间桶中计算平均的评价指标.

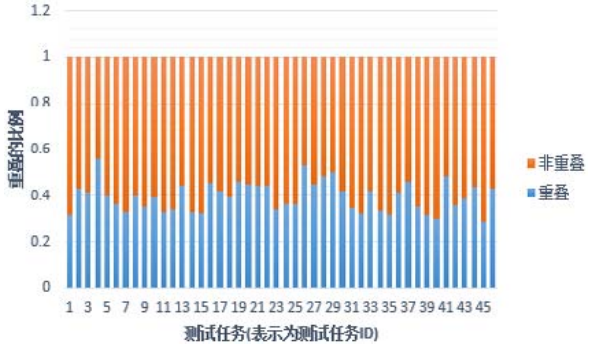


Fig.9 Overlap of selected workers by the three aspects

图 9 3 个方面选择出的工作者的重合率

如图 10 所示:历史数据的多少确实对模型的性能存在影响.在历史数据积累较少时,模型的性能相对较低.这是因为模型是基于历史数据的.实际上,在历史数据少的情况下,也没有更好的方法来预测.然而随着历史数据的增加,本文方法的性能迅速趋近于前面留一交叉验证实验的结果(留一交叉验证利用了更多的历史数据,如前文所述,除了该众测任务的数据,其余数据全部作为训练集合).由此可知:本文方法能有效利用历史数据,并且迅速达到较优的预测结果.

此外,在考虑测试任务的先后顺序后,我们发现测试任务的先后顺序对于本文方法的性能并没有明显的影响,即在考虑众测任务的时间顺序的情况下,模型随着时间不断接近交叉验证结果.这说明用户随着时间的历史经验积累是比较稳定、持续的过程(如果用户性质随着时间发生剧烈变化,那么后一个时间点上的性能有可能相比前一个时间点不增加,甚至倒退).如图 10 所示:不同时间点的性能曲线,随着时间是一个比较平稳并比较快速的收敛过程,并没有出现不增长或者倒退,说明了用户的历史经验积累是稳定的,并不存在剧烈的波动变化.

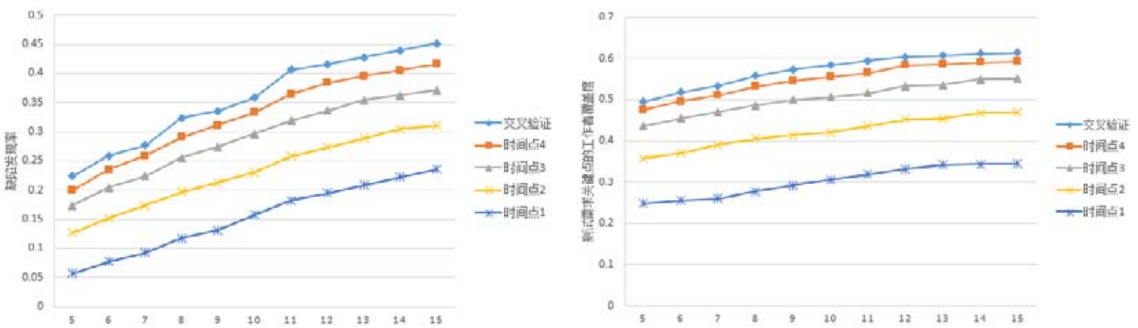


Fig.10 Performance of our approach in different time points

图 10 不同时间点上方法的性能对比

5 讨 论

5.1 *k*的设置

众所周知,众测经常涉及很多的工作者来执行某测试任务.读者可能会质疑实验中的 k 设置的比较小.这是因为第 1.3 节提到:本文用到的实验数据集中,对于某个测试任务,平均有 37 个工作者提交测试报告,平均 15 个工作者检测到缺陷.真实的众测平台上的真实场景,使得我们采用这样的实验设置.进一步的,我们方法的目的是选择尽可能少的工作者,检测到尽可能多的缺陷.这能够在保持测试产出不变的情况下,减少测试任务发布者

的开销,使他们更愿意在众测平台上发布任务,从而促进众测平台的繁荣.

此外,实验验证结果显示:当选择同样数目的工作者时,相比其他方法,本文方法能够检测到更多的缺陷.因此认为当 k 变大时,本文方法仍然可以得到较好的性能.

5.2 工作者选择策略

读者可能认为,对于测试需求中的每个技术术语,工作者选择方法需要选择多于一个工作者来进行覆盖.这是因为在推送模式下,不是每个工作者都会接受邀请并且执行测试任务的.实际上,在本文方法中,对于测试需求中的每个技术术语,确实能够选择到多于一个的工作者.第 1 个原因是,本文方法同时考虑其他方面,例如主动性和相关性,这会帮助选择到覆盖同一技术方面的工作者;第 2 个原因是,当实现多样性时,我们使用概率相关的度量,该度量在选择人员时,只是暂时降低已经满足的技术方面的概率,并不是完全去掉这些已经满足的技术方面.这样的处理使得本文方法可以为某一技术方面选择多于一个的工作者.

5.3 方法对新老工作者倾向性分析

由于本文方法基于工作者在众测平台的历史数据,为众测任务选取工作者,那么本文方法是否倾向于为众测平台工作时间较长的老工作者,而忽视刚进入到众测平台、历史数据不多的新工作者呢?首先,在第 4.4 节的研究问题 1 中,通过比较本文方法和单纯选取老用户方法的性能,结果显示本文方法性能明显优于单纯选取老用户的方法,其中,老用户就是平台上工作时间最长的老工作者集合(类似基线方法中的活跃用户方法).这个结果说明,单纯选择老用户并不能提高缺陷发现率.进一步的,我们对本文方法的所选工作者进行深入分析,如图 11 所示,在比较了本文方法和老用户方法在 46 个测试任务中,当 k=15 时所选出的工作者提交过测试报告的分布.我们可以发现:本文方法并不倾向选出提交测试报告最多的那一部分老工作者,而主要集中在比较有经验、在不同的任务中又有相应专业知识的工作者.

众测平台上有部分新工作者,即刚进入平台、还没有历史数据的工作者.由于本文方法是基于工作者的历史进行的建模和推荐,所以不能为他们推荐任务.后续工作会考虑基于工作者填写的属性信息为他们进行推荐,从而增加他们参与任务的积极性.新工作者可以自己搜索在线正在发布的任务去参与,待工作者有历史数据后,再由本文方法为其推荐众测任务.

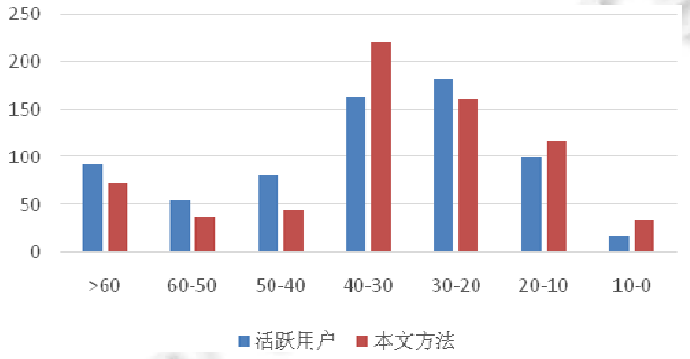


Fig.11 Number of test reports of workers selected by our approach

图 11 方法选择的工作者历史提交测试报告的数目

5.4 对有效性的威胁

对外部有效性的威胁主要是本研究的通用性.

- 首先,实验数据包含了从中国最大的众测平台之一收集到的 46 个测试任务,我们不能完全保证在别的实验环境下也能得到同样的结果.然而,我们的数据集相对较大,而且其中包含各个领域的项目(例如音乐、工具、游戏等),这帮助我们在一定程度上减轻了这个威胁;
- 其次,本研究中的所有众测报告都是用中文写的,我们不能保证在其他语言的众测项目上得到同样的

结果.然而,因为我们没有进行语义分析,只是对文本进行分词,用词作为标记来进行建模,因此,该威胁得到极大的减轻.

对于内部有效性的威胁,方法中涉及的 3 个参数θ_1,θ_2和θ_3可能影响我们的实验结果.为了控制该方面的威胁,我们抽样得到部分数据,并用交叉检验的方式来选择可以得到最好性能的参数集合.此外,我们度量主动性是假设主动性在一定的时间窗口内是保持稳定的,即,工作者的主动性在一定时间内不会随时间发生剧烈变化.然而在很长的时间范围内,工作者的主动性确实可能发生转变,比如有些原来主动的工作者变得不主动了、原来不主动的工作者逐渐主动.我们计划在未来的工作中,研究时间对于主动性的影响.

对于构造有效性的威胁,本研究考虑主动性、相关性和多样性这 3 个方面.这 3 个方面是从不同的角度设计的:单个工作者的独特性、一组工作者的多样性、他们的经验和任务的关系等.然而,其他方面也可能会影响测试需求的覆盖度和缺陷的检测率.为了解决这个威胁,需要对其他方面进行研究和建模.

6 相关工作

6.1 众 测

随着工业环境下众测模式的快速发展,众测也吸引着越来越多学术界的研究者.众测通过将众包的概念引入测试领域,能够帮助集中的软件开发和测试工程师发现缺陷[9−11].

众测相关的研究主要分为两个方向.

- 用众测这种新兴的测试模式来辅助解决传统软件测试中的问题.Pastore 等人[12]研究是否能够用众测解决 oracle 问题,他们将反映当前程序的行为组织成断言,并将这些断言作为众测任务发布到众测平台上,工作者需要评估这些断言的正确性.Liu 等人[13]将众测应用到可用性测试的研究中,他们通过经验研究,发现众测对于可用性测试的适用性和价值.Nebeling 等人[14]开发了一个工具包,可以支持众测模式下的网页测试,该工具包不仅能够快速地招募大量的工作者,还能够在不同的条件下评估网站.
- 关注如何解决众测环境下产生的新问题.Feng 等人[1]提出一种方法对众测环境下的测试报告进行排序,他们综合运用多样性策略和风险策略,动态选择测试报告进行检查.Wang 等人[2,3]提出的方法可以从大量的测试报告中分类得到真正含有缺陷的测试报告.Tung 等人[15]研究如何更有效地为协同的测试任务分配工作者,他们将该问题建模为线性规划问题.

我们的工作关注的是众测环境下新产生的问题,也就是如何为众测任务选择一组合适的工作者.根据我们的调研,这是第 1 个此种类型的工作.

6.2 缺陷检测和测试覆盖

缺陷检测是软件测试活动中的一个重要目标[16].在传统的测试中,很多方面被认为会影响缺陷检测,例如测试代码质量、测试用例选择、测试充分性准则等.Athanasiou 等人[17]通过反映测试代码质量的 3 个方面——完整性、有效性和可维护性来评估测试代码质量.结果显示,测试代码质量和缺陷检测率之间存在显著相关性.Rothermel 等人[18]给出几种方法对测试用例进行排序,结果表明,测试用例排序可以显著地提升缺陷检测率.Zhou 等人[19]研究哪些测试的充分性准则对于测试 java 数据库应用是最适合的,结果发现,语句覆盖或者分支覆盖是最有效的.不同于之前的工作,我们的研究聚焦人员相关的因素,这是众测环境下新产生的,并且对于众测是很关键的.

覆盖性是软件测试活动的另一个指标,已有的研究从多个不同的角度研究覆盖性,例如测试用例的覆盖、测试数据的覆盖等.Gopinath 等人[20]将覆盖性准则作为测试集的质量指标.Leon 等人[21]基于多元可视化技术提出一种新的测试数据选择方法.Mondal 等人[22]在几个真实的测试用例选择的案例研究中,比较了代码覆盖和测试用例多样化的关系.本文方法关乎测试需求的的覆盖性,我们从工作者的角度研究覆盖性,并且用人员经验的多样性来提高这个覆盖度.此外,我们不仅从多样性的角度研究测试需求的覆盖性,还考虑可能影响覆盖性的其他方面.

6.3 人员选择

软件开发已经成为一项越来越开放的活动,其中经常涉及来自开放的大众工作者.为某个软件开发任务推荐合适的工作者正变得越来越重要.有很多相关工作关注为各种各样的软件开发任务选择合适的工作者,例如推荐缺陷修复人、为新成员推荐导师、推荐某个领域的专家等.

Jeong 等人[23]研究如何为一个新的缺陷报告推荐可能修复该缺陷的开发者,他们引入了基于马尔科夫链的图模型来建模缺陷报告在人员之间的转移历史.Tamrawi 等人[24]提出一种新的方法进行缺陷修复人的推荐,他们通过缓存开发者的缺陷修复历史以及基于模糊集的方法.Canfora 等人[25]通过挖掘邮件列表和版本控制系统的数据,为开源社区中的新成员推荐导师.Ma 等人[26]提出了使用专长的概念,并且通过评估专家推荐的效果说明该概念的可行性.

已有的研究要么只是推荐一个工作者,要么假设推荐的一组工作者是相互独立的.然而在众测环境下,我们需要选择一组工作者,并且工作者之间是互相依赖的,因为他们共同完成一个团队工作.

7 结束语

由于众测环境下的工作者分布在不同地域,有着不同的测试经验,哪些工作者执行某一测试任务会大大影响测试需求关键点覆盖度和缺陷检测率.本文识别了众测环境下影响工作者缺陷发现的 3 个方面,分别是主动性、相关性和多样性,并且提出一种众测环境下的人员选择方法.该方法能够同时考虑主动性、相关性和多样性,从而提高测试需求关键点覆盖度和缺陷检测率.我们基于百度众测的真实数据验证了本方法,结果显示了方法的有效性,当选择 15 个工作者时,本文方法可以得到 62%的测试需求关键点覆盖度和 45%的缺陷检测率,优于基线方法.

在未来的工作中,我们计划研究其他可能影响众测环境下缺陷发现的方面,考虑工作者的主动性、专业方向是否会随时间发生转变.并且,我们一直和百度保持着密切的合作,计划将本文方法部署到线上环境,从而更好地验证方法在实际环境下的有效性.

References:

[1] Feng Y, Chen Z, Jones JA, Fang C, Xu B. Test report prioritization to assist crowdsourced testing. In: Proc. of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015). New York: ACM Press, 2015. 225−236.

[2] Wang J, Cui Q, Wang Q, Wang S. Towards effectively test report classification to assist crowdsourced testing. In: Proc. of the 10th ACM/IEEE Int'l Symp. on Empirical Software Engineering and Measurement (ESEM 2016). 2016. 6:1−6:10.

[3] Wang J, Wang S, Cui Q, Wang Q. Local-Based active classification of test report to assist crowdsourced testing. In: Proc. of the 31st IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE 2016). 2016. 190−201.

[4] Hochba DS. Approximation algorithms for NP-hard problems. ACM Sigact News, 1997,28(2):40−52.

[5] Hiemstra D. Using Language Models for Information retrieval. Taaluitgeverij Neslia Paniculata, 2001.

[6] Canfora G, Di Penta M, Oliveto R, Panichella S. Who is going to mentor newcomers in open source projects? In: Proc. of the ACM SIGSOFT 20th Int'l Symp. on the Foundations of Software Engineering (ESEC/FSE 2012). New York: ACM Press, 2012. 44:1−44:11.

[7] Dror G, Koren Y, Maarek Y, Szpektor I. I want to answer; Who has a question? Yahoo! Answers recommender system. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2011). New York: ACM Press, 2011. 1109−1117.

[8] Zhao Z, Wei F, Zhou M, Chen W, Ng W. Crowd-Selection query processing in crowdsourcing databases: Atask-driven approach. In: Proc. of the 18th Int'l Conf. on Extending Database Technology (EDBT 2015). 2015. 397−408.

[9] Chen Z, Luo B. Quasi-Crowdsourcing testing for educational projects. In: Companion Proc. of the 36th Int'l Conf. on Software Engineering (ICSE Companion 2014). New York: ACM Press, 2014. 272−275.

[10] Mao K, Capra L, Harman M, Jia Y. A survey of theuse of crowdsourcing in software engineering. Technical Report. 2015.

[11] Feng JH, Li GL, Feng JH. A survey on crowdsourcing. Chinese Journal on Computers, 2015,38(9):1713−1726 (in Chinese with English abstract).

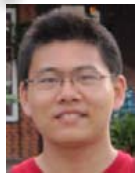
[12] Pastore F, Mariani L, Fraser G. Crowd oracles: Can the crowd solve the oracle problem? In: Proc. of the 2013 IEEE 6th Int'l Conf. on Software Testing, Verification and Validation (ICST 2013). 2013. 342−351.

[13] Liu D, Bias RG, Lease M, Kuipers R. Crowdsourcing for usability testing. Proc. of the American Society for Information Science and Technology, 2012,49(1):1−10.
[14] Nebeling M, Speicher M, Grossniklaus M, Norrie MC. Crowdsourced Web Site Evaluation with Crowdstudy. Springer-Verlag, 2012.
[15] Tung YH, Tseng SS. A novel approach to collaborative testing in a crowdsourcing environment. Journal of Systems and Software, 2013,86(8):2143−2153.
[16] Bertolino A. Software testing research: Achievements, challenges, dreams. In: Proc. of the 2007 Future of Software Engineering (FOSE 2007). Washington: IEEE Computer Society, 2007. 85−103.
[17] Athanasiou D, Nugroho A, Visser J, Zaidman A. Test code quality and its relation to issue handling performance. IEEE Trans. on Software Engineering, 2014,40(11):1100−1125.
[18] Rothermel G, Untch RH, Chu C, Harrold MJ. Test case prioritization: An empirical study. In: Proc. of the IEEE Int'l Conf. on Software Maintenance (ICSM'99). 1999. 179−188.
[19] Zhou C, Frankl P. Empirical studies on test effectiveness for database applications. In: Proc. of the 2012 IEEE 5th Int'l Conf. on Software Testing, Verification and validation (ICST 2012). 2012. 61−70.
[20] Gopinath R, Jensen C, Groce A. Code coveragefor suite evaluation by developers. In: Proc. of the 36th Int'l Conf. on Software Engineering (ICSE 2014). New York: ACM Press, 2014. 72−82.
[21] Leon D, Podgurski A, White LJ. Multi variatevisualization in observation-based testing. In: Proc. of the 22nd Int'l Conf. on Software Engineering (ICSE 2000). New York: ACM Press, 2000. 116−125.
[22] Mondal D, Hemmati H, Durocher S. Exploring testsuite diversification and code coverage in multi-objectivetest case selection. In: Proc. of the 2015 IEEE 8th Int'l Conf. on Software Testing, Verification and Validation (ICST 2015). 2015. 1−10.
[23] Jeong G, Kim S, Zimmermann T. Improving bugtriage with bug tossing graphs. In: Proc. of the 7th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symp. on The Foundations of Software Engineering (ESEC/FSE 2009). New York: ACM Press, 2009. 111−120.
[24] Tamrawi A, Nguyen TT, Al-Kofahi JM, Nguyen TN. Fuzzy set and cache-based approach for bug triaging. In: Proc. of the 19th ACM SIGSOFT Symp. and the 13th European Conf. on Foundations of Software Engineering (ESEC/FSE 2011). New York: ACM Press, 2011. 365−375.
[25] Ma D, Schuler D, Zimmermann T, Sillito J. Expertrecommendation with usage expertise. In: Proc. of the IEEE Int'l Conf. on Software Maintenance (ICSM 2009). 2009. 535−538.
[26] Yan X, Guo J, Lan Y, Cheng X. A biterm topic model for short texts. In: Proc. of the 22nd Int'l Conf. on World Wide Web (WWW 2013). New York: ACM Press, 2013. 1445−1456.
[27] Wang S, Liu T, Tan L. Automatically learning semantic features for defect prediction. In: Proc. of the 38th Int'l Conf. on Software Engineering (ICSE 2016). New York: ACM Press, 2016. 297−308.
[28] Villarroel L, Bavota G, Russo B, Oliveto R, Penta M. Release planning of mobile apps based on user reviews. In: Proc. of the 38th Int'l Conf. on Software Engineering (ICSE 2016). New York: ACM Press, 2016. 14−24.
[29] Zhou M, Mockus A. What make long term contributors: Willingness and opportunity in OSS community. In: Proc. of the 34th Int'l Conf. on Software Engineering (ICSE). 2012. 518−528.

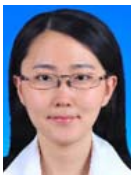
附中文参考文献:

[11] 冯剑红,李国良,冯建华.众包技术研究综述.计算机学报,2015,38(9):1713−1726.

崔强(1985－),男,辽宁抚顺人,博士生,CCF学生会员,主要研究领域为众测,推荐算法.



谢淼(1988－),男,博士,工程师,主要研究领域为数据挖掘,软件工程.



王俊杰(1987－),女,博士,副研究员,主要研究领域为缺陷预测,经验软件工程,众测.

王青(1964－),女,博士,研究员,博士生导师,CCF高级会员,主要研究领域为软件过程技术,需求工程,软件质量与管理.