计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20160653
53(12):2768-2782,2016

软件仓库挖掘领域:贡献者和研究热点

江 贺 陈 信 张静宣 韩雪娇 徐秀娟
(大连理工大学软件学院 辽宁大连 116024)
(jianghe@dlut.edu.cn)

Mining Software Repositories: Contributors and Hot Topics

Jiang He, Chen Xin, Zhang Jingxuan, Han Xuejiao, and Xu Xiujuan
(*School of Software, Dalian University of Technology, Dalian, Liaoning* 116024)

Abstract Software updates and evolves continuously over time, software repositories accumulate massive data. How to effectively collect, organize, and make use of these data has become a key problem in software engineering. Mining Software Repositories (MSR) aim to mine useful knowledge contained in complex and diversified data to improve the quality and productivity of software. Although some studies have elaborately summarized the background, history, and prospects about MSR, existing studies do not present systematically the most influential author, institution, and country as well as the major research topics and their transitions over time. Therefore, this study combines the existing classical publication analysis frameworks and algorithms to analyze the relationships among publications related to MSR, and presents some important domain knowledge for researchers in detail. To effectively tackle this task, we construct a framework named MSR Publication Analysis Framework (MSR-PAF). MSR-PAF consists of three components which can be used to create a dataset for the study, conduct a bibliography analysis, and implement a collaboration pattern analysis, respectively. The results of the bibliography analysis show that the most productive author, institution, and country are Ahmed E. Hassan, University of Victoria, and USA, respectively. The most frequent keyword is software maintenance and the most influential author is Abram Hindle. In addition, the results of the collaboration pattern analysis show that Abram Hindle is the most active author, and open source project and software maintenance are the most popular research topics.

Key words publication analysis; collaboration pattern analysis; data mining; mining software repositories; big data

摘 要 随着时间的推移,软件不断地更新和演化,软件仓库中累积了海量的数据,如何有效地收集、组织、利用软件工程中涌现的软件大数据是一个至关重要的问题.软件仓库挖掘(mining software repositories, MSR)通过挖掘软件仓库中繁杂多变的数据中蕴含的知识来提高软件的质量和生产效率.虽然一些研究工作详细阐述了MSR的背景、历史和前景,但现有的研究工作并未系统地呈现MSR领域中最有影响力的作者、机构、国家以及最受欢迎的研究主题和主题变迁等领域知识.因此,结合已有的经典的文献分析框架和算法来分析MSR相关文献,并呈现一些MSR基本领域知识.为了实现MSR文献分析,建立了一个包含3个组件的MSR文献分析框架(MSR publication analysis framework, MSR-

收稿日期:2016-08-24;**修回日期**:2016-10-24
基金项目:国家自然科学基金项目(61370144);教育部新世纪优秀人才支持计划基金项目(NCET-13-0073)
This work was supported by the National Natural Science Foundation of China (61370144) and the Program for New Century Excellent Talents in University of Ministry of Education of China (NCET-13-0073).

PAF),这 3 个组件分别被用来创建数据集、执行基础文献分析、实施合作模式分析.基础文献分析结果表明:最高产的作者、机构、国家/地区分别是 Ahmed E. Hassan,University of Victoria 和美国,最有影响力作者是 Ahmed E. Hassan,最频繁的关键词是 software maintenance.合作模式分析的结果显示 Abram Hindle 是 MSR 领域最活跃的作者,open source project 和 software maintenance 是最流行的研究主题.

关键词　文献分析;合作模式分析;数据挖掘;软件仓库挖掘;大数据

中图法分类号　TP311

在互联网的推动下,软件工程正经历重大变革,软件的规模和复杂性急剧增加.为了方便软件管理,一些工具如版本控制系统、缺陷追踪系统等已被广泛应用到软件开发活动中,记录软件的每一次测试活动、每一次代码变更、每一次缺陷修复等[1].随着时间的推移,软件仓库中积累了海量的、不同类型的数据,包括开发过程中的源代码、需求文档;软件测试时的测试实例、bug 报告;系统运行时的日志文件、事件记录等[2].这些数据呈现出体量(volume)、增速(velocity)、多样(variety)、价值(value)、真伪(veracity)、可验性(verification)、可变性(variability)以及临近性(vicinity)等多"V"特点[3],对软件工程提出了重大挑战.因此,如何有效地收集、组织、利用这些大数据来帮助改善软件的质量和生产效率已成为大数据背景下软件工程中一个至关重要的问题.

软件仓库挖掘(mining software repositories,MSR)是一个新兴的软件工程领域,通过数据挖掘技术分析软件仓库中海量的数据,来提高软件的质量和生产效率[4-6].我们引入一个典型的软件仓库挖掘任务——开发者优先级识别,来详细呈现软件仓库挖掘过程.开发者优先级识别是指根据开发者的贡献大小,确定开发者的优先级序列[7],辅助软件开发工作.Xuan 等人[7]首先以 Eclipse 和 Mozilla 的 bug 仓库为数据源,收集 2011 年之前的所有报告.然后,预处理每个 bug 报告,抽取报告中的标识、提交者、修复者、摘要、描述、创建时间以及评论信息,生成 2 个实验数据集.之后,在能够识别开发者优先级的领导力网络[8]的基础上进行改进,为所有开发者增加一个虚拟的开发者,并建立原始开发者和虚拟开发者间双向链接,提出一种新的领导力网络,能够识别基于组件和基于产品的开发者优先级.最后,将改进的网络应用于收集到的数据集,并调研 4 个研究问题来验证开发者优先级的有效性.

综上,软件仓库挖掘一般流程为:收集数据、预处理数据(特征提取)、寻找/改进/设计合适的数据挖掘算法、运用数据挖掘算法解决软件工程问题[6,9-10],如图 1 所示,其中软件工程数据(software engineering data)在软件仓库挖掘中起着关键作用.软件工程数据种类繁多,可以分为序列(如执行路径)、图(如程序依赖图)、文本(如 bug 报告、e-mail)[5].这些数据常常涉及 3 个因素,即人(people)、过程(processes)和产品(products),可以称为"3P"因素[5].人包括软件开发者、测试者、工程管理者和终端用户;过程包含软件活动的各个阶段,如软件测试、软件维护等;产品包括结构化产品(如代码)和非结构化产品(如文档).为了促进软件仓库挖掘领域的发展,2004 年第 1 届国际软件仓库挖掘研讨会(international workshop on mining software repositories, WMSR)在苏格兰首府爱丁堡举行,之后软件仓库挖掘在学术界和工业界受到了广泛的重视和研究.

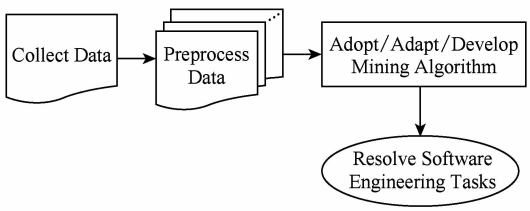

Fig. 1　The procedure of mining software repositories.

图 1　软件仓库挖掘流程

虽然 MSR 吸引了大量研究者,但现有的研究工作并未为这些研究者系统地呈现该领域的最有影响力的作者、研究机构、国家/地区,以及最热门的研究主题和主题变迁等领域知识.一些综述性的研究只是概括性地总结了 MSR 的背景、历史和值得研究的问题[9-10],并没有量化的方法来揭示 MSR 丰富的领域知识.随着专业知识的提高,研究者更希望对 MSR 领域进行深入挖掘,了解 MSR 论文作者间的合作关系,掌握 MSR 领域的研究主题动态变化趋势,从而合理地推断出未来的发展方向.WMSR 作

为 MSR 领域内一个重要的国际会议,在 MSR 领域有着很大的影响力,其收录的 MSR 相关论文无论是数量还是质量都具有很强的代表意义,研究 WMSR 上的文献信息能够帮助我们了解一些有价值的 MSR 领域知识.因此,本文主要工作是分析 WMSR 中文献信息,识别最高产的作者、机构、国家/地区、最频繁的关键词、最有影响力的作者和论文,并分析作者间的合作关系、热点研究主题以及作者的研究兴趣,帮助研究者深入了解 MSR 领域知识.在后续章节中出现的 MSR 文献分析特指 WMSR 文献分析,从而得出的结论主要适应于 WMSR 上收录的论文.

在本文中,我们采用文献分析技术[11-15].最初的文献分析研究通常借助数理统计方法来揭示某一领域的基本信息,包括论文、作者、机构、国家/组织[11-12].后来随着研究的深入,人们不再拘泥于简单的数据统计,而是采用数据挖掘等方法来分析文献内部蕴含的知识和关系,如特定主题论文分布情况、研究主题逐年变化趋势,以及作者之间的合作关系等,这些研究内容可以归结为基础文献分析(bibliography analysis)和合作模式分析(collaboration pattern analysis).长期的研究也形成了一套行之有效的文献分析框架和技术[14-16],其主要步骤为:确定数据源、收集数据、预处理数据、执行相关文献分析.各种算法和度量标准也被应用到文献分析领域,如 GN(Girvan-Newman)社区聚类算法[16]、文本处理技术、数据挖掘技术以及 APS(adjusted productivity score)指数[17]、ACS(adjusted citation score)指数[18]、NCII (normalized citation impact index)指数[19].调研显示,现有的框架和技术能被广泛地应用到不同领域的文献分析研究中.

为了实现 MSR 文献分析,我们构建了一个 MSR 文献分析框架(MSR publication analysis framework, MSR-PAF),该框架包含 3 个组件:1)数据收集组件,用来建立文献分析所需的数据集.我们首先从 WMSR 上收集已发表的论文标题,然后利用网络爬虫工具从 DBLP, IEEE Xplore, ACM 上爬取作者全名、机构、国家/地区、关键词、摘要等信息,最后从 Google Scholar 中抽取论文的引用次数.2)基础文献分析组件,通过实施产量分析和影响力分析,识别出最高产的作者、机构、国家/地区以及最频繁的关键词,同时找到最有影响力的作者和论文.3)合作模式分析组件,通过构建 3 个关系网络,即作者合著网络(co-authorship network)、关键词共现网络(co-occurrence keyword network)和作者-关键词共现网络(author co-keyword network),分别分析作者之间的合作关系、主要的研究主题以及作者的研究兴趣,并使用 NetDraw[20] 工具可视化这 3 个关系网络.基础文献分析结果显示最高产的作者、机构、国家/地区分别是 Ahmed E. Hassan, University of Victoria 和美国,最频繁的关键词是"software maintenance",最有影响力的作者和论文是 Ahmed E. Hassan 和 "When do changes induce fixes?".另外,合作模式分析结果显示 Abram Hindle 是 MSR 领域最活跃的作者,open source project 和 software maintenance 是最流行的研究主题.

本文的贡献有 3 点:

1) 为了实施 MSR 文献分析,我们构建了一个 MSR 文献分析框架,即 MSR-PAF,该框架包含 3 个组件,我们创建了一个完整的数据集用于 MSR 文献分析;

2) 在执行基础文献分析时,我们使用数理统计方法实施产量分析,同时引入 H 因子和 NCII 指数实施影响力分析;

3) 在执行合作模式分析时,我们生成 3 个关系网络,包含作者合著网络、关键词共现网络和作者-关键词共现网络,分析作者之间的合作关系、主要的研究主题以及作者的研究兴趣.

1 相关研究工作

本节详细讨论相关研究工作,主要包括 2 个领域:软件仓库挖掘和文献分析.

1.1 软件仓库挖掘

MSR 研究覆盖软件开发的各个阶段,包括需求、设计、实施、测试、调试、维护和部署,其涉及到的软件工程数据可以划分为 3 类[5]:

1) 序列.这类数据通常是软件在执行过程中动态生成的结构化信息,包含执行路径、co-change 等信息.比如,crash 报告系统能够自动地生成 crash 报告,这些报告通常包含系统执行过程中的调用栈信息.许多研究通过抽取调用栈信息来计算 crash 报告的相似度并自动地实现 crash 报告分桶(crash report bucketing)[21-22],还有一些研究通过挖掘调用栈信息来帮助开发者识别 crash 根源[23].

2) 图.这类数据往往能够直观形象地呈现软

件工件间的关系,包括动态/静态调用图、程序依赖图等.例如程序依赖图是一种带标签的有向图、模拟程序或过程语句之间的依赖关系.通过挖掘程序依赖图,可以提取程序内在关系,从而发掘隐藏的信息[24-25].

3) 文本.这类数据通常是人工撰写的非结构化信息,包括 bug 报告、e-mail、文档等.例如,测试者通过执行软件测试为软件的异常行为撰写 bug 报告,这些报告往往包含较多的自然语言信息,然而,人工检测大量的 bug 报告是一项十分繁重的任务.因此,为了减少人工检测代价,研究者提出了一种典型的文本挖掘任务,即 bug 报告重复检测(duplicate bug report detection)[26-29].许多研究利用常见的文本挖掘方法,如自然语言处理技术(natural language processing, NLP)[26]、信息检索技术(information retrieval, IR)[27]、主题模型(topic modeling)[28]或机器学习(machine learning)[29]抽取特征或者建立向量空间模型来计算文本相似度,从而实现重复检测.

实际上,MSR 文献分析研究也可以看作一种特殊的软件仓库挖掘任务,其使用的数据集是基于文本的.通过挖掘数据集中包含的信息来识别高产作者、机构、国家/地区,并发现最频繁的关键词、最有影响力的作者和论文,同时,分析 MSR 领域作者间的合作关系、主要的研究主题以及作者的研究兴趣.

1.2　文献分析

文献分析(publication analysis)主要是采用数理统计和数据挖掘等方法对某个特定领域的文献进行深入地挖掘,使该领域的研究者能够系统地了解这个领域的研究背景、历史和现状,明确该领域内最流行的研究主题和方向[14].传统的文献分析通常简单地统计文献的基本信息,如论文标题、作者、机构、国家/地区、关键词等.大量的文献分析研究聚集在智能交通领域,Wang[11]简单统计了 2000 年至 2009 年发表在 T-ITS (IEEE Transaction on Intelligent Transportation System)期刊上的文献.Li 等人[12]收集了 T-ITS 上 10 年的文献,并通过产量分析识别出该领域最高产的作者、机构、国家/和地区.近年来,随着研究的深入,文献分析的内容不断扩充,延伸到影响力分析、社会网络分析、聚类分析、文章话题分析等各个方面,因此一些典型的数据挖掘方法也被引入到文献分析研究中.Tang 等人[30]收集了 T-ITS 上 2010 年至 2013 年出版的所有文献,并对该领域的研究主题分类,识别出 5 个热点研究主题.Xu 等人[16]收集了该期刊上所有的论文,并执行了全面的基础文献分析和合作模式分析,他们引入了 GN 聚类算法和 3 个关系网络对作者合作模式以及主题变迁进行深入分析.在推荐系统领域,Park 等人[14]利用一些重要的关键词搜索几个主要数据库,从 31 个期刊中精心挑选出 164 篇论文,划分为 8 类,并使用数据挖掘技术检测这些论文,识别出推荐系统领域内流行的研究主题.在云计算领域,Heilig 等人[13]从 Elsevier 数据库中收集了总计 15 376 篇论文,这些论文发表于 2008 年至 2013 年,他们主要执行了产量分析、影响力分析以及研究主题分析.

与上述任务类似,我们的工作是分析 MSR 文献信息,挖掘 MSR 领域知识.我们收集 WMSR 上文献并执行文献分析,主要分为基础文献分析和合作模式分析.

2　MSR 文献分析框架

本节详细阐述 MSR 文献分析框架,由 3 个组件组成,如图 2 所示,包括一个数据收集组件、一个基础文献分析组件和一个合作模式分析组件.数据收集组件用来创建我们研究所需要的数据集;基础文献分析组件针对论文中的单一类别的信息执行统计分析,从而识别最高产的作者、机构、国家/地区和最频繁的关键词,并分析作者和论文的影响力,主要包括产量分析和影响力分析;合作模式分析组件针对多种信息的关联关系来挖掘隐藏的知识,通过构造 3 个关系网络来研究作者间的合作关系、主要的研究主题以及作者的研究兴趣.

2.1　数据收集

为了实现 MSR 文献分析,我们需要建立一个完备的数据集.我们选取 WMSR 作为我们的数据源,并收集 2004 年至 2016 年所有发表在 WMSR 上的论文标题.在我们的研究中,主要包括基础文献分析和合作模式分析.基础文献分析又包括产量分析和影响力分析,产量分析涉及到的信息包括作者、机构、国家/地区、关键词;影响力分析涉及到的关键信息是论文的引用次数.合作模式分析研究作者间的合作关系、主要的研究主题以及作者的研究兴趣、涉及到的信息包括每篇论文的所有作者以及关键词.通过仔细调研,发现有些论文并没有提供关键词信息,因此我们试图从摘要和标题中抽取一些主题词来补充关键词.我们采用关键词抽取模型[31],其过程有 3 个步骤:

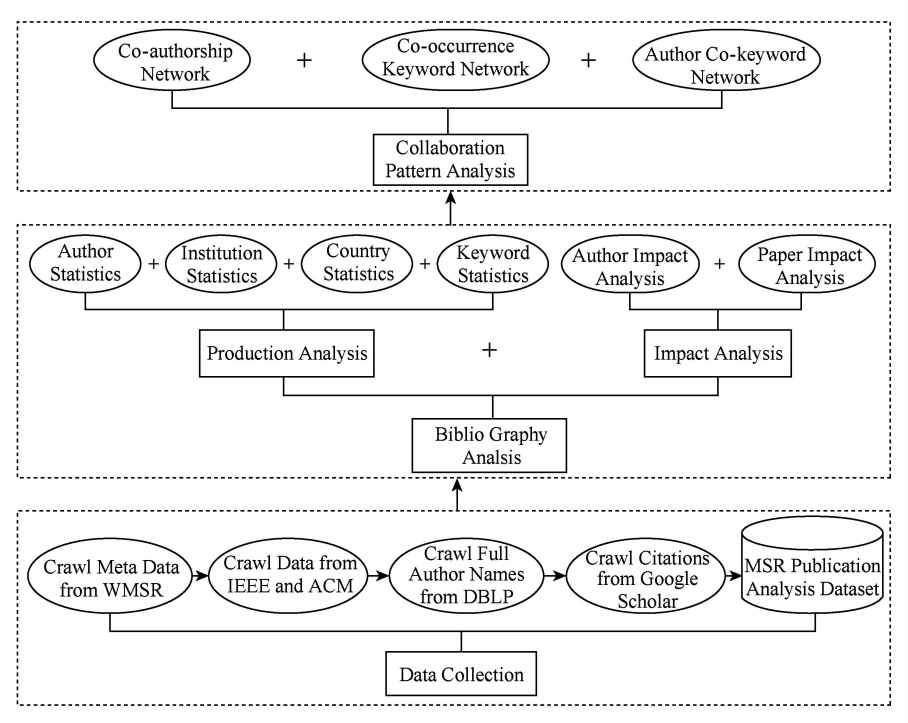

Fig. 2 The MSR publication analysis framework.

图 2 MSR 文献分析框架

1) 移除停用词. 对于一些如 the, is, we 等对关键词抽取来说毫无意义的词,我们建立一个停用词表[32],从摘要和标题中删除这些词.

2) 对剩下的词分别建立 x-元词(x 为单词个数,取值为 1～4)权重矩阵,权重的值为单词或术语在标题和摘要中出现的次数.

3) 对所有的 x-元词按权重进行降序排序,然后取权重最高的 $n(n\leqslant 10)$个词作为关键词.

综上,我们需要收集的信息包括标题、作者、机构、国家/地区、关键词、摘要、论文引用次数. 数据集的建立过程包含 4 个步骤:

1) 收集 2004 年至 2016 年所有发表在 WMSR 上的论文的标题,作为数据集的元数据.

2) 利用网络爬虫工具从 IEEE Xplore 和 ACM 数据库中抽取一些重要的信息,包括作者、机构、国家/地区、摘要、关键词.

3) 考虑到 IEEE Xplore 和 ACM 数据库中提供的作者姓名通常是缩写,因此我们利用网络爬虫工具从 DBLP 中自动抽取作者的全名.

4) 利用网络爬虫工具从 Google Scholar 中抽取论文的引用次数.

通过以上这 4 个步骤,我们收集了 MSR 文献分析所需的相关数据,并构建了一个完整的数据集. 该数据集包含不同类型的数据,呈现复杂而多相的特点.

2.2 基础文献分析

基础文献分析包括 2 个方面,即产量分析和影响力分析,主要针对单一类别的信息,采用统计分析方法来挖掘 MSR 基本的领域知识,如图 2 所示. 本节详细介绍产量分析和影响力分析的实施方法.

2.2.1 产量分析实施方法

产量分析主要是识别最高产的作者、机构、国家/地区以及最频繁的关键词. 在产量分析中,最简单直接的方法就是数理统计. 例如,为了找到最高产的作者,首先统计每个作者在 WMSR 上发表的论文的数量,然后使用排序方法找到最高产的作者.

2.2.2 影响力分析实施方法

1) 作者影响力分析. 在作者影响力分析中,我们引入 H 因子(H factor)[33]来度量单个作者的影响力.

H 因子:又称为 H 指数,是 Hirsch[33]于 2005 年提出的一种衡量作者影响力的指标,其综合考虑了作者发表的论文的质量和数量. 对于一些作者,虽然发表的论文数量较多,然而论文的质量并不高,即所有论文的引用数量都较低. 因此,H 因子综合考虑论文的质量和数量,其主要思想为:如果一个作者发表了 h 篇论文,其被引次数不得少于 h 次. 具体过程为:对某个作者在某个时段内发表的论文,按被引

次数从高到低排列,排序后每篇论文会得到一个序号 i,将每篇论文的序号 i 和被引次数进行比较,找到序号 h 的论文,使得该论文的序号 h 小于或等于它的被引次数,而下一篇论文,其序号 $h+1$ 大于它的被引次数.

H 因子已经被广泛接受并用于衡量不同领域作者的影响力.例如,Alcaide 等人[34]通过 H 因子来评估生物医学中 20 个主要作者的科学研究的影响力;Oppenheim[35]使用 H 因子对信息领域的科学家进行排序;Bornmann 和 Daniel[36]也应用 H 因子到博士后奖学金申请人的评选工作中.在文献[37]中,Alonso 等人对 H 因子的优点、缺点、应用以及各种改进版本进行了系统地总结.很多研究者的 H 因子能在 Google Scholar 中查询到,在本文我们并不直接使用 Google Scholar 中的 H 因子,因为其衡量的是作者在所有研究领域的影响力.我们需要计算所有作者在 MSR 领域的 H 因子,然后根据 H 因子对作者排序.

2) 论文影响力分析.在论文影响力分析中,我们引入 NCII 指数[19]来度量论文的影响力.

NCII 指数:通常情况下,论文的引用次数与其发表的时间有着很大的关系,也就是说,一篇论文发表的时间越早,其被引用的次数可能越多.从而导致不同时期出版的论文难以比较它们的影响力.因此,考虑到出版时间对引用数量的影响,Holsapple 等人[19]提出了一个新的影响力计算标准,即 NCII 指数.其计算为

$$NCII=\frac{\text{论文的引用次数}}{\text{论文发表的时间长度}},\tag{1}$$

从式(1)可以看出,NCII 指数实际上代表了论文每年的平均引用次数.相比较于总的引用次数,使用 NCII 指数作为论文影响力评价标准更加合理.目前,NCII 指数已被广泛用于评估领域科研论文的影响力.例如,Serenko 和 Bontis[38]利用 NCII 指数来计算知识管理和智能资本相关文献的影响力;在智能交通领域,Xu 等人[16]使用 NCII 指数对该领域的文章进行影响力排序;另外,基于 NCII 指数的思想,Cheng 等人[39]提出了类似的标准化评分(normalized score),对人工智能领域的 1224 个期刊杂志的影响力进行了排序.在本文我们首先计算出每篇论文 NCII 指数;然后根据 NCII 指数排序,分析论文的影响力.

2.3　合作模式分析

合作模式分析研究作者间的合作关系、MSR 领域主要研究主题以及作者的研究兴趣.通过分析信息之间的相互联系,挖掘 MSR 领域中一些隐藏的领域知识.为了完成这些关键问题的分析,我们构建 3 个重要的关系网络,即作者合著网络、关键词共现网络、作者-关键词共现网络.其中作者合著网络与关键词共现网络相互独立,分别基于作者间的依赖关系和关键词间的依赖关系,揭露作者间的合作关系以及流行的研究主题;而作者-关键词共现网络基于作者和关键词间的依赖关系,揭露作者研究兴趣.本节阐述合作模式分析的详细过程.

2.3.1　GN 聚类算法

GN 是一种经典的社区发现算法,属于分裂的层次聚类算法[40].基本思想是不断地删除网络中具有相对于源节点的最大边介数(edge betweenness)(一条边的边介数是指通过该边的最短路径的条数)的边,再重新计算网络中剩余的边相对于源节点的边介数,直到所有边被消除.然而,在不知道社区数目的情况下,GN 算法无法确定选取哪种网络状态.因此,Clauset[41]引入了模块度的概念,提出了一种改进的 GN 算法.其基本步骤如下:

1) 计算网络中所有边的边介数;

2) 找到边介数最高的边并将该边从网络中删除掉,记录新网络状态下的模块度和网络状态;

3) 重复步骤 1 和步骤 2,直到每个节点就是一个退化的社区为止,最后把模块度最大的状态作为分裂的结果.

模块度(modularity)Q 是一种评价社区划分质量的标准[32],其计算公式为

$$Q=\sum_i(e_{ii}-a_i^2),\tag{2}$$

其中,e_{ii} 表示网络中第 i 个社区中连接 2 个不同节点的边在所有边中所占的比例,a_i 表示与第 i 个社区中的节点相连的边在所有边中所占的比例.

2.3.2　作者合著网络

在一篇论文中,可能存在多个作者,这些作者相互合作共同完成论文的撰写.同一作者可能与不同的作者合作,具有不同的合作关系.作者合著网络使用 GN 算法对作者进行聚类,揭示作者间紧密的合作关系.

定义 1. 给定一个关系网络 $N=\{A,B,W\}$.其中,A 代表点的集合,即作者集合;B 代表边的集合,即作者间的合作关系集合,B 中的每一个元素 b_{xy} 表

示作者 a_x 和作者 a_y 共同完成了一篇论文;W 表示权重集合,即作者之间的合作次数集合,其值是 2 个作者合作完成的论文数量.

作者合著网络分析. 对该网络实行 $w(A')$ 操作,并使得模块度 Q 的值最大. 标志 $w(A')(\forall A' \subseteq A)$ 指找到一个划分 $A'_1, A'_2, \cdots, A'_k$,其中 A'_j 代表一个社区的作者集合,A'_j 中的每个作者 $a_i(a_i \in A)$ 至少与另一个作者存在合著关系,k 为社区的个数,且需要满足 2 个条件:

$$A = A'_1 \cup A'_2 \cup \cdots \cup A'_k, \tag{3}$$

$$A'_m \cap A'_n = \varnothing, m \neq n. \tag{4}$$

2.3.3 关键词共现网络

一般情况下,论文会提供一些关键词来表明其核心研究主题,当几个关键词出现在同一篇论文中,意味着这些关键词有着一定的相关性. 关键词共现网络使用 GN 算法对关键词聚类,找到网络中流行的研究主题.

定义 2. 给定一个关系网络 $N=\{K,B,W\}$. 其中,K 是点的集合,即关键词集合;B 是边的集合,代表关键词之间共现关系,B 中的每一个元素 b_{xy} 表示关键词 k_x 和关键词 k_y 之间的共现关系;W 表示权重集合,其值为同时出现这 2 个关键词的论文的数量.

关键词共现网络分析. 对该网络实行 $w(K')$ 操作,并使得模块度 Q 的值最大. 标志 $w(K')$ 的定义与 $w(A')$ 相同.

2.3.4 作者-关键词共现网络

通常,作者完成 1 篇论文时都会使用一些关键词来表明该论文的研究主题,当 2 篇论文的作者使用相似或相同的关键词时,意味着这些作者之间可能有着相近或相同的研究兴趣. 作者-关键词共现网络使用 GN 算法对作者聚类,每一类中的作者都有着相近或相同的研究兴趣.

定义 3. 给定一个关系网络 $N=\{A,AK,T\}$. 其中,A 是点的集合,即作者集合;AK 是边的集合,每一个元素 ak_{xy} 表示作者 a_x 和作者 a_y 使用过相同的关键词;T 表示权重集合,其值为 2 个作者使用相同关键词的数目.

作者-关键词共现网络分析. 对该网络实行 $w(A')$ 操作,并使得模块度 Q 的值最大.

2.3.5 关系网络分析过程

关系网络分析主要是借助 GN 聚类算法对网络中的节点聚类,将网络的节点划分到不同的簇. 然而,上述 3 个关系网络都是带有权重的网络,传统的 GN 聚类算法不能直接应用于这 3 个网络. 因此,Xu 等人[16]定义了一个新的概念,即边值(edge value),其值等于边介数除以权重. GN 算法通过不断地删除边值最大的边,来寻找模块度最大的网络状态. 使用 GN 算法聚类以后,在对每个簇评价时需要使用到一种指标,即平均节点度 A_d(average degree)[16]. 下面,我们详细介绍这个指标:

平均节点度是社会网络中某个点所连接的边的权重的平均值[16]. 以作者合著网络为例,平均节点度是作者平均合作次数. 假设存在一个子网络 $N'=\{A',B',W'\}$,A', B', W' 为 A, B, W 的子集,则有:

$$A_d = \frac{|B'|}{|A'|}. \tag{5}$$

3 基础文献分析结果

本节主要介绍 MSR 文献分析数据集,并从 2 个方面即产量分析和影响力分析来呈现基础文献分析结果.

3.1 数据集

我们数据集的数据来源于 2004 年至 2016 年 WMSR 收录的所有论文. 为了创建 MSR 文献分析数据集,我们首先从 WMSR 上抽取论文标题作为元数据,然后通过网络爬虫工具从 DBLP, ACM, IEEE Xplore, Google Scholar 中抽取作者全名、机构、国家/地区、关键词、摘要以及引用次数. 该数据集包含 529 篇论文和 961 位作者,这些作者来自 35 个国家/地区,隶属于 254 个不同的机构. 图 3 显示了 WMSR 每年收录的论文的数量,从图 3 中可以看出,在 2012 年(含)之前,每年 WMSR 收录的文章数量都在 40 篇以下,从 2013 年开始文章数量有所增加,这表明近年来更多的学者开始关注 MSR 领域.

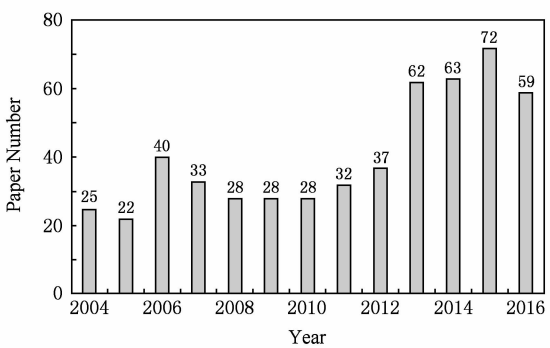

Fig. 3 The number of publications per year in WMSR.

图 3 WMSR 每年文章数量

3.2　产量分析结果

本节呈现产量分析结果,即识别最高产的作者、机构、国家/地区以及最频繁的关键词.

3.2.1　作者统计

我们的数据集中收集了所有论文的作者全名,为了识别最高产的作者,需要对作者信息进行预处理:

首先,由于一些特殊字符,需要统一作者全名,如"Yann-Gaël Guéhéneuc"和"Yann-Gael Gueheneuc"应该表示同一个作者,我们用后者代替前者;然后,去掉重复的作者,并统计每个作者发表的论文数量.

我们的数据集中包含 529 篇论文,大多数论文的作者数量为 1～6 位,极少数作者数量超过 7 位,平均作者数量为 3.29 位,共涉及 961 位不同的作者,其中 674 位作者仅发表了 1 篇论文.表 1 显示了最高产的 10 位作者以及他们发表的论文数量.从表 1 中可以看出,排名第 1 的作者是 Ahmed E. Hassan,在 WMSR 上共发表了 23 篇论文;排在第 2 位和第 3 位的是 Abram Hindle 和 Daniel M. German,在 WMSR 上均发表了 22 篇论文;其余 7 位作者在 WMSR 上发表的论文数量都超过 10 篇.

Table 1 The Information of the Most Productive Authors

表 1　高产作者信息

Ranking	Author	Paper Number
1	Ahmed E. Hassan	23
2	Abram Hindle	22
3	Daniel M. German	22
4	Bram Adams	18
5	Gregorio Robles	17
6	Jesus M. Gonzalez-Barahona	16
7	Christian Bird	15
8	Georgios Gousios	12
9	Michele Lanza	11
10	Thomas Zimmermann	11

3.2.2　机构统计

为了识别最高产的机构,我们需要对机构信息进行预处理:

1) 在数据集中,每一篇论文的每一个作者都对应一个机构,必然存在多个作者来自于同一个机构.因此同一篇论文中,同一机构仅统计一次.

2) 在不同的论文中,由于不同作者的表达方式或者写作习惯不同,同一机构可能有不同的名称.因此,需要人工统一机构的名称,比如 Ecole Polytechnique de Montréal 和 Polytechnique de Montréal 实际上表示同一机构.

3) 部分大学包含多个分校,比如加州大学(University of California)包含 10 个分校,这些分校间相互独立,即不共享研究成果.因此,需要区分这些分校.

4) 一些公司或企业的研究机构也会参与科学研究,这些研究机构可能分布在不同的国家/地区,但共享研究成果.比如,IBM Watson Research Lab 和 IBM Haifa Research Lab 分别位于美国和以色列.因此,我们不区分这些机构,即统一使用公司名称.

通过以上 4 个步骤,我们发现共有 254 个不同的组织或机构,其中 135 个机构仅发表了 1 篇论文.表 2 列出了前 10 的机构的名称、发表的论文数量以及它们所属的国家/地区.从表 2 中可以看出,排名前 3 的大学是加拿大的 University of Victoria, Queen's University, University of Waterloo,分别发表了 32,31,29 篇论文,其他 7 所大学所发表的论文都超过 10 篇.在前 10 的大学中有 5 所大学位于加拿大,另外 5 所大学分别位于荷兰、美国、西班牙、瑞士和德国.可见,隶属加拿大的研究机构对 MSR 领域的发展有着一定的贡献.

Table 2　The Information of the Most Productive Institutions

表 2　高产机构信息

Ranking	Institution	Paper Number	Countries
1	University of Victoria	32	Canada
2	Queen's University	31	Canada
3	University of Waterloo	29	Canada
4	Delft University of Technology	25	The Netherlands
5	University of California at Davis	21	USA
6	Universidad Rey Juan Carlos	19	Spain
7	University of Alberta	19	Canada
8	Ecole Polytechnique de Montréal	18	Canada
9	University of Zurich	16	Switzerland
10	Saarland University	15	Germany

3.2.3　国家/地区统计

为了识别最高产的国家/地区,我们需要对数据集中的国家信息进行预处理:

1) 极少数论文中虽然提供了机构信息,然而缺失国家信息.因此,我们需要仔细核对这些机构所属国家/地区.

2) 一些公司或企业的研究机构可能位于不同的国家/地区.因此,我们需要区分这些国家/地区.

例如 IBM Watson Research Lab 和 IBM Haifa Research Lab 分别位于在美国和以色列,其研究成果应该区分美国和以色列.

通过上述预处理的 2 个步骤,我们统计出该数据集中包含 35 个国家/地区,其中有 9 个国家/地区仅发表了 1 篇论文,按照发表的论文数量对这些国家/地区进行排序,图 4 显示这些国家/地区发表论文的数量信息.从图 4 中可以看出,最高产的 10 个国家分别是美国、加拿大、荷兰、德国、瑞士、日本、英国、意大利、西班牙和法国;美国和加拿大分别发表了 174 和 146 篇论文,与机构统计结果相比,美国才是最高产的国家.这是因为美国有着更多的机构参与了 MSR 领域研究,而在加拿大,仅有几所大学参与 MSR 领域研究.另外,观察发现美国和加拿大发表的论文数量占总数量一半以上,主导着 MSR 领域的发展.中国作者参与了 13 篇论文的撰写,排名为 11.

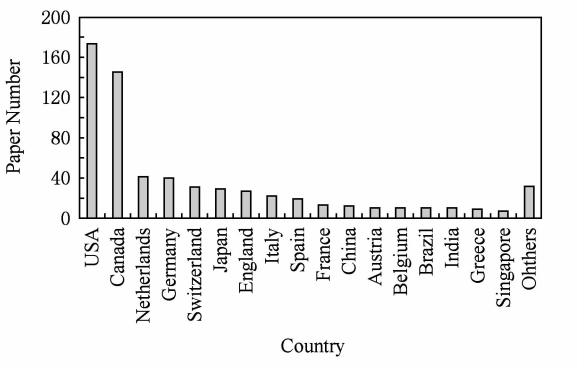

Fig. 4 The publication numbers of different countries.

图 4 各国发表的 MSR 论文数量

3.2.4 关键词统计

为了识别最频繁的关键词,我们需要对数据集中的关键词信息进行预处理:

1) 同样关键词中某个单词可能是复数也可能是单数.因此我们将复数变成单数,但仅考虑将结尾为"s"和"ies"的词转化为原型.

2) 在关键词中,存在一些对关键词统计毫无意义词,如 software engineering, mining software repositories, data mining 等,我们收集这些关键词并放入停用词列表[32],然后自动移除这些关键词.

3) 不同作者有着不同的写作习惯和表达方式,他们会使用不同的关键词来表示相同的主题,比如 bug, defect, fault 等.因此,我们建立了一个同义词表[32],将不同的词替换为同一个词,如将 defect 和 fault 替换为 bug.

通过以上 3 个步骤,我们统计出所有的不同的关键词,并计算每个关键词的频率,根据它们的频率排序,表 3 显示了前 10 个最频繁的关键词.从表 3 中可以看出,"software maintenance"是最频繁的关键词,109 篇论文使用过该关键词,这说明在软件仓库挖掘领域,软件维护是最主要研究方向.其原因是软件仓库挖掘所涉及到的数据大部分源于软件维护阶段;排在第 2 和第 3 的是"Open source project"和"Software configuration management and version control system",频率分别为 87 次和 58 次.这 2 个关键词获得较高排名的原因是从开源工程获取数据最为容易,而软件版本演化是重要的研究主题.其他的频繁的关键词包括"Software post development issue", "Java", "Documentation", "Software quality","Human factor", "Performance", "Public domain software".

Table 3 The Information of the Most Frequent Keywords

表 3 最频繁的关键词的信息

Ranking	Keyword	Frequency
1	Software Maintenance	109
2	Open Source Project	87
3	Software Configuration Management and Version Control System	58
4	Software Post Development Issue	52
5	Java	41
6	Documentation	39
7	Software Quality	37
8	Human Factor	33
9	Performance	32
10	Public Domain Software	32

3.3 影响力分析

本节呈现影响力分析结果,主要分为作者影响力分析和论文影响力分析 2 个方面.

3.3.1 作者影响力分析

MSR 文献分析数据集收集了所有论文的引用次数,我们根据引用次数计算所有作者的 H 因子,然后根据 H 因子对作者进行排序,表 4 记录了前 10 位作者的信息.从表 4 中可以看出,排名前 3 的作者是 Ahmed E. Hassan, Daniel M. German, Abram Hindle,H 因子分别为 14,14,11,他们均来自加拿大,所在的机构也是高产机构;其他 7 位作者均来自加拿大、西班牙、荷兰等高产国家,可见,来自高产国家/地区的作者往往有着较大的影响力.

Table 4　The Information of 10 Authors with the Highest H Factor

表 4　H 因子最高的前 10 位作者的信息

Ranking	Author	H Factor	Institution
1	Ahmed E. Hassan	14	Queen's University
2	Daniel M. German	14	University of Victoria
3	Abram Hindle	11	University of Alberta
4	Gregorio Robles	11	Universidad Rey Juan Carlos
5	Bram Adams	10	Ecole Polytechnique de Montreal
6	Christian Bird	10	Microsoft Research
7	Jesus M. Gonzalez-Barahona	10	Universidad Rey Juan Carlos
8	Thomas Zimmermann	9	Microsoft Research
9	Georgios Gousios	8	Radboud University Nijmegen
10	Michael W. Godfrey	8	University of Waterloo

3.3.2　论文影响力分析

我们收集了每篇论文的引用次数,表 5 呈现了引用次数最高的 10 篇论文的标题、引用次数、作者、国家和年份信息.从表 5 中可以看出,引用次数最高的论文大多发表于 2004 年至 2007 年,仅有 2 篇论文分别发表于 2009 年和 2010 年.可见,引用次数和发表年份有着很大的关系.排名前 3 的论文分别被引用了 489,442,253 次,其他 7 篇论文的引用次数均在 100 次以上.这些高引论文的作者大多数来自德国、美国和瑞士.很明显,高产国家参与 MSR 研究更早,所发表的论文引用次数自然更高.

我们计算所有论文的 NCII 指数,然后根据 NCII 指数对论文进行排名.表 6 呈现了 NCII 指数最高的前 10 篇论文标题、引用次数、NCII 指数、作者、国家和年份信息.排名前 10 的论文中发表于 2013 年和 2014 年各有 2 篇,其他发表于 2005 年、2006 年、2007 年、2009 年、2010 年、2012 年各有 1 篇.NCII 指数最高的 3 篇论文分别是"When do changes induce fixes?", "Mining email social networks", "The promises and perils of mining GitHub",其值均超过 40.实际上,NCII 指数最高的论文"When do changes induce fixes?"也有着最高的引用次数.

Table 5　The Information of the 10 Most Cited Publications

表 5　最高引的 10 篇论文的信息

Ranking	Title	Citation	Author	Country	Year
1	When do changes induce fixes?	489	Jacek Sliwerski, Thomas Zimmermann, Andreas Zeller	Germany	2005
2	Mining email social networks	442	Christian Bird, Alex Gourley, Premkumar T. Devanbu, Michael Gertz, Anand Swaminathan	USA	2006
3	Preprocessing CVS Data for Fine-Grained Analysis	253	Thomas Zimmermann, Peter Weißgerber	Germany	2004
4	How Long Will It Take to Fix This Bug?	243	Cathrin Weiss, Rahul Premraj, Thomas Zimmermann, Andreas Zeller	Germany	2007
5	An extensive comparison of bug prediction approaches	198	Marco D'Ambros, Michele Lanza, Romain Robbes	Switzerland	2010
6	The perils and pitfalls of mining SourceForge	194	James Howison, Kevin Crowston	USA	2004
7	MAPO: mining API usages from open source repositories	173	Tao Xie, Jian Pei	USA	2006
8	The promises and perils of mining git	162	Christian Bird, Peter C. Rigby, Earl T. Barr, David J. Hamilton, Daniel M. German, Prem Devanbu	USA	2009
9	Understanding source code evolution using abstract syntax tree matching	156	Iulian Neamtiu, Jeffrey S. Foster, Michael Hicks	USA	2005
10	Applying Social Network Analysis to the Information in CVS Repositories	151	Luis Lopez-Fernandez, Gregorio Robles, Jesús M. González-Barahona	Spain	2004

Table 6 The Information of 10 Publications with the Highest NCII

表 6 NCII 指数最高的 10 篇论文的信息

Ranking	Title	Citation	NCII	Author	Country	Year
1	When do changes induce fixes?	489	44.45	Jacek Sliwerski, Thomas Zimmermann, Andreas Zeller	Germany	2005
2	Mining email social networks	442	44.2	Christian Bird, Alex Gourley, Premkumar T. Devanbu, Michael Gertz, Anand Swaminathan	USA	2006
3	The promises and perils of mining GitHub	85	42.5	Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. Germán, Daniela Damian	Canada	2014
4	The GHTorent dataset and tool suite	111	37	Georgios Gousios	Netherlands	2013
5	An extensive comparison of bug prediction approaches	198	33	Marco D'Ambros, Michele Lanza, Romain Robbes	Switzerland	2010
6	How Long Will It Take to Fix This Bug?	243	27	Cathrin Weiß, Rahul Premraj, Thomas Zimmermann, Andreas Zeller	Germany	2007
7	The impact of code review coverage and code review participation on software quality: a case study of the qt, VTK, and ITK projects	51	25.5	Shane McIntosh, Yasutaka Kamei, Bram Adams, Ahmed E. Hassan	Canada	2014
8	App store mining and analysis: MSR for app stores	97	24.25	Mark Harman, Yue Jia, Yuanyuan Zhang	UK	2012
9	The impact of tangled code changes	72	24	Kim Herzig, Andreas Zeller	UK	2013
10	The promises and perils of mining git	162	23.14	Christian Bird, Peter C. Rigby, Earl T. Barr, David J. Hamilton, Daniel M. Germán, Premkumar T. Devanbu	USA	2009

另外,这些论文的作者基本上来自于德国、美国、加拿大、荷兰、英国、日本等一些高产国家.可见,高产国家的论文有着较大的影响力.实际上,NCII 指数平衡了论文的引用次数和发表时间关系,即论文发表的时间越早,并不代表论文的影响力就越高,在一定程度上能更准确地反映出论文的影响力.

4 合作模式分析结果

本节主要通过 3 个关系网络,包括作者合著网络、关键词共现网络、作者-关键词共现网络来呈现合作模式分析结果.

4.1 作者合著网络

一个人对某个领域的影响力大小,与他和该领域其他作者合作次数有着很大的关系.另外,影响力越大的作者,对整个领域的贡献也就越大,与他合作的作者就可能越多.通过使用 GN 算法对作者合著网络中的节点(即作者)聚类,分析作者之间的合作关系.在我们的研究中,为了深入地挖掘作者间的合作关系,仅考虑那些发表论文数量超过 2 篇的作者.在构建的作者合著网络中,共包含 404 个节点和 2 536 条边.在这个网络中,由于一些作者与另外一些作者可能没有合作关系,因此该网络通常由一些连通子图(社区)组成.我们借助 NetDraw 工具[20]可视化作者合著网络中最大的 2 个连通子图.

第 1 个连通子图的模块度为 0.719,平均节点数是 6.173 6,如图 5 所示.在这个连通块中共包含 288 个节点、1 778 条边,分属 16 个簇.由于每个作者的合作次数不同,我们用不同大小的点来区分合作次数的多少,用不同的颜色来区分这些簇.在这个连通块中,拥有合作次数最多的作者是 Ahmed E. Hassan,共参与合作 35 次,其所在簇的作者大多数来自加拿大;排名第 2 的是 Bram Adams,其参与合作的次数为 34 次;接着是 Christian Bird,合作次数为 32 次.

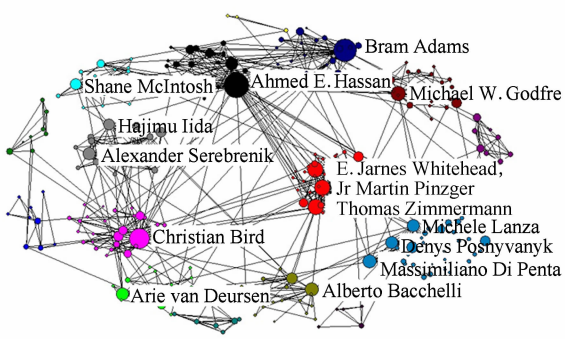

Fig. 5 The largest connected subgraph.

图 5 最大的连通子图

第 2 个连通子图的模块度为 0.649,平均节点数是 6.903,如图 6 所示.第 2 个连通块中作者的平均合作次数要高于第 1 个连通块的作者平均合作次数.在这个连通块中共包含 93 个节点、642 条边,分属 8 个簇.在这个连通块中,合作次数最多的作者是 Abram Hindle,达到 60 次,也是 MSR 领域参与合作次数最多的作者,即最活跃的作者;接着是 Daniel M. German 和 Katsuro Inoue,参与合作的次数分别达到 50 次和 28 次;另外,Jesus M. Gonzalez-Barahona 也有着较多的合作次数.

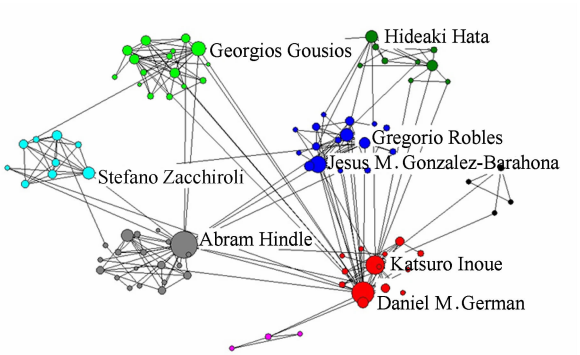

Fig. 6　The second largest connected subgraph.

图 6　第 2 大连通子图

4.2　关键词共现网络

在 1 篇已发表的论文中,一般会提供 3～5 个关键词作为标签,标注该论文的研究主题.同时出现在一篇论文中的关键词可能围绕着同样的研究主题.通过使用 GN 算法对关键词共现网络中的节点(即关键词)聚类来分析 MSR 领域中热点研究主题.为了准确地分析出主要的研究主题,我们考虑移除那些在论文中出现次数少于 5 次的关键词.这是因为当出现次数较小时,该关键词所代表的研究主题可能不是热点研究主题.在生成的关键词共现网络中,包含 93 个节点、340 条边,每 2 个节点之间的边表示 2 个关键词在不同论文中出现的次数总计超过 5 次.我们借助 NetDraw 工具[20]可视化关键词共现网络.

图 7 是生成的关键词共现网络,其模块度为 0.473,所有的关键词被划分为 12 个簇,分别用红色数字标出.划分在同一簇中的关键词有着一定的关系,比如“information retrieval”和“duplicate bug report detection”被聚集在簇 11 中,这是因为信息检索技术是解决重复 bug 检测的一个重要方法.“data acquisition”和“data analysis”被聚集在簇 6 中,这 2 个主题分别表示数据采集和数据分析,有着很强的相关性.我们把节点数最高的节点作为主题词,每个簇代表了一个研究主题.通过聚类我们发现,最大的 4 个簇,即簇 1,2,3,4 分别围绕“open source project”,“software maintenance”,“performance,test”,“documentation”四个主题词.这些主题词是 MSR 领域最热门的研究主题,同时,也是软件工程领域最常见的研究主题.因此,关键词共现网络能够解析出 MSR 领域热门研究主题.

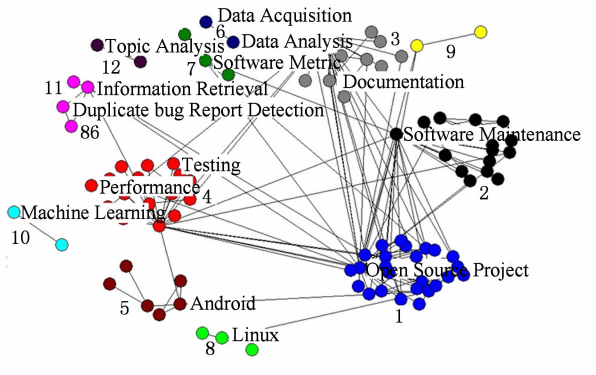

Fig. 7　The keyword co-occurrence network.

图 7　关键词共现网络

4.3　作者-关键词共现网络

在 2 篇不同的论文中,可能使用相同的关键词来描述论文的研究主题,这些相同的关键词表明这些作者可能具有相同或相近的研究方向.通过使用 GN 算法对作者-关键词共现网络中的点(即作者)聚类,来分析哪些作者有着相同的研究兴趣.在我们的研究中,为了重点研究一些具有代表性的作者的研究兴趣,我们过滤掉那些发表的论文数量少于 2 篇的作者,实际上,这些作者在 MSR 领域并不具有突出贡献.生成的网络包含 126 个作者、340 条边.我们借助 NetDraw 工具[20]可视化作者-关键词共现网络.

图 8 是作者-关键词共现网络,其模块度为 0.837,所有节点共被划分为 30 个簇,大多数簇中的节点都较少.我们详细分析其中最大的 3 个簇:

簇 1.这个簇是作者-关键词共现网络中最大的

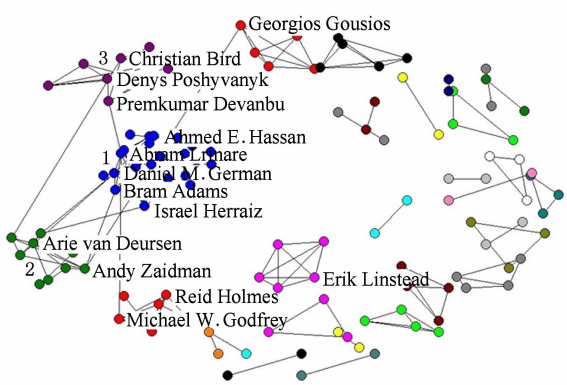

Fig. 8　The author co-keyword network.

图 8　作者-关键词共现网络

簇,以 Ahmed E. Hassan 和 Abram Hindle 为主导,包括 Daniel M. German, Bram Adams, Israel Herraiz 等作者,实际上,在这个簇中的作者大多数都是高产作者.主要关注软件维护领域,包括软件演化、代码推荐等方向.

簇 2. 这个簇也包含较多的作者,主要以 Andy Zaidman 和 Arie van Deursen 为主导.主要关注软件测试和软件开发领域,比如测试实例自动化生成和基于拉式的软件开发等方向.

簇 3. 这个簇也有一定数量的作者,主要以 Christian Bird, Denys Poshyvanyk, Premkumar Devanbu 为主导.其主要的研究方向是开源工程和社交网络等.

5 结束语

本文主要工作是 MSR 文献分析研究,分为基础文献分析和合作模式分析.为了高效地完成这项工作,我们建立了 MSR 文献分析框架,即 MSR-PAF.MSR 文献分析框架由 3 个组件组成:

1) 第 1 个组件用来创建数据集.我们收集 WMSR 上的所有文献标题作为元数据,从 IEEE Xplore 和 ACM 数据库中爬取作者、机构、国家/地区、关键词、摘要等信息,然后从 DBLP 中爬取作者的全名,最后从 Google Scholar 中爬取论文的引用次数,最终创建 MSR 文献分析数据集.

2) 第 2 个组件执行基础文献分析,我们使用数理统计方法识别最高产的作者、机构、国家/地区、最频繁的关键词,同时引入 H 因子和 NCII 指数来检测最有影响力的作者和论文.

3) 第 3 个组件实施合作模式分析,我们利用 3 个关系网络,包括作者合著网络、关键词共现网络、作者-关键词共现网络来分析作者之间的合作关系、主要研究主题以及作者的研究兴趣.文献分析结果表明 Ahmed E. Hassan 是最高产的作者,open source project 和 software maintenance 是最流行的研究主题.将来,我们会更多地关注 MSR 文献分析研究,扩展 MSR 文献分析数据源,更加深入地挖掘 MSR 文献中蕴含的知识.

参 考 文 献

[1] Zhou Minghui, Guo Changguo. New thought of software engineering based big data [J]. Communications of the CCF, 2014, 10(3): 37-42 (in Chinese)
(周明辉, 郭长国. 基于大数据的软件工程新思维[J]. 中国计算机学会通讯, 2014, 10(3): 37-42)

[2] Zhang Dongmei, Han Shi, Lou Jianguang, et al. Software analytics-key points and practice [J]. Communications of the CCF, 2014, 10(3): 29-36 (in Chinese)
(张冬梅, 韩石, 楼建光, 等. 软件解析学——要点与实践[J]. 中国计算机学会通讯, 2014, 10(3): 29-36)

[3] He Keqing, Li Bing, Ma Yutao, et al. Key techniques of software engineering in the era of big data [J]. Communications of the CCF, 2014, 10(3): 8-18 (in Chinese)
(何克清, 李兵, 马于涛, 等. 大数据时代的软件工程关键技术[J]. 中国计算机学会通讯, 2014, 10(3): 8-18)

[4] Xie Tao, Pei Jian, Hassan A E. Mining software engineering data [C] //Proc of IEEE ICSE'07 Compaion. Piscataway, NJ: IEEE, 2007: 172-173

[5] Xie Tao, Thummalapenta S, Lo D, et al. Data mining for software engineering [J]. Computer, 2009, 42(8): 55-62

[6] Li Xiaochen, Jiang He, Ren Zhilei. Data driven feature extraction for mining software repositories [J]. Computer Science, 2015, 42(9): 159-164 (in Chinese)
(李晓晨, 江贺, 任志磊. 面向软件仓库挖掘的数据驱动特征提取方法[J]. 计算机科学, 2015, 42(9): 159-164)

[7] Xuan Jifeng, Jiang He, Ren Zhilei, et al. Developer prioritization in bug repositories [C] //Proc of IEEE ICSE'07. Piscataway, NJ: IEEE, 2012: 25-35

[8] Lü Linyuan, Zhang Yicheng, Yeung C H, et al. Leaders in social networks, the delicious case [J]. PloS One, 2011, 6(6): e21202

[9] Hassan A E, Xie Tao. Software intelligence: The future of mining software engineering data [C] //Proc of the 10th ACM FSE/SDP Workshop on Future of Software Engineering Research. New York: ACM, 2010: 161-166

[10] Eunjoo L E E, Chisu W U. A survey on mining software repositories [J]. IEICE Trans on Information and Systems, 2012, 95(5): 1384-1406

[11] Wang Feiyue. Publication and impact: A bibliographic analysis [J]. IEEE Trans on Intelligent Transportation Systems, 2010, 11(2): 250-250

[12] Li Linjing, Li Xin, Li Zhenjiang, et al. A bibliographic analysis of the IEEE Transactions on Intelligent Transportation Systems literature [J]. IEEE Trans on Intelligent Transportation Systems, 2010, 11(2): 251-255

[13] Heilig L, Voβ S. A scientometric analysis of cloud computing literature [J]. IEEE Trans on Cloud Computing, 2014, 2(3): 266-278

[14] Park D H, Kim H K, Choi I Y, et al. A literature review and classification of recommender systems research [J]. Expert Systems with Applications, 2012, 39(11): 10059-10072

[15] Li Linjing, Li Xin, Cheng Changjian, et al. Research collaboration and ITS topic evolution: 10 years at T-ITS [J]. IEEE Trans on Intelligent Transportation Systems, 2010, 11(3): 517-523

[16] Xu Xiujuan, Wang Wei, Liu Yu, et al. A bibliographic analysis and collaboration patterns of IEEE Transactions on Intelligent Transportation Systems between 2000 and 2015 [J]. IEEE Trans on Intelligent Transportation Systems, 2016, 17(8): 2238-2247

[17] Lindsey D. Production and citation measures in the sociology of science: The problem of multiple authorship [J]. Social Studies of Science, 1980, 10(2): 145-162

[18] Ward P L. Foundations of Library and Information Science [M]. New York: Anmol Publications, 2006: 3287-3292

[19] Holsapple C W, Johnson L E, Manakyan H, et al. Business computing research journals: A normalized citation analysis [J]. Journal of Management Information Systems, 2015, 11(1): 131-140

[20] Borgatti S P. Netdraw network visualization [R/OL]. Cambridge: Analytic Technologies, 2002 [2016-08-01]. http://www.analytictech.com/netdraw/netdraw.htm

[21] Podgurski A, Leon D, Francis P, et al. Automated support for classifying software failure reports [C] //Proc of IEEE ICSE'03. Piscataway, NJ: IEEE, 2003: 465-475

[22] Dang Yingnong, Wu Rongxin, Zhang Hongyu, et al. ReBucket: A method for clustering duplicate crash reports based on call stack similarity [C] //Proc of IEEE ICSE'12. Piscataway, NJ: IEEE, 2012: 1084-1093

[23] Kim S H, Zimmermann T, Nagappan N. Crash graphs: An aggregated view of multiple crashes to improve crash triage [C] //Proc of the 41st IEEE/IFIP Int Conf on Dependable Systems & Networks (DSN). Piscataway, NJ: IEEE, 2011: 486-493

[24] Zimmermann T, Nagappan N. Predicting defects using network analysis on dependency graphs [C] //Proc of ACM ICSE'08. New York: ACM, 2008: 531-540

[25] Chang R Y, Podgurski A, Yang J. Discovering neglected conditions in software by mining dependence graphs [J]. IEEE Trans on Software Engineering, 2008, 34(5): 579-596

[26] Runeson P, Alexandersson M, Nyholm O. Detection of duplicate defect reports using natural language processing [C] //Proc of IEEE ICSE'07. Piscataway, NJ: IEEE, 2007: 499-510

[27] Wang Xiaoyin, Zhang Lu, Xie Tao, et al. An approach to detecting duplicate bug reports using natural language and execution information [C] //Proc of ACM ICSE'08. New York: ACM, 2008: 461-470

[28] Nguyen A T, Lo D, Nguyen T N, et al. Duplicate bug report detection with a combination of information retrieval and topic modeling [C] //Proc of IEEE ASE'12. Piscataway, NJ: IEEE, 2012: 70-79

[29] Sun Chengnian, Lo D, Wang Xiaoyin, et al. A discriminative model approach for accurate duplicate bug report retrieval [C] //Proc of ACM ICSE'10. New York: ACM, 2010: 45-54

[30] Tang Shaohu, Li Zhengxi, Chen Dewang, et al. Theme classification and analysis of core articles published in IEEE Transactions on Intelligent Transportation Systems from 2010 to 2013 [J]. IEEE Trans on Intelligent Transportation Systems, 2014, 15(6): 2710-2719

[31] Hacohenkerner Y. Automatic extraction of keywords from abstracts [C] //Proc of the 7th Int Conf on Knowledge-Based and Intelligent Information and Engineering Systems. Berlin, Springer, 2003: 843-849

[32] OSCAR. The public-access stop word list [EB/OL]. 2016 [2016-10-22]. http://oscar-lab.org/chn/resource.htm

[33] Hirsch J E. An index to quantify an individual's scientific research output [J]. Proceedings of the National Academy of Sciences of the United States of America, 2005, 102(46): 16559-16572

[34] Alcaide G G, Gómez M C, Zurián J C V, et al. Scientific literature by Spanish authors on the analysis of citations and impact factor in biomedicine (1981—2005) [J]. Revista Española De Documentación Científica, 2008, 31(3): 344-365

[35] Oppenheim C. Using the H-index to rank influential British researchers in information science and librarianship [J]. Journal of the American Society for Information Science & Technology, 2007, 58(2): 297-301

[36] Bornmann L, Daniel H. What do we know about the h index? [J]. Journal of the American Society for Information Science & Technology, 2007, 58(9): 1381-1385

[37] Alonso S, Cabrerizo F J, Herrera-Viedma E, et al. H-index: A review focused in its variants, computation and standardization for different scientific fields [J]. Journal of Informetrics, 2009, 3(4): 273-289

[38] Serenko A, Bontis N. Meta-review of knowledge management and intellectual capital literature: Citation impact and research productivity rankings [J]. Knowledge and Process Management, 2004, 11(3): 185-198

[39] Cheng C H, Holsapple C W, Lee A. Citation-based journal rankings for AI research: A business perspective [J]. AI Magazine, 1996, 17(2): 87-97

[40] Girvan M, Newman M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences, 2002, 99(12): 7821-7826

[41] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks [J]. Physical Review E, 2004, 70(6): 066111

Jiang He, born in 1980. PhD, Professor and PhD supervisor at the School of Software, Dalian University of Technology, China. Member of the China Computer Federation and the ACM. His main research interests include search based software engineering, software testing and mining software repositories.

Chen Xin, born in 1987. PhD candidate. His main research interests include software testing and mining software repositories, etc.

Han Xuejiao, born in 1993. Master candidate. Her main research interest is mining software repositories.

Zhang Jingxuan, born in 1988. PhD candidate. His main research interests include mining software repositories and API document analysis, etc.

Xu Xiujuan, born in 1978. PhD and assistant professor at the School of Software, Dalian University of Technology, China. Her main research interests include applications of data mining, intelligent transportation systems, recommender systems, and social network.

《计算机研究与发展》征订启事

《计算机研究与发展》(Journal of Computer Research and Development)是中国科学院计算技术研究所和中国计算机学会联合主办、科学出版社出版的学术性刊物,中国计算机学会会刊. 主要刊登计算机科学技术领域高水平的学术论文、最新科研成果和重大应用成果. 读者对象为从事计算机研究与开发的研究人员、工程技术人员、各大专院校计算机相关专业的师生以及高新企业研发人员等.

《计算机研究与发展》于 1958 年创刊,是我国第一个计算机刊物,现已成为我国计算机领域权威性的学术期刊之一. 并历次被评为我国计算机类核心期刊,多次被评为“中国百种杰出学术期刊”. 此外,还被《中国学术期刊文摘》、《中国科学引文索引》、“中国科学引文数据库”、“中国科技论文统计源数据库”、美国工程索引(Ei)检索系统、日本《科学技术文献速报》、俄罗斯《文摘杂志》、英国《科学文摘》(SA)等国内外重要检索机构收录.

国内邮发代号:2-654;国外发行代号:M603

国内统一连续出版物号:CN11-1777/TP

国际标准连续出版物号:ISSN1000-1239

联系方式:

100190 北京中关村科学院南路 6 号《计算机研究与发展》编辑部

电话: +86(10)62620696(兼传真);+86(10)62600350

Email:crad@ict.ac.cn

http://crad.ict.ac.cn