多版本软件抽象语法树分析

周航 MF1833105

Motivation

团队合作的开源项目是一个很值得研究的问题，我们主要考虑 Commit 间、连续发布版本间等项目的总 AST 变化，得到一些可以值得深究的结论，希望可以借此能合理安排团队资源配置，发掘软件演化规律、指导演化方向。

Research Question

RQ1. 连续的 Commit 间，模块 AST 的变化可能有多大?
RQ2. 连续的发布版本间，模块 AST 的变化可能有多大?(Bug-fix 版本、功能版本)
RQ3. 在什么情况下，相近的两次 Commit 会使得模块 AST 发生大的变化?
RQ4. 开发过程中，模块 AST 有没有可能发生连续的大的变化?

Methods

RQ1.
我们对开源项目 matplotlib 作为实验数据对象，用 git log 命令获取格式化项目提交总数据，随后主要是对文本进行分析：

1.统计 commit sha 值及其对应的修改文件列表(过滤掉非 py 文件)
2.对每一个 sha 值使用 git checkout --quiet {sha} -- {filename}命令指定文件回退到指定版本， 使用 git checkout --quiet {sha} ~1 -- {filename}回退到指定版本的前一版本，随后进入 AST 模块
3.在 AST 模块使用编辑距离算法计算文件 AST 结构差异值累加值 distance
4.将 commit sha 值、distance、文件修改数目输出到 commit_dist.txt

针对 commit_dist.txt 进行抽样，选取其中连续 500 次 commit 画出直方统计图

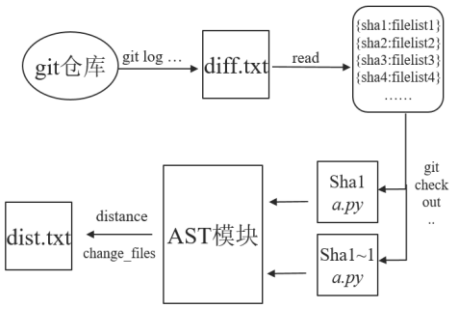
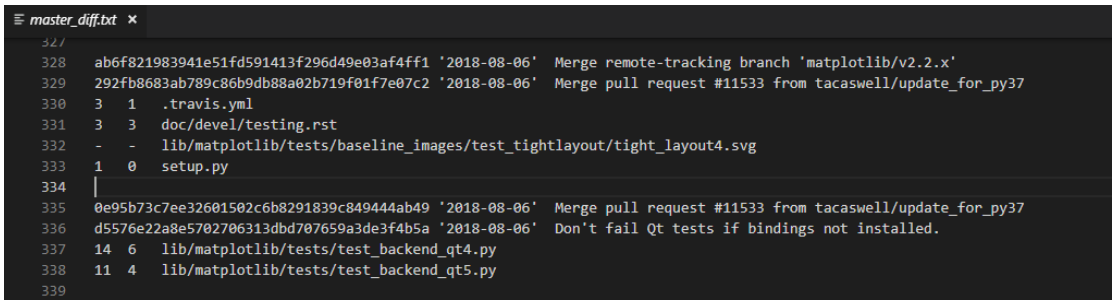
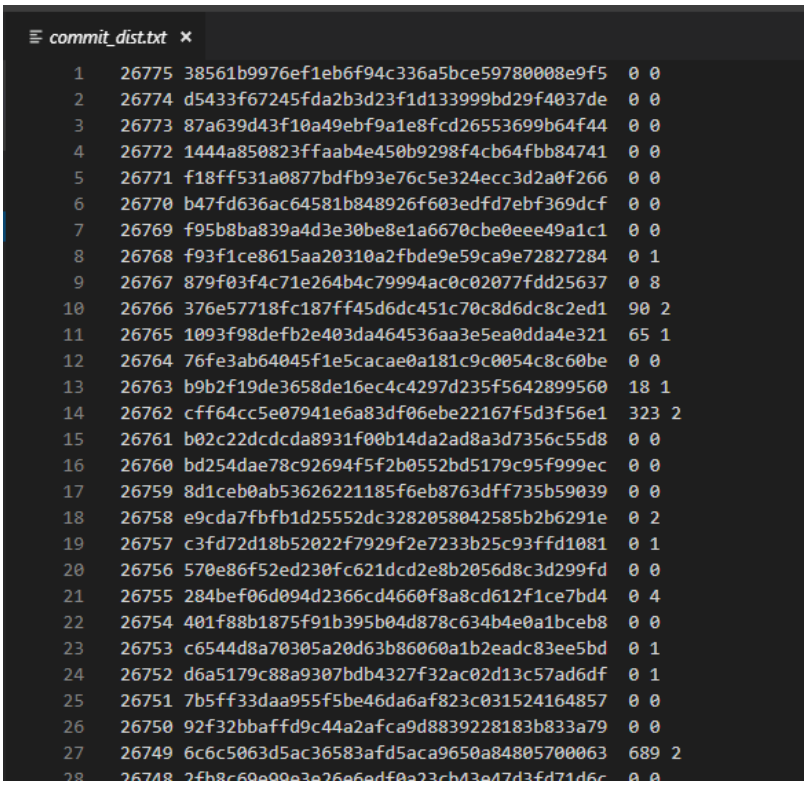


Fig1.实验方法



```
master_diff.txt
328  ab6f821983941e51fd591413f296d49e03af4ff1 '2018-08-06'  Merge remote-tracking branch 'matplotlib/v2.2.x'
329  292fb8683ab789c86b9db88a02b719f01f7e07c2 '2018-08-06'  Merge pull request #11533 from tacaswell/update_for_py37
330  3   1   .travis.yml
331  3   3   doc/devel/testing.rst
332  -   -   lib/matplotlib/tests/baseline_images/test_tightlayout/tight_layout4.svg
333  1   0   setup.py
334
335  0e95b73c7ee32601502c6b8291839c849444ab49 '2018-08-06'  Merge pull request #11533 from tacaswell/update_for_py37
336  d5576e22a8e5702706313dbd707659a3de3f4b5a '2018-08-06'  Don't fail Qt tests if bindings not installed.
337  14  6   lib/matplotlib/tests/test_backend_qt4.py
338  11  4   lib/matplotlib/tests/test_backend_qt5.py
339
```

Fig2.diff 信息



```
commit_dist.txt
1   26775 38561b9976ef1eb6f94c336a5bce59780008e9f5  0 0
2   26774 d5433f67245fda2b3d23f1d133999bd29f4037de  0 0
3   26773 87a639d43f10a49ebf9a1e8fcd26553699b64f44  0 0
4   26772 1444a850823ffaab4e450b9298f4cb64fbb84741  0 0
5   26771 f18ff531a0877bdfb93e76c5e324ecc3d2a0f266  0 0
6   26770 b47fd636ac64581b848926f603edfd7ebf369dcf  0 0
7   26769 f95b8ba839a4d3e30be8e1a6670cbe0eee49a1c1  0 0
8   26768 f93f1ce8615aa20310a2fbde9e59ca9e72827284  0 1
9   26767 879f03f4c71e264b4c79994ac0c02077fdd25637  0 8
10  26766 376e57718fc187ff45d6dc451c70c8d6dc8c2ed1  90 2
11  26765 1093f98defb2e403da464536aa3e5ea0dda4e321  65 1
12  26764 76fe3ab64045f1e5cacae0a181c9c0054c8c60be  0 0
13  26763 b9b2f19de3658de16ec4c4297d235f5642899560  18 1
14  26762 cff64cc5e07941e6a83df06ebe22167f5d3f56e1  323 2
15  26761 b02c22dcdcda8931f00b14da2ad8a3d7356c55d8  0 0
16  26760 bd254dae78c92694f5f2b0552bd5179c95f999ec  0 0
17  26759 8d1ceb0ab53626221185f6eb8763dff735b59039  0 0
18  26758 e9cda7fbfb1d25552dc32820580425 85b2b6291e  0 2
19  26757 c3fd72d18b52022f7929f2e7233b25c93ffd1081  0 1
20  26756 570e86f52ed230fc621dcd2e8b2056d8c3d299fd  0 0
21  26755 284bef06d094d2366cd4660f8a8cd612f1ce7bd4  0 4
22  26754 401f88b1875f91b395b04d878c634b4e0a1bceb8  0 0
23  26753 c6544d8a70305a20d63b86060a1b2eadc83ee5bd  0 1
24  26752 d6a5179c88a9307bdb4327f32ac02d13c57ad6df  0 1
25  26751 7b5ff33daa955f5be46da6af823c031524164857  0 0
26  26750 92f32bbaffd9c44a2afca9d8839228183b833a79  0 0
27  26749 6c6c5063d5ac36583afd5aca9650a84805700063  689 2
```

Fig3. commit.dist 信息

RQ2.

收集了 v1.0.0~v3.0.1 28 个版本

1. 使用 git reset –hard tag 命令回退指定版本
2. 递归遍历文件目录获取文件名和文件内容
3. 比较相邻版本相同的文件名的内容的 AST 结构差异值 distance 及修改文件数
4. 将每个版本与其之前版本的 distance 和 changed_file 可视化在折线图中观察规律

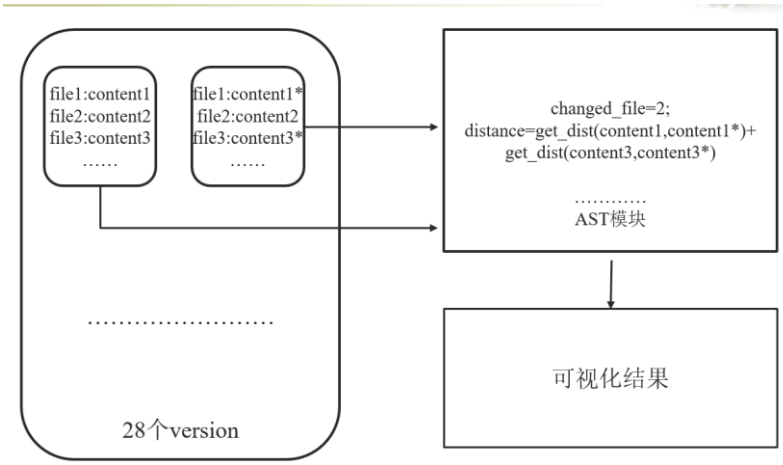


Fig4.RQ2 流程图

RQ3.

1. 将 git log 信息与 RQ1 得到的 commit_dist.txt 整合并将 distance 从小到大排序
2. 选取排序最后的 1000 次 commit，统计提交语句中单词词频
3. 选取排序前 1000 次 commit,统计提交词频
4. 人工审查得到结论

RQ4.

1. 将 git log 信息与 RQ1 得到的 commit_dist.txt 整合并将 distance 从小到大排序
2. 设计 window 和 threshold 的值，记录下连续超过 window 次 commit 且每次的 distance 超过 threshold 的开发过程 Window=5 ， Threshold=max（dist）*0.8
3. 人工审查得到结论

Experiment Results

RQ1.

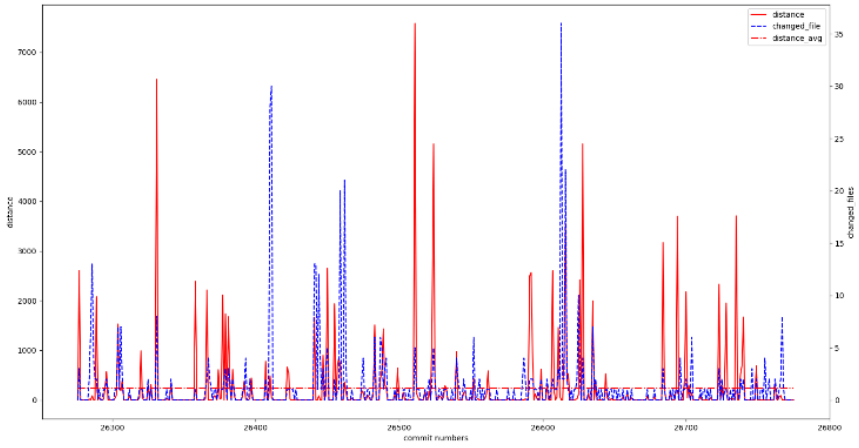


Fig5.RQ1 实验结果（distance、changed_file 数）

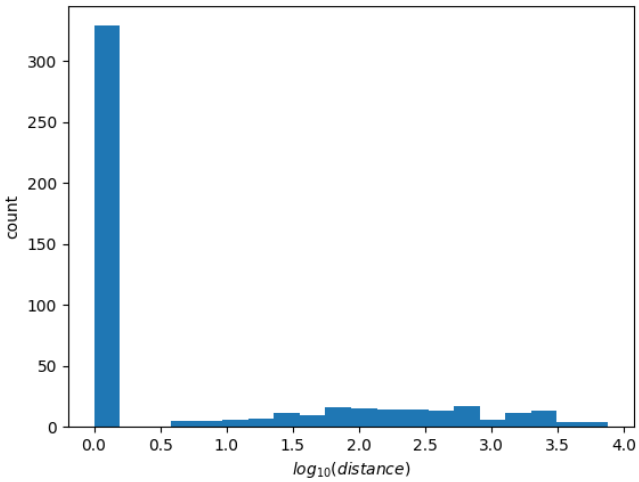


Fig6.RQ1 实验结果（distance 分布）

在我们的 dataset 一共获取了 26775 个 commit 并进行了相邻 commit 之间的分析，有如下发现：

有 16227 个 commit 与上一版本模块 AST 变化为 0（60.6%）

大部分 commit 模块 AST 变化值在均值以下

会出现个别相邻 commit 间模块 AST 变化值出现峰值

可能原因：remove/add /test

差异值与 Commit 数目统计直方图成长尾效应

RQ2.

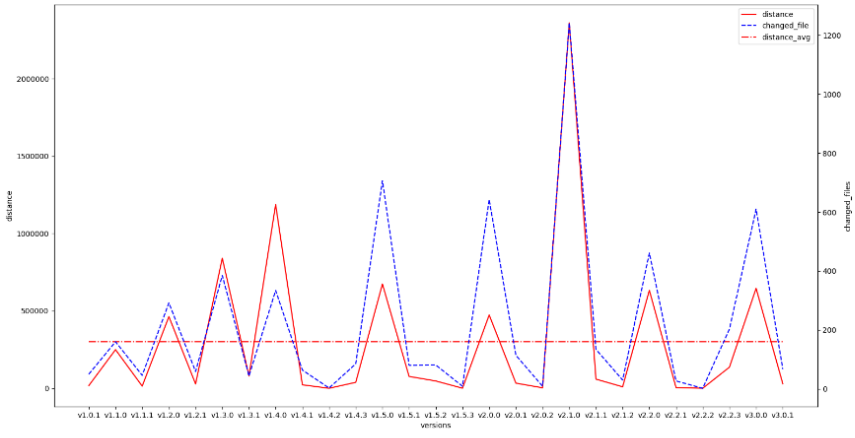


Fig7.RQ2 实验结果 版本间 AST 变化

在我们的 dataset 一共收集了 28 个版本并递归比较了其版本间的变化，分析折线图得到了以下发现：

发布版本（x.x.0）会出现 AST 变化值峰值

Bug-Fix 版本间 AST 变化值处于均值以下

发布版本间 AST 变化值处于均值以上

RQ3.

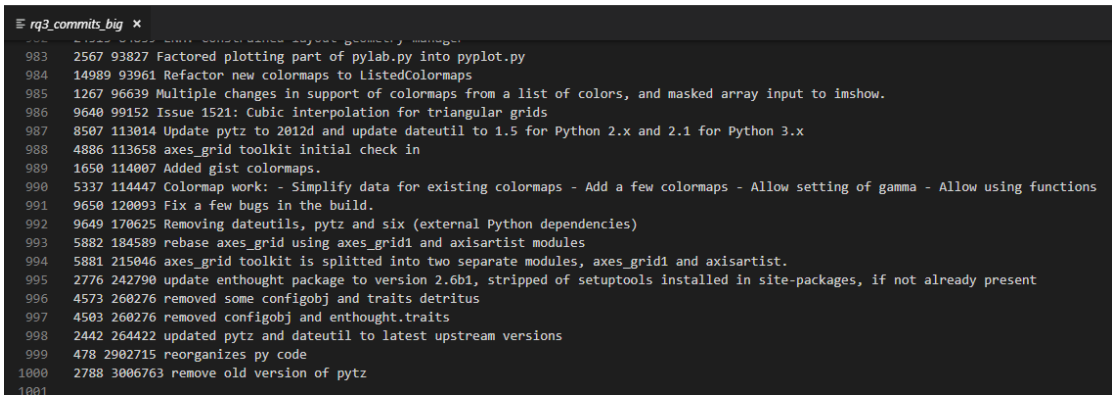


Fig8.RQ3 实验结果 AST 差异较大的 commit 信息

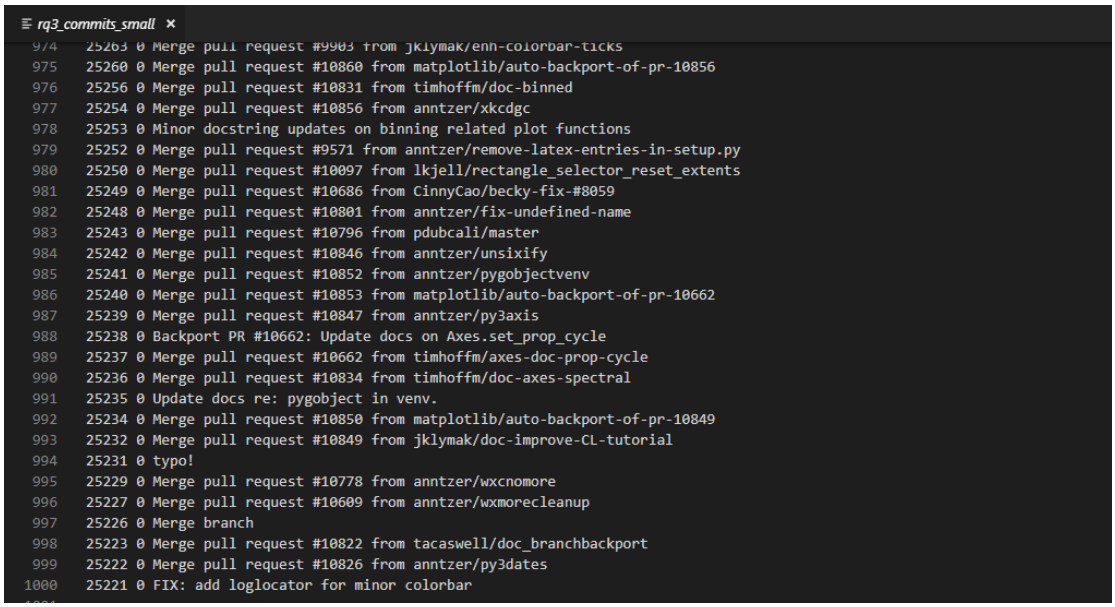


Fig9.RQ3 实验结果 有大量 AST 差异为 0 的 commit

针对模块 AST 变化值进行排序后，统计了变化值最小和最大的 1000 次 commit 并进行词频统计，有如下发现

变化值最大的 1000 次 commit：

“add”，“remove”，“test”等频率较高可能原因是增删代码、增加测试、导入包变化、语法变化等“remove old version of pytz”//删除旧版本“Add tests for installing into a completely clean virtual environment”//增加新 test

变化值最小的 1000 次 commit：

含变化值为 0 的结果为“merge” ，“pull” ，“request”等频率较高例如“Merge pull request #10965 from matplotlib/auto-backport-of-pr-10962”

不含变化值为 0 的结果：“fix” ，“add” ，“test”等频率较高，可能原因更多在于 bug 的修复等例如“fix pep8 issue”//修复 bug

RQ4.

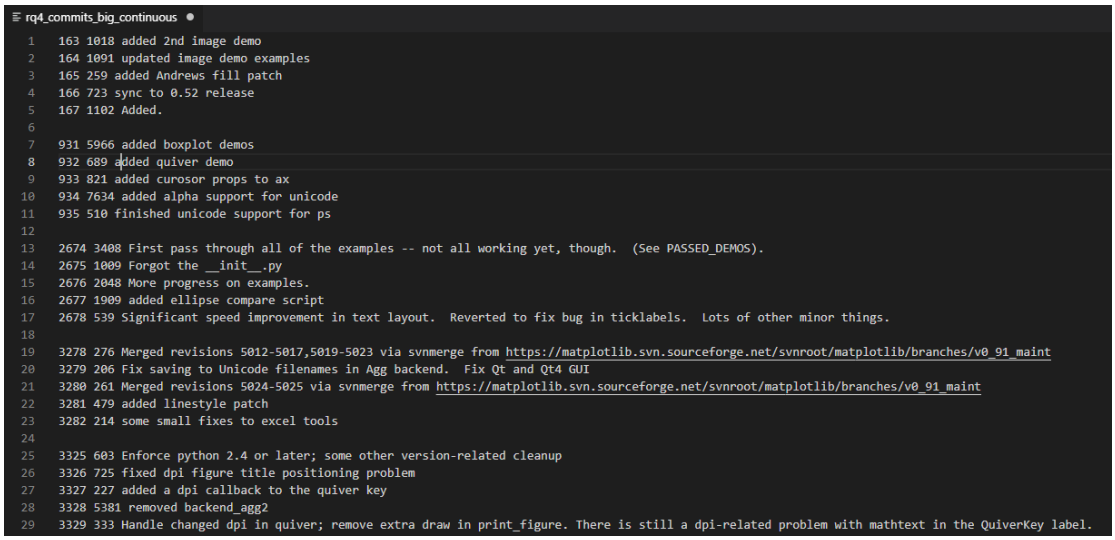


Fig10.RQ4 实验结果 连续开发 AST 差异较大的 commit 信息

在我们设置的 threshold 和 window 下我们对统计结果进行审查有如下发现：

在 26775 个 commit 间我们仅找到了 51 个这样的连续开发过程，模块 AST 发生连续大的变化

存在来自同一提交者同一模式的 commit：

第 5395 次~5409 次 commit 均为“test conversion: move old-style test to new-style test”

第 9937 次~9945 次 commit 均为“Clean-up and move **”

Conclusion

通过以上四个研究问题，我们挖掘了一些开源项目在开发过程中 AST 抽象语法树差异变化得到了一些有趣的规律。研究方法和这些规律可以在某种程度上客观衡量项目的健康程度以及指引软件演化的方向。