regression to get the final math model. After we have got the matrix of target dataset. We use the formula sigmoid to make sure that all the final result is between 0~1, “0” indicates the poor open source projects while “1” indicates the mature ones. The equation is

$$S(x) = \frac{1}{1 + e^{-a_1 x_1 - a_2 x_2 - a_3 x_3 \ldots\ldots}} \tag{1}$$

The graph of sigmoid[7] function is as blow, when we only take two treats into account.

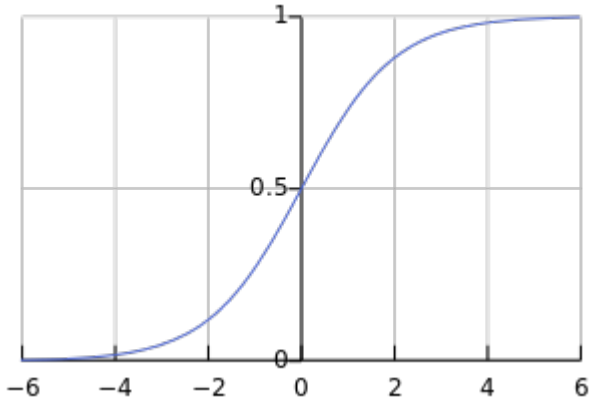

Fig.2 sigmoid function

From the formula above, we fill all the X with the value of every projects’ column value and then used gradient descent method to get the regression mode. Gradient descent is a first-order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent is also known as steepest descent, or the method of steepest descent. When known as the latter, gradient descent should not be confused with the method of steepest descent for approximating integrals.

After all the data pass through the logical regression algorithm, we get the coefficients of the

equation(1), and these coefficients indicate the weight of traits, The bigger the coefficient is, the bigger its trait influences the maturity of the project.

We use this math model to evaluate every open source project and get the rank of these projects.

2.2 Natural Logarithm

After using the math model get score of every open source project. We found there is too much distinct between each score, especially between the better ones and the worse ones. That is because we use the method normalization[8]. When we get the data in to evaluation method, some of these score is too small, nearly zero, what's more some of these score are as big as 100.

In order to reflects the degree of these projects objectively. Natural Logarithm is used to make the score curve more smoothly.

The original model is as blow:

$$Final\_score = \log_B A + C \tag{2}$$

The parameter A in the formular is defined as the score of project. Then B is used to adjust the degree of the score.

2.3 Textual Analysis

We find that textual analysis could also be used in OSPAF to determine the degree of activeness of a open source project. Here is the graph of our textual analysis,

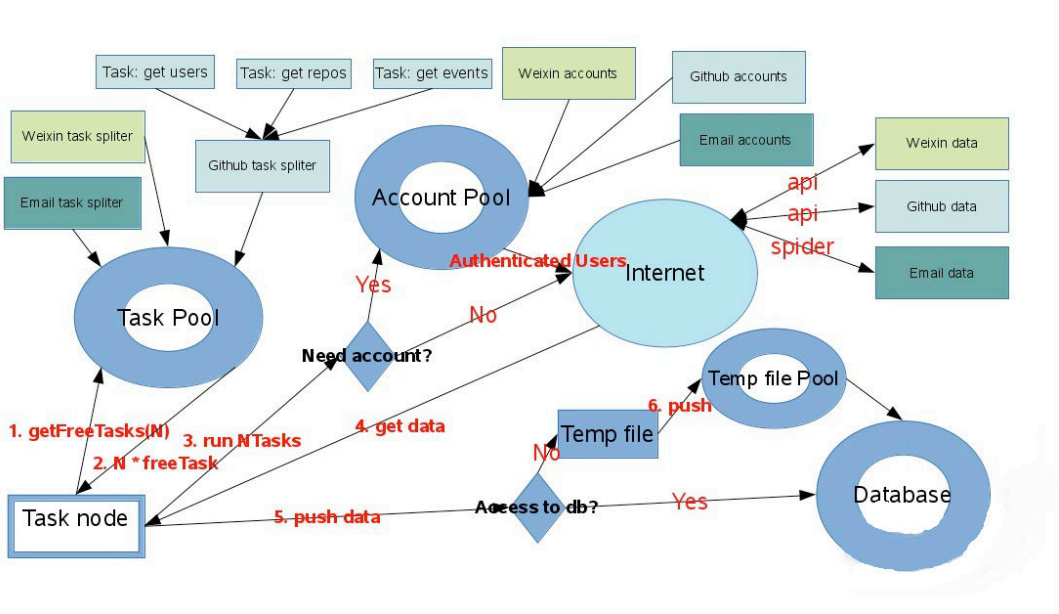

Fig.3 textual analysis graph

We could get to know if the issues of a projects could be solved in time by analyzing the commits information of the projects[9]. What's more, we could value the atmosphere of a open source organization by analyzing their email information or even wechart information.

3 Result

In this paper, we take ten thousand data of open source projects into account, and finally extract 27 traits, and the weight of every trait is as follow:

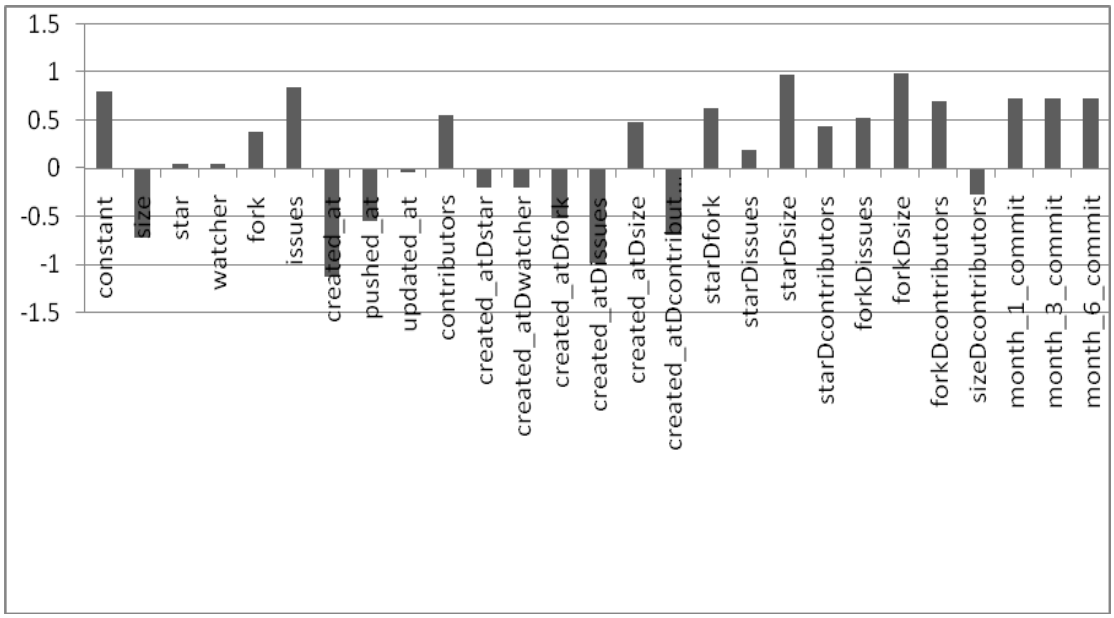

Fig.4 weights of the traits

From these figures, the positive figure means this trait has a positive influence on the maturity of the open source project, the negative figure reflects this trait has a negative effect on the maturity of the open source project. For instance, the trait fork/size influence the maturity of the open source project most[10], so if we want to find a mature project, we would better find the one with a big number of fork/size.

Here is the graph show how we get the result of every open source project using OSPAF.

```
100  6555  100  6555    0     0   2725      0  0:00:02  0:00:02 --:--:--  2726
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
  0 30283    0   203    0     0     89      0  0:05:40  0:00:02  0:05:38    88
100 30283  100 30283    0     0  11298      0  0:00:02  0:00:02 --:--:-- 11295
{"score": "98.327639", "name": "ReactiveCocoa/ReactiveCocoa", "level": "activity_3"}
[Finished in 36.1s]
```

Fig.5 graph of platform

Here are some examples that we used OSPAF to get the activeness of some open source projects.

Tab.2 some results of OSPAF

Organization	Name	Score	Level
Twitter	twemproxy	96.83920	activity_2
Dmlc	minerva	2.69092	inactivity_2
Nodejs	node	98.561180	activity_3
Github	Rebel	0.039443	inactivity_1
ReactiveCocoa	ReactiveCocoa	98.327639	activity_3
Sampsyo	beets	88.317246	normal_3

From this chart, we get to know that some famous projects such as node and ReactiveCocoa have a relative high level of activeness. We distinct the activeness of open source projects into three levels, the inactive ones, the normal ones and the active ones, the bigger the number after each level, the more active this projects is.

4 Conclusion

It is always a problem for users to evaluate which open source is more mature or more suitable. OSPAF offers a way to help resolve this problem, by collecting the data from Github and dealing these data with machine learning algorithm, we get the score of each open source project, in that case we can talk about which one is better.

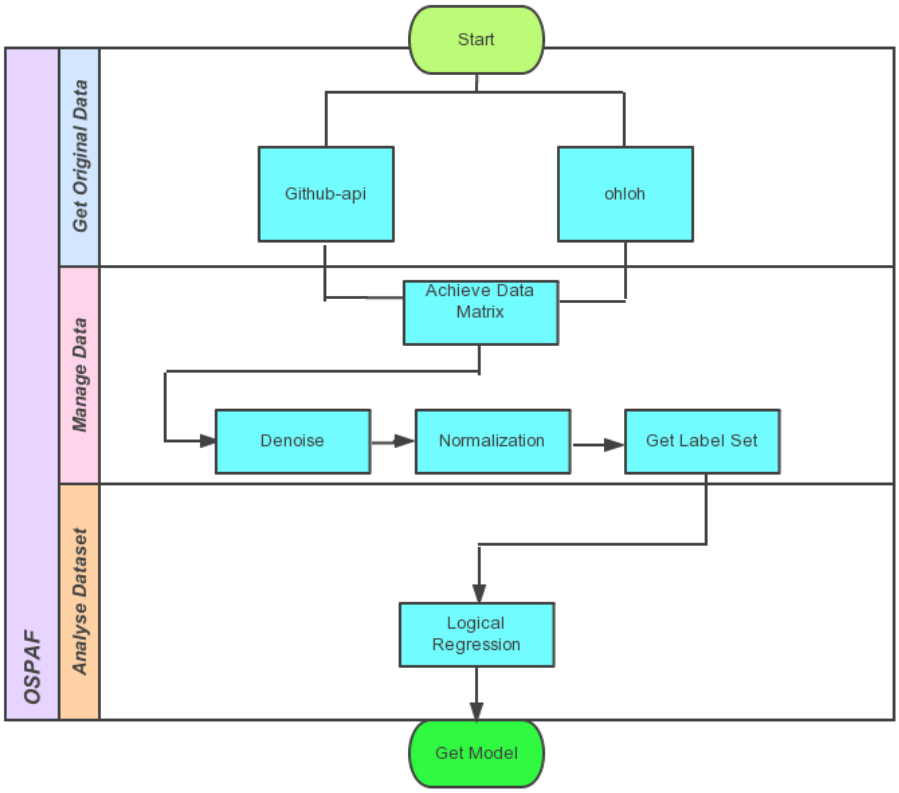

Fig.6 flow chart of ospaf

What's more, OSPAF also helps users or the developers to get to know which index influence the maturity of the open source project most, and how much it influenced. We have already finished first version of OSPAF and have published it on the Github.

References

[1] Thung F, Bissyande T F, Lo D, et al. Network Structure of Social Coding in GitHub[J]. Proceedings of the Euromicro Conference on Software Maintenance & Reengineering Csmr, 2013, 88(2):323 - 326.
[2] , Bissyande T F, Lo D, et al. Network Structure of Social Coding in GitHub[C].Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on. IEEE, 2013:323 - 326.
[3] Jing Jiang,Li Zhang; Lei Li. Understanding project dissemination on a social coding site[J]. Reverse Engineering (WCRE), 2013,53(3): 132 - 141
[4] Hastie H, T, Tibshirani R. 1998. Additive Logistic Regression: A statistical view of Boosting[J]. Annals of Statistics, 1998, 28(1):2000.
[5] D. Surian, D. Lo, and E.-P. Lim. Mining collaboration patterns from a large developer network[J]. WCRE, 2010,65(46): 269-273.
[6] Pollock J L. The logical foundations of goal-regression planning in autonomous agents[J]. Artificial Intelligence, 1998, 106(2):267-334.
[7] 刘涵, 刘丁. 基于模糊 sigmoid 核的支持向量机回归建模[J]. 控制理论与应用, 2006, 23(2):204-208.
[8] Gardner B J. The natural exponential comes before the natural logarithm.[J]. International Journal of Mathematical Education in Science & Technology, 1994, (1):5.

[9] Carley K. Coding Choices for Textual Analysis: A Comparison of Content Analysis and Map Analysis[J]. Sociological Methodology, 1993, 23:75-126.
[10] Jurado F, Rodriguez P. Sentiment Analysis in monitoring software development processes: An exploratory case study on GitHub's project issues[J]. Journal of Systems & Software, 2015,104(3):82–89.

基于数据挖掘的开源项目成熟度分析工具

李博
（北京邮电大学,信息与通信学院）

摘要：近些年，开源项目一直是计算机工程领域的热门话题。越来越多的公司或是个人开发者加入了贡献开源项目的行列。特别是一些比较有名的开源社区，像是 python 或是 Mozilla。但是，随着开源项目越来越多，具有相似功能的开源项目也变得越来越多，这使得开源项目使用者很难区分哪一个开源项目更好，更适合自己的工程。开源项目成熟度分析工具（ospaf）是一款开源项目成熟的的评估工具，初衷是帮助开源项目使用者评估每个开源项目的健壮程度。OSPAF 以大数据为根据，通过数据挖掘的方法计算评估模型，从而对每个开源项目进行评估。

关键词：人工智能，开源项目评估，数据统计，机器学习，大数据
中图分类号：请查阅《中国图书馆分类法》