Data set validity: why the large majority of the projects collected should not be handled as toy projects

P. Vassiliadis 2021-01-18

Introduction

The goal of this document is to make a concise and high-level, yet clear, assessment of whether the schema evolution dataset, collected in 2019, involves projects that are "toy" projects, class assignments, or other projects that do not demonstrate effectively the way Free Open-Source Software projects evolve their schemata.

Proactive Filtering at project selection

As mentioned in the paper, we have applied several filtering criteria to eliminate, to the extent possible, toy projects. Specifically:

- When selecting projects from Github Activity Data and Libraries.io, we applied a filter requesting that each eligible project has more than one contributor, more than 0 stars, and is a non-forking project.
- Afterwards, we excluded all results whose file descriptions included the terms 'test' or 'demo' or 'example' in the path.

Therefore, the original selection criteria included filters to reduce the possibility of toy projects being selected. As we describe next, the mining of the history of the entire history of collected projects, verifies the selection process to a large extent.

Project Durations: a breakdown

The first post-hoc assessment of the validity of the dataset collection is performed with respect to the distribution of the time span of the updates made to each *project* (attn: not schema, but project). *Short project durations would signify either (a) that the project is too recent, xor, (b) that the project was abandoned soon after its start. Long durations on the other hand, signify projects that were maintained over a longer period of time*.

Caveat: naturally, *this duration-based hypothesis* does not replace the in-depth study of each individual project. However, with 195 projects at hand (a) this is not feasible with any reasonable effort, and, (b) it *is a more conservative criterion than the actual study would possibly reveal* (i.e., one can see how a more in-depth could possibly validate some of the short-PUP projects, e.g., as young ones, recently ignited, where it seems rather difficult to imagine how other projects could be invalidated).

Data Collection. As the paper also mentions, in May 2019, we cloned the projects from Github and "git logged" (i.e., extracted) the history of their commits. So, for each commit, the date, list of modified files, and commit message was retrieved.

By keeping record of the dates of first and last commit of the project and the first and last commit of the schema, we were able to obtain the Project Update Period (PUP) and the Schema Update Period (SUP) that are mentioned in the paper and here.

	<=12M	[13-24M]	>24M	TOTAL	<=12M	[13-24M]	>24M	TOTAL
FROZEN	7	4	23	34	21%	12%	68%	100%
AL.FROZEN	17	10	38	65	26%	15%	58%	100%
FS+Frozen	10	4	11	25	40%	16%	44%	100%
MODERATE	4	4	21	29	14%	14%	72%	100%
FS+Low	5	1	14	20	25%	5%	70%	100%
ACTIVE	1	1	20	22	5%	5%	91%	100%
	44	24	127	**195**	23%	12%	65%	100%

Table 1. Breakdown of project durations per taxon; right hand side percentages refer to each rows

Observations. Observe Table 1. We group the projects by

(a) taxon, and

(b) the time span of project updates, PUP (i.e., the time between the first commit and the last commit our git clone tracked) in (i) less than 1 year, (ii) between 1 and 2 years, and (iii) longer than 2 years.

With the exception of Focused Shot and Frozen, the rest of the tax demonstrate an overwhelming majority of projects having PUP higher than 2 years. If one extends the validity time span to include PUP between 1 and 2 years, 151 out of 195 projects are valid candidates for study (i.e., 3 out of 4).

We find the above finding to be quite strong a statement. Take into consideration that half the taxa demonstrate very small -or even zero- amount of change at their schema. At the same time, and in sharp contrast to the above, their vast majority, esp., in the taxa of higher activity, the project time spans are significant.

Threats to validity. Despite all these, are there any counter-arguments supporting the possibility that the studied projects could be of toy nature? The only potentially invalid projects would be the 7 + 17 = 24 projects of the Frozen + Almost Frozen categories. Clearly, such a statement requires a more in-depth study, to see what is going on with these projects. However, their number is quite small (24/195), therefore, we argue that their presence does not alter the significance of our results by any means.

Summary. *Summarizing, we can say with confidence, that for at least the two thirds of the projects, …*

- *…the projects demonstrate project update time spans that do not pertain to toy projects, but to projects of systematic development, and,*
- *… developers have been updating the project, but not the schema.*

The joint breakdown of Project Update Period and Schema Update Period

We have drilled-down to the aforementioned breakdown, by exploring the joint distribution of PUP and SUP, and we present the results in Table 2. The columns of the Table represent completed years for the Schema Update Period, whereas the rows of the Table represent the completed years for the Project Update Period of the monitored projects. The upper right triangle could only be empty, of course. In black, bold font, we depict the marginal sums. In the more extreme column and row we depict cumulative sums that we will explain in the sequel.

For 195 prjs SUP Years $\longrightarrow$

PUP Years $\downarrow$	**0**	**1**	**2**	**3**	**4**	**5**	**7**	**8**	**Grand Total**	**Inv. Cumul. sum**
0	20%	0%	0%	0%	0%	0%	0%	0%	**20%**	100%
1	9%	5%	0%	0%	0%	0%	0%	0%	**14%**	80%
2	11%	5%	3%	0%	0%	0%	0%	0%	**18%**	66%
3	8%	2%	4%	2%	0%	0%	0%	0%	**14%**	48%
4	6%	2%	2%	2%	2%	0%	0%	0%	**13%**	33%
5	3%	1%	1%	1%	1%	1%	0%	0%	**6%**	20%
6	2%	0%	1%	0%	1%	1%	1%	0%	**5%**	14%
7	0%	1%	0%	1%	1%	1%	0%	0%	**3%**	9%
8	0%	1%	1%	1%	0%	0%	0%	0%	**2%**	7%
9	0%	0%	0%	1%	0%	0%	0%	0%	**1%**	5%
10	0%	1%	0%	0%	1%	1%	0%	1%	**3%**	4%
11	1%	0%	0%	0%	0%	0%	0%	0%	**1%**	2%
12	0%	0%	0%	0%	0%	0%	0%	1%	**1%**	1%
16	0%	1%	0%	0%	0%	0%	0%	0%	**1%**	1%
Grand Total	**59%**	**15%**	**11%**	**6%**	**4%**	**3%**	**1%**	**2%**	**100%**	
Cumul. Sum	59%	74%	85%	91%	95%	98%	98%	100%		

Table 2. Joint distribution of quantized PUP and SUP in completed years; the year labels indicate round-down lower limits -- e.g., a label of 0 indicates a period of [0 months .. 12 months), a label of 1 indicates a period between [13 months… 24 months), etc. The percentages are all with respect to the 195 projects of the dataset.

The results show a single cell, [0,0] with a 20% of projects that could be eligible for an invalidation test (the difference with the 21% of Table 1 is due to the limit: one is <=12 months and the other is < 12 months). *The rest of the projects exceed 1 year of PUP*.

Coming to the cumulative sums, the situation is inverse in terms of how the distribution of projects behave, and thus the point of having an inverse cumulative sum for PUP and a cumulative sum for SUP.

- The inverse cumulative sum for PUP starts counting at projects with the maximum PUP (16 years) and ends at the 20% of projects with 0 completed years of PUP.
- The cumulative sum for SUP starts with the 59% of projects having less than one completed year of SUP and ends at the maximum SUP of 8 completed years.

Observe the inverse cumulative sum for **Project Update Period**. 80% of the projects have project duration greater than 1 year, 66% of the projects have a duration greater than 2 years, 48% (almost half!) the projects have a duration greater than 3 years, etc. *This is a clear statement of the validity of the data set, as it includes projects that were being updated for a long period of time*.

At the very same time, for **Schema Update Period**, 59% of the projects have less than 1 year of Schema Update Period, 74 less than 2 Years, etc. In other words, we have practically the inverse distribution of values compared to Project Update Period, grossly biased towards short values. *In other words, whereas the projects kept having updates for long periods of time, the schema updates were focused in much shorter periods*.