Predicting Bug Priority: Project Proposal

Alexandar Mihaylov, Luisa Rojas

December 4, 2017

1 Problem Statement

In this project, the focus will be on time optimization in regard to the bug resolution process. The approach to this problem will be the automation of bug priority classification. Thus, given any high impact bug, the objective will be to identify its priority and classify it into one out of five priority categories: Critical, Trivial, Blocker, Major or Minor.

Bug priority categorization allows for improved project management, which includes the delegation of bug resolution amongst developers of different expertise, as well estimated bug fix times. For example, given a bug priority of critical, the bug could be delegated to the most experienced developer to most adequately deal with its resolution.

2 Motivation

Various methods in Software Engineering research focus on predicting, localizing, and triaging bugs; however, they rarely consider their impact on the system.

This is very relevant to the open problem of bug resolution since prior to issue assignation to a developer, first, the nature of the fix needs to be determined. Currently, this task is done manually by project managers in order to adequately delegate bug fixes according to the developer's area of expertise. The automation of this task would allow for efficient resource allocation.

3 Dataset

The dataset used in this project is the one created in the work *"Dataset of High Impact Bugs: Manually Classified Bug Reports"* [1], which contains a set of 1,000 issues of for high impact bugs from 4 Apache projects: Ambari, Camel, Derby, and Wicket[1]. The issues were all mined from the projects' JIRA[2] repositories, which allowed for the extraction of data shown in Figure 1.

[1] This dataset is currently available at http://oss.sys.wakayama-u.ac.jp/?p=1009.

[2] JIRA is an project management tool used for issue and software tracking [2], available at https://www.atlassian.com/software/jira

TABLE II
INFORMATION INCLUDED IN OUR DATASET

NAME	Info.
issue_id	Issue ID
type	Type of an issue (BUG or IMPROVEMENT)
status	Status of an issue (Resolved or Closed)
resolution	Resolution type of an issue (FIXED only)
component	Target component/s
priority	Priority of an issue
reporter	Repoter's name
created	Time and Day of an issue reported
assigned	Time and Day of an issue assigned
assignee	Assignee's Name
resolved	Time and Day of an issue resolved
time_resolved	Time to resolve an issue (created to resolved)
time_fixed	Time to fix an issue (assigned to resolved)
summary	Summary of an issue
description	Descriptions of an issue
affected_version	Versions affected by an issue
fixed_version	Versions of a fixed issue
votes	Number of votes
watches	Number of watchers
description_words	Number of words used in descriptions
assignee_count	Number of assignees
comment_count	Number of comments for an issue
commenter_count	Number of developers who comment on an issue
commit_count	Number of commits to resolve an issue
file_count	Number of committed files to resolve an issue
files	Committed file names and paths

Figure 1: The categories included, per entry, from the JIRA issues dataset [1].

4 Methodology

The classification approach, as well as the tools used to implement it, will be described in the subsections below.

4.1 Machine Learning

We will implement a Neural Network to classify the 5 different types of bugs discussed previously. The different aspects to be taken into account are the following:

1. **Feature Selection**: Too many features are provided by the original dataset. For this reason, the features to be used will need to be selected manually based on relevance and information availability according to the bug resolution cycle. Then, from the remaining features, different combinations can be tested to find the optimal feature set.

2. **Sentiment Analysis**: Textual attributes will be quantified by applying sentiment analysis models. These new features generated would replace the textual ones.

3. **Hidden Layers**: We aim to determine the optimal number of hidden layers, as well as their perceptron count and corresponding activation functions.

Finally, the model will be applied to all projects and the results will be compared.

4.2 Tools

The system will be developed using Python as the main programming language as well as the Tensorflow library for the implementation of the Machine Learning models.

References

[1] M. Ohira, Y. Kashiwa, Y. Yamatani, H. Yoshiyuki, Y. Maeda, N. Limsettho, K. Fujino, H. Hata, A. Ihara, and K. Matsumoto. A dataset of high impact bugs: Manually-classified issue reports. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 518–521, May 2015.

[2] M. Ortu, G. Destefanis, B. Adams, A. Murgia, M. Marchesi, and R. Tonelli. The jira repository dataset: Understanding social aspects of software development. In *Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering*, PROMISE '15, pages 1:1–1:4, New York, NY, USA, 2015. ACM.