

面向对象软件内聚度量数据分布的实证研究*

周晓聪¹, 赖蔚¹, 温剑丰²

¹(中山大学 数据科学与计算机学院, 广东 广州 510275)

²(广东技术师范学院 计算机科学学院, 广东 广州 510665)

通讯作者: 周晓聪, E-mail: isszxc@mail.sysu.edu.cn



摘要: 度量数据的分布信息对于理解和使用面向对象软件度量有重要意义. 人们对面向对象软件规模度量、耦合度度量乃至继承维度的度量数据的分布都有研究, 但对除内聚度缺乏度 LCOM 之外的内聚度量数据的分布却缺乏研究. 已有的实证研究表明, LCOM 并不是好的内聚度量, 因此探讨其他内聚度量数据分布很有必要. 对包括内聚度缺乏度、基于连通性的内聚度度和基于相似性的内聚度度量总共 17 个度量指标在 112 个 Java 开源软件项目的分布情况进行实证研究, 对每个度量指标的每个项目数据使用幂律分布和对数正态分布进行拟合, 并使用荟萃分析方法对拟合结果进行了分析. 实证研究结果表明, 非规范化的内聚度量可使用对数正态分布和幂律分布拟合, 但规范化的基于相似性的内聚度量(包括 CC、LSCC、SCOM 和 SCC)需要排除方法数小于等于 1 或字段数为 0 的特殊类才能使用对数正态分布拟合, 而基于连通性的内聚度度量(包括 TCC、LCC、DCD 和 DCI)则只有对应的非规范化版本的数据才符合对数正态分布或幂律分布. 实证研究可帮助人们更好地理解和使用内聚度度量, 特别是可以帮助人们如何利用已有的方法确定内聚度度量的阈值.

关键词: 面向对象软件; 内聚度度量; 统计分布; 实证研究

中图法分类号: TP311

中文引用格式: 周晓聪, 赖蔚, 温剑丰. 面向对象软件内聚度量数据分布的实证研究. 软件学报, 2018, 29(10):3051-3067. <http://www.jos.org.cn/1000-9825/5290.htm>

英文引用格式: Zhou XC, Lai W, Wen JF. Empirical study on the distribution of object-oriented software cohesion metrics. Ruan Jian Xue Bao/Journal of Software, 2018, 29(10):3051-3067 (in Chinese). <http://www.jos.org.cn/1000-9825/5290.htm>

Empirical Study on the Distribution of Object-Oriented Software Cohesion Metrics

ZHOU Xiao-Cong¹, LAI Wei¹, WEN Jian-Feng²

¹(School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510275, China)

²(School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China)

Abstract: The distribution information of measurement data is important for understanding and using object-oriented software metrics. However, there is no research on the distribution of object-oriented software cohesion metrics except the CK metric LCOM. Unfortunately, existing studies show that LCOM is not a good metric for class cohesion of object-oriented software. Therefore, it is necessary to investigate the distribution of other cohesion metrics. Based on 112 Java open-source software systems, the distribution of measurement data of 17 cohesion metrics, including those lack of cohesion metrics (e.g. LCOM1, LCOM2, LCOM3, LCOM4, LCOM5),

* 基金项目: 广东省自然科学基金(2015A030310394); 广东省重大科技专项(2016B030305006, 2014B010112001); 国家自然科学基金(61672546)

Foundation item: Natural Science Foundation of Guangdong Province of China (2015A030310394); Important Science and Technology Specific Project of Guangdong Province of China (2016B030305006, 2014B010112001); Natural Science Foundation of China (61672546)

收稿时间: 2016-09-21; 修改时间: 2016-12-10, 2017-03-22; 采用时间: 2017-04-06; jos 在线出版时间: 2017-07-12

CNKI 网络优先出版: 2017-07-12 15:34:00, <http://kns.cnki.net/kcms/detail/11.2560.TP.20170712.1534.010.html>

connectivity-based cohesion metrics (e.g. TCC, LCC, DCI, DCD), and similarity-based cohesion metrics (e.g. CC, LSCC, SCOM, SCC), are investigated empirically in this paper. The results show that the data of non-normalized cohesion metric can be fitted into log-normal distribution and power law distribution, whereas the data of similarity-based cohesion metrics should be fitted into log-normal distribution only when excluding those special class with at most one method and without field data, and, for connectivity-based cohesion metrics, the data of their corresponding non-normalized cohesion metric follow log-normal distribution and power law distribution. This study provides important information for researchers to understand and use cohesion metrics, in particular, to identify thresholds for these metrics using the approaches in the literatures.

Key words: object-oriented software; cohesion metrics; statistic distribution; empirical study

1 引言

软件度量(software measurement)是评价和管理软件质量的重要手段,软件度量指标(software metrics)是在软件度量过程中使用数值描述度量对象(即某个软件构件)属性的函数或方法^[1].软件项目包含多个要度量的软件构件,它们在同样的软件度量指标上取不同的值.在评测软件质量时,软件开发人员需要基于软件度量指标数据的分布信息确定度量指标的阈值或参考值以区分软件质量的好坏^[2-5],因此确定软件度量指标数据的分布是理解和应用度量指标的关键.

对于面向对象软件,人们从代码规模、内聚度、耦合度和继承这4个维度提出了许多软件度量指标,并对它们在评测软件构件的易错性(fault proneness)、易变性(change proneness)等方面的作用进行了广泛研究^[6,7].在这些度量指标数据的分布方面也存在许多实证研究.例如,Succi等人对最早提出的面向对象软件度量指标集——CK度量集^[8]的度量数据分布进行了深入的实证研究^[9];Wheeldon和Counsell对3个Java开源系统进行了研究,发现与类的规模和耦合度有关的12个度量的数据的分布符合幂律分布^[10];Baxter等人扩展了Wheeldon和Counsell的工作,使用幂律分布、对数正态分布和两参数的Weibull分布对56个Java开源系统的18个与类的规模和耦合度有关的度量的数据进行加权最小二乘拟合,发现其中多数度量支持幂律分布或对数正态分布^[11];Concas对一个大型Smalltalk软件的10条与类的规模、耦合度和继承相关的性质(度量)进行了实证研究,发现它们符合幂律分布或对数正态分布^[12].另外,还有许多学者对于与类的规模、耦合度相关的性质(度量)进行了研究,特别是考察了幂律分布、对数正态分布、双Pareto分布是否能够描述这些度量数据的分布^[3,13-16].最近,Destefanis等人使用10个Java开源软件和10个Python软件对9个与类的规模、耦合度和继承相关度量的数据的分布进行了研究,并基于度量数据的分布比较了Java软件和Python软件的不同^[17].

但是,已有的对面向对象软件度量指标数据分布的研究工作主要关注于代码规模和耦合度度量,而对于内聚度量,除最早提出的CK度量集中的内聚度缺乏度LCOM度量外,其他多数内聚度量都缺乏与数据分布相关的研究.面向对象软件的内聚度主要度量类的内聚度,衡量类的成员之间联系的紧密程度,是评测面向对象软件质量的重要因素.人们提出了许多类内聚度量指标^[18-20],这些指标对软件易错性^[21,22]、易变性^[7]和可重用性^[23]有一定的预测能力,并在评测软件设计缺陷方面有重要作用^[2].例如,Lanza和Marinescu使用类内聚度量指标TCC检测所谓的“上帝类(god class)”这种软件设计缺陷^[2],在假定这个度量指标数据分布是正态分布的情况下,以TCC值小于1/3作为“上帝类”的特征之一.虽然LCOM度量的分布已有相关研究,然而不少实证研究表明,它不是好的类内聚度量指标^[6,24],因此很有必要探索更多内聚度量指标数据的分布情况.

本文基于Qualitas Corpus(<http://www.qualitascorpus.com>)中的112个Java开源项目^[25]对17个内聚度量指标的数据分布进行研究,研究的问题是:能否使用幂律分布或对数正态分布模型拟合内聚度量数据.我们在使用EasyFit(<http://www.mathwave.com/products/easyfit.html>)软件对多个内聚度量数据分布情况进行初步研究的基础上选择这两种分布进行详细研究.目前对软件规模、耦合度度量指标数据分布的研究也多数是使用这两种分布进行拟合,而且目前对度量阈值确定方法的研究也都基于度量数据分布符合重尾分布,幂律分布和对数正态分布则是最典型的重尾分布.在对内聚度量指标数据分布进行分析的基础上,我们也会对在确定度量阈值时如何选择和应用这些内聚度量指标给出建议.

本文第2节介绍我们对内聚度量数据分布进行拟合和分析的方法.第3节给出对Qualitas Corpus中112

个 Java 软件开源项目的内聚度量数据分布及其统计量特征的实证研究结果,基于实例探讨实证研究结果在内聚度量阈值的确定以及如何选择内聚度量检测软件设计缺陷等方面的应用,并分析我们实证研究的局限性.第 4 节概述相关工作.第 5 节是文章结论和下一步工作展望.

2 度量数据分布分析方法

本文主要是基于大量 Java 开源项目报告 Java 软件的内聚度量指标数据可能的分布和相关的统计量,使得人们对当前 Java 开源软件的内聚度量情况有更好的理解,从而在构建软件质量模型时可以更好地选择与内聚度相关的度量指标.

我们对每个内聚度量指标对于每个项目的类的度量结果使用幂律分布和对数正态模型进行拟合,计算拟合的 Kolmogorov-Smirnov 距离统计量(简称 KS 统计量)及相应的 p 值.对于每个项目来说,某个度量指标数据 KS 统计量越小,或相应的 p 值越大,表明数据越符合所要拟合的分布.

但是,对于同一个度量指标,基于不同项目的度量数据进行拟合得到的 p 值可能不一样,有的可能小于给定的显著水平(从而相应的数据不符合所拟合的分布),有的可能大于给定的显著水平.为了得到一个汇总的综合结果,我们采用两种方法:一种是使用荟萃分析^[26]中的汇总假设检验 p 值的方法,另外一种是我们对于拟合 p 值是否大于给定显著水平进行符号检验.

荟萃分析^[26]是一种将研究同一问题的多个研究结果汇总成综合结果的统计方法.问题的研究结果可以是实验研究中因果关系的效应(effect)、假设检验的 p 值以及变量相关关系的相关系数等.对每个度量指标,一个项目的数据的幂律分布拟合或对数正态分布拟合就得到一个 p 值,假设 N 个项目得到的 p 值为 P_1, P_2, \dots, P_N .假设检验结果的汇总分析是要基于这 N 个 p 值给出如下全局原假设 H_0 (global null hypothesis)及对应的备选假设 H_1 的检验 p 值.

H_0^G : N 次检验的所有原假设都为真,也即每个项目的数据都符合所拟合的理论分布.

H_1^G : N 次检验中有些备选假设为真,也即有些项目的数据不符合所拟合的理论分布.

我们使用文献[26]中介绍的 Fisher 方法将分布拟合的 p -值进行汇总分析,这种方法基于这样的事实:在全局原假设成立的情况下,只要代表 p 值的随机变量有一致的分布,随机变量 $-2\ln P$ 则符合自由度为 2 的 χ^2 分布,进而 N 个这样的值是自由度为 $2N$ 的 χ^2 变量.因此,当 $-2\sum_{i=1}^N \ln P_i$ 超过自由度为 $2N$ 的 χ^2 分布的 $100(1-\alpha)\%$ 分位数临界值时,在显著水平 α 上拒绝全局原假设,进一步地,也可以基于这个分布给出全局检验的 p 值,它等于自由度为 $2N$ 的 χ^2 分布中大于等于 $-2\sum_{i=1}^N \ln P_i$ 的概率.

除了使用 Fisher 方法对分布拟合得到的 p 值进行汇总分析之外,我们还借鉴文献[9]中分析度量 NOC 和 DIT 的取值范围的方法对一个度量指标的 N 个项目数据的分布拟合 p 值大于给定显著水平是否是偶然现象进行符号检验,也即对于每个度量指标,检验如下假设.

H_0^p : 对度量指标数据进行分布拟合得到的 p 值大于给定显著水平 α_T 的概率小于等于 0.5.

H_1^p : 对度量指标数据进行分布拟合得到的 p 值大于给定显著水平 α_T 的概率大于 0.5.

对于每个度量指标的数据进行拟合, N 个项目得到 N 个 p 值 P_1, P_2, \dots, P_N .根据符号检验的理论^[27],统计其中小于等于 α_T 的 $P_i, i=1, \dots, N$ 的个数 K ,然后基于二项分布计算概率:

$$P(X \leq K) = \sum_{i=0}^K \binom{N}{i} \left(\frac{1}{2}\right)^i \left(1 - \frac{1}{2}\right)^{N-i} = \frac{1}{2^N} \sum_{i=0}^K \binom{N}{i}.$$

若 $P(X \leq K)$ 小于等于 0.05,则我们以显著水平 0.05 拒绝上述原假设 H_0^p ,而接受备选假设 H_1^p ,也即认为对度量指标数据进行分布拟合得到的 p 值大于给定显著水平 α_T 更为常见.进一步可确定一个最大的 α_{\max} ,使得对 N 项目进行分布拟合中得到的 p 值小于等于 α_{\max} 的个数 K_{\max} 仍满足 $P(X \leq K_{\max})$ 小于等于 0.05.显然, α_{\max} 越大,我们越有理由相信该度量指标数据符合所拟合的理论分布.

后面我们也将上述方法用于分析度量指标数据取最大值、最小值比例情况的分析.例如,对于某个度量指

标的 N 个项目的数据,分别统计每个项目该度量指标取最小值的类占总类数的比例,得到 N 个比例值 r_1, r_2, \dots, r_N . 对于某个阈值 T , 检验如下假设.

$H_0^>$: 度量指标数据取最小值类占比大于给定阈值 T 的概率小于等于 0.5.

$H_1^>$: 度量指标数据取最小值类占比大于给定阈值 T 的概率大于 0.5.

统计 N 个比例值 r_1, r_2, \dots, r_N 中小于等于给定阈值 T 的项目个数 K , 如果 $P(X \leq K)$ 小于等于 0.05, 则可认为度量指标数据取最小值类数占比大于给定阈值 T (比小于等于该阈值) 更常见. 进一步地, 可确定一个最大可能阈值 T_{\max} , 使得小于等于该阈值 T_{\max} 的项目个数 K_{\max} 仍满足 $P(X \leq K_{\max}) \leq 0.05$.

3 内聚度度量分布的实证研究结果

3.1 内聚度度量指标

目前人们已经提出许多类内聚度度量指标. 最早是 Chidamber 和 Kemerer 在 1991 年提出的类内聚度缺乏度量 LCOM1, 定义为不引用相同字段的方法对数目, 后来这两位学者在提出 CK 度量集时将其修正为不引用相同字段的方法对与引用相同字段方法对数目之差, 被称为 LCOM2, 或直接称为 LCOM. CK 度量集还包括类的加权方法数 WMC、对象间耦合度 CBO、类责任集 RFC、子类个数 NOC 和继承深度 DIT 等度量指标, 是当前应用最为广泛、被研究得最多的面向对象软件度量指标集.

虽然类内聚度缺乏度量 LCOM2 被广泛使用, 但研究表明, 它并不是好的内聚度度量^[6,24], 所以人们陆续提出了许多类内聚度度量指标. Briand 等人 1998 年给出了一个面向对象内聚度度量框架^[18], 综述了此前的内聚度度量. Al-Dallal 等人近几年的研究则总结了 1998 年以后提出的类内聚度度量^[20,28-30]. 我们参考 Briand 等人给出的面向对象内聚度度量框架, 以及已有对内聚度度量的实证研究^[6,7,20,28], 选择 17 个具有代表性的、被广泛研究的内聚度度量指标作为实证研究的对象.

(1) LCOM1、LCOM2、LCOM3、LCOM4 和 LCOM5 都基于方法之间是否引用相同字段定义, 并且从内聚度缺乏的角度进行度量, 值越大, 表明类的内聚度越低. Coh 是 LCOM5 的修正版本, 它的值越大, 表明内聚度越高. 这些度量中, LCOM5 和 Coh 是规范化度量, 即有最大值和最小值. LCOM5 的最大值是 2, 最小值是 0, Coh 的最大值和最小值则分别是 1 和 0.

(2) CAMC、NHD 和 ICH 则基于方法参数类型或参数个数进行计算, 它们都不是规范化度量.

(3) TCC、LCC、DCD 和 DCI 则基于方法对字段的引用关系和方法之间的调用关系确定类的两个方法是否关联(连通), 然后计算有关联的方法对数占总方法对数之比度量类的内聚度, 因此, 这些度量被称为基于连通性的内聚度量(connectivity-based cohesion metrics)^[22]. 这 4 个内聚度度量指标, 如果只考虑有关联的方法对数, 而不进行规范化(即除以总方法对数), 则可得到这 4 个度量的非规范化版本, 我们使用 TCCp、LCCp、DCDp 和 DCIp 分别表示这 4 个度量指标的非规范化版本.

(4) CC、LSCC、SCOM 和 SCC 都是规范化度量, 它们虽然也是基于方法对字段的引用和方法之间的调用关系进行计算, 但考察的是两方法之间的相似性, 因此被称为基于相似度的内聚度度量(similarity-based cohesion metrics)^[21].

表 1 总结了我们所研究的内聚度度量指标的基本含义, 其中, 度量指标缩写中带字母 p 的都是我们从文献已有度量指标中基于是否规范化而派生出来的指标. 表 1 对这些内聚度度量指标进行了简单分类, 并对它们是否规范化进行了标注.

总的来说, 现有的类内聚度度量通常根据方法对字段的引用、方法之间的调用等关系度量类成员之间联系的紧密程度, 是评测类设计质量的重要方面之一. 例如, Lanza 和 Marinescu 使用 TCC 度量检测“上帝类”这种软件设计缺陷^[2], 而 Al-Dall 和 Briand 使用 LSCC 度量判断一种方法是否该从类中移除(即移除方法重构(move method refactoring)): 若一个类移除某个方法后的内聚度 LSCC 值比保留这个方法时的 LSCC 值还大的话, 那么这个方法可以考虑移除^[21].

Table 1 Software cohesion metrics used in this study

表 1 软件内聚度量指标及其含义

指标	含义	规范	类别
LCOM1	类内聚度缺乏度,不引用相同字段的方法对数目	否	内聚度缺乏度
LCOM2	类内聚度缺乏度,不引用相同字段方法与引用相同字段方法对数之差	否	
LCOM3	以方法为顶点,两方法引用相同字段则有边构成的无向图的连通分支数	否	
LCOM4	以方法为顶点,两方法引用相同字段或有调用关系则有边构成无向图的连通分支数	否	
LCOM5	基于引用字段方法数计算,等于(方法数-引用字段平均方法数)/(方法数-1)	是	
LCOM1p	LCOM1/方法对总数	是	
LCOM2p	LCOM2/方法对总数	是	
Coh	基于引用字段方法数计算,引用字段平均方法数/方法数	是	基于连通性
TCC	基于方法间接引用共同字段关系计算,有关系方法对数/方法对总数	是	
LCC	基于方法间接引用共同字段关系的传递闭包计算,有关系方法对数/方法对总数	是	
DCD	基于方法间接引用字段和方法调用关系计算,有关系方法对数/方法对总数	是	
DCI	基于方法间接引用字段和方法调用关系的传递闭包计算,有关系方法对数/方法对总数	是	
TCCp	基于方法间接引用共同字段关系计算,有关系方法对数	否	
LCCp	基于方法间接引用共同字段关系的传递闭包计算,有关系方法对数	否	
DCDp	基于方法间接引用字段和方法调用关系计算,有关系方法对数	否	基于参数
DCIp	基于方法间接引用字段和方法调用关系的传递闭包计算,有关系方法对数	否	
CAMC	参数出现矩阵中 1 的总数/矩阵元素总数,参数类型 j 出现在方法 i 的参数则参数出现矩阵 $P_{ij}=1$	是	
NHD	方法对海明距离和/方法对总数,两方法海明距离是它们在参数出现矩阵对应两行同是 1 或同是 0 个数和	是	基于相似性
ICH	所有方法信息流总和,方法信息流是方法所多态直接调用方法的参数个数以调用次数为权的加权和	否	
CC	以两方法引用共同字段数除引用字段总数定义相似性,此相似性总和/方法对总数	是	
SCOM	以两方法(引用共同字段数/引用字段少者字段数)/(引用字段总数/类字段总数)定义相似性	是	基于相似性
LSCC	以两方法引用共同字段数/类字段总数定义相似性,此相似性总和/方法对总数	是	
SCC	基于方法调用矩阵和字段类型访问矩阵计算方法-方法、字段-字段、字段-方法各种内聚度的加权平均值	是	

3.2 度量数据收集

我们以 Qualitas Corpus 的 QualitasCorpus-20130901r 版本中收集的 112 个 Java 开源项目为实证研究对象,这些项目不仅有规模较大的软件开发平台 Eclipse,还有常用的应用服务器 tomcat,UML 视图绘制软件 argouml,游戏 freecol,软件分析工具 jmeter、emma、findbugs,对象关系映射框架 hibernate,数据挖掘平台 weka 等等,大多数开源软件在相应的应用领域中都具有很好的代表性.Qualitas Corpus 中的项目集也是近年来许多有关软件度量的实证研究所使用的数据集^[4,17,31]。

表 2 给出了上述 112 个项目数(FILE)、总的类数(CLS)、顶层类数(TopCLS)、有效代码行数(ELOC)和源代码文本行数(LOPT)的基本统计数据,即最小值、 Q_1 、中位数、 Q_3 、最大值、均值、方差等,从表 2 可以看到,这 112 个项目总共包括 11 万多个文件,12 万多个类,其中有 10 万多个顶层类,有效代码行数超过 1 400 万,代码文本行数超过 2 600 万行.这 112 个项目的规模分布非常广泛,文件数从 21~18990 个,有效代码行数从 2150~2428342.关于这些项目的更详细信息可参考 Qualitas Corpus 的网页(<http://www.qualitascorpus.com>)(AutoBVT).

Table 2 Descriptive statistics for the studied open source Java projects in Qualitas Corpus

表 2 Qualitas Corpus 中 Java 开源项目的基本情况

指标	最小值	Q_1	中位数	Q_3	最大值	均值	方差	总数
FILE	21	189.0	473.5	1 035.0	18 990	1 035.54	2 267.70	115 981
CLS	26	198.5	513.0	1 108.5	20 267	1 154.14	2 675.55	129 264
TopCLS	16	159.0	416.5	910.5	16 030	894.21	1 923.73	100 151
ELOC	2 150	21 436.0	51 222.5	112 103.0	2 428 342	131 440.93	305 165.57	14 721 384
LOPT	2 709	34 383.5	90 446.5	222 818.0	4 106 988	237 326.25	533 941.15	26 580 540

我们使用 Java 语言编写工具收集了这 112 个项目顶层类的度量数据,除了上面给出的内聚度量指标外,还收集了类实现的方法数(IMPMTD)和声明的字段数(FLD)以及类的有效代码行数(ELOC)这 3 个软件规模度量指标的数据,类实现的方法是指在这个类中有方法体的方法,我们在计算类的内聚度时只考虑类实现的方法和声明的字段数(不考虑类继承的方法和字段).对于没有字段、没有实现方法或只有一种实现方法的特殊类,我们根据 Al-Dallal 的研究^[28]来设定这些特殊类的内聚度量指标值。

这样一共得到 112 个数据表,每个数据表是一个项目中顶层类的度量数据,其中的每一行包含该项目中的一个类在不同度量指标的取值,因此数据表的一列给出了一个度量指标在这个项目中的各个类的取值,是这个度量指标的一个样本数据,从而每个度量指标都有 112 个样本数据.下面我们所说的某个度量指标的一个样本数据指的就是该度量指标在一个项目中的各个类的取值.

3.3 度量数据分布分析

对每个度量指标的每个样本数据,我们使用 R 软件程序包 `powerLaw` 提供的函数对其进行幂律分布和对数正态分布的拟合^[32],得到拟合的 KS 统计量,并计算相应的 p 值,这个 p 值可用于判断拟合的好坏程度, p 值越大说明拟合得越好.与直接使用拟合优度检验(goodness-of-fit test)中的 Kolmogorov-Smirnov 检验(简称 KS 检验)不同,程序包 `powerLaw` 所进行的拟合会使用极大似然估计法估计用于拟合数据的最佳理论分布参数,并同时支持连续和离散的幂律分布和对数正态分布拟合.若直接使用 KS 检验,则需要另外估计用于比较的理论分布参数,并且只适合于连续分布的拟合优度检验.

我们对规模度量 IMPMTHD、FLD、ELOC 和内聚度量 LCOM1、LCOM2、LCOM3、LCOM4、ICH 使用离散幂律分布和离散对数正态分布模型进行拟合,而对其他内聚度量使用连续幂律分布和连续对数正态分布模型进行拟合.

表 3 给出了这些度量指标数据在使用分布模型进行拟合时的 p 值的统计情况,其中,第 2 列给出了每个度量指标的有效 p 值个数.当一个度量指标在某个项目的类的不同取值个数小于 8 时,对其进行拟合没有价值,将得到无效的拟合 p 值.表 3 的第 3 列和第 6 列给出了每个度量指标拟合 p 值超过显著水平阈值 0.05 的项目数,第 4 列和第 6 列给出了能够设置的最大显著水平阈值.

Table 3 Results of fitting into power-law and lognormal distributions

表 3 各个度量指标的数据分布模型拟合的 p 值统计

度量指标	有效 p 值数	对数正态分布 拟合 p 值大于 0.05 的项目数	对数正态分布 拟合的 α_{\max}	对数正态分布 拟合 Fisher 汇总 p 值	幂律分布拟合 p 值大于 0.05 的项目数	幂律分布拟合 的 α_{\max}	幂律分布拟合 Fisher 汇总 p 值
IMPMTHD	112	111	0.948 926 7	1	109	0.860 545 9	1
FLD	109	108	0.990 489 9	1	107	0.899 660 0	1
ELOC	112	112	0.954 373 5	1	107	0.762 162 1	0.999 990 6
LCOM11	112	110	0.882 419 5	0.999 882 7	109	0.731 697 8	1
LCOM2	110	110	0.922 180 6	1	108	0.776 783 9	1
LCOM1p	112	15	0.000 000 0	0	5	0.000 000 0	0
LCOM2p	110	49	0.004 571 5	8.3411E-271	13	0.000 000 0	0
LCOM3	110	106	0.955 721 5	1	108	0.879 660 8	1
LCOM4	101	95	0.954 164 0	0.999 996 9	98	0.915 233 4	1
LCOM5	112	38	0.000 503 4	6.6322E-319	40	0.000 617 8	3.2848E-318
Coh	112	59	0.025 883 1	3.5052E-144	32	0.000 082 3	0
TCC	112	32	0.000 001 5	0	17	0.000 000 0	0
LCC	111	19	0.000 000 0	0	6	0.000 000 0	0
DCD	112	28	0.000 000 1	0	11	0.000 000 0	0
DCI	111	5	0.000 000 0	0	2	0.000 000 0	0
TCCp	111	109	0.840 506 5	1	110	0.783 658 2	1
LCCp	111	106	0.658 159 7	0.991 518 2	108	0.632 367 1	0.999 893 5
DCDp	112	109	0.845 408 2	1	111	0.786 212 5	1
DCIp	112	106	0.620 976 0	0.940 122 4	109	0.615 056 9	0.999 971 7
CAMC	112	48	0.001 640 8	1.9368E-265	23	0.000 000 0	0
NHD	112	89	0.440 051 9	5.5277E-36	11	0.000 000 0	0
ICH	110	110	0.950 587 2	1	108	0.844 074 4	1
CC	112	44	0.000 263 2	0	31	0.000 000 1	0
LSCC	112	49	0.003 687 5	3.2872E-285	30	0.000 000 3	0
SCOM	112	44	0.002 711 5	5.8298E-307	29	0.000 000 4	0
SCC	112	92	0.496 100 1	2.3048E-16	79	0.154 411 0	1.1975E-65

可以看到:

(1) 软件规模度量指标 IMPMTHD、FLD 和 ELOC 的最大显著水平阈值都显著大于 0.5,而且 Fisher 检验的 p 值接近 1,说明这些度量指标的所有项目数据都既可使用幂律分布也可使用对数正态分布拟合;

(2) 非规范化的内聚度量指标 LCOM1、LCOM2、LCOM3、LCOM4、TCCp、LCCp、DCDp、DCIp 和 ICH 的最大显著水平阈值也都显著大于 0.5,而且 Fisher 检验的 p 值接近 1,说明这些度量指标的所有项目数据也都既可使用幂律分布也可使用对数正态分布拟合;

(3) 规范化的内聚度量指标 NHD 和 SCC 的对数正态分布拟合最大显著水平阈值大于 0.4,但 Fisher 检验的 p 值远小于 0.05,说明这两个指标虽然不能得到所有项目都可使用对数正态分布拟合,但是多数项目(超过 65 个项目)的对数正态分布拟合的 p 值大于 0.4.指标 SCC 的幂律分布拟合最大显著水平阈值为 0.15,也表示它的多数项目数据的幂律分布拟合 p 值大于 0.15.也就是说,对于 NHD 和 SCC,度量数据符合对数正态分布(比不符合对数正态分布)更为常见,而 SCC 的度量数据符合幂律分布的情况也比较常见;

(4) 除 NHD 和 SCC 以外的规范化内聚度量指标(两种分布拟合)的最大显著水平阈值都远小于 0.05,说明这些度量指标的数据既不能使用幂律分布也不能使用对数正态分布进行拟合.

那么为什么规范化的内聚度量指标都不能使用对数正态分布或幂律分布这种重尾分布进行拟合呢?经过分析发现,每个项目都有不少类在这些度量指标上取理论上的最小值和理论上的最大值.对每个度量指标的每个项目的数据,计算统计量 R_{min} 和 R_{max} .

$$R_{min} = \frac{\text{项目中在该度量指标上取最小值的类个数}}{\text{该项目的类总数}},$$

$$R_{max} = \frac{\text{项目中在该度量指标上取最大值的类个数}}{\text{该项目的类总数}}.$$

对于 112 个项目的 R_{min} 和 R_{max} 以阈值 $T=10\%$ 进行符号检验.

$H_0^{R_{min}}$: 各项目度量指标数据的 R_{min} 值大于给定阈值 T 的概率小于等于 0.5.

$H_1^{R_{min}}$: 各项目度量指标数据的 R_{min} 值大于给定阈值 T 的概率大于 0.5.

$H_0^{R_{max}}$: 各项目度量指标数据的 R_{max} 值大于给定阈值 T 的概率小于等于 0.5.

$H_1^{R_{max}}$: 各项目度量指标数据的 R_{max} 值大于给定阈值 T 的概率大于 0.5.

分别计算最大可能的阈值 T_{max} 和,使得小于该阈值 T_{max} 的项目个数 K_{max} 仍满足 $P(X \leq K) \leq 0.05$,取 $N=112$,我们得到 $K_{max}=47$.

Table 4 Ratios of classes with minimal or maximal values in normalized cohesion metrics

表 4 规范化内聚度量指标数据最小值/最大值类占总类数比例情况统计

度量	R_{min} 均值	R_{min} 方差	$R_{min}>10\%$ 的项目数	R_{min} 最大可能阈值 T_{max}	R_{max} 均值	R_{max} 方差	$R_{max}>10\%$ 的项目数	R_{max} 最大可能阈值 T_{max}
LCOM1p	0.188 1	0.103 4	97	0.155 0	0.252 3	0.101 5	108	0.219 6
LCOM2p	0.476 8	0.108 5	112	0.442 3	0.091 9	0.066 3	39	0.069 1
LCOM5	0.105 8	0.060 7	58	0.085 2	0.081 0	0.072 1	25	0.055 9
Coh	0.188 6	0.098 5	94	0.154 3	0.099 8	0.058 3	50	0.079 9
TCC	0.251 3	0.101 5	107	0.219 5	0.193 1	0.102 8	99	0.160 0
LCC	0.251 3	0.101 5	107	0.219 5	0.256 9	0.107 1	107	0.226 3
DCD	0.149 1	0.078 9	82	0.125 0	0.216 6	0.102 7	104	0.186 9
DCI	0.149 1	0.078 9	82	0.125 0	0.381 3	0.116 4	112	0.349 6
CAMC	0.047 7	0.052 9	14	0.027 2	0.097 6	0.077 3	38	0.072 9
NHD	0.193 1	0.113 0	93	0.149 4	0.045 6	0.047 2	11	0.028 9
CC	0.258 3	0.103 7	108	0.227 6	0.151 1	0.097 2	77	0.118 3
LSCC	0.249 2	0.104 2	106	0.218 6	0.139 8	0.085 6	73	0.110 3
SCOM	0.258 3	0.103 7	108	0.227 6	0.146 1	0.087 6	72	0.115 4
SCC	0.183 9	0.102 4	94	0.148 1	0.039 9	0.028 4	3	0.031 9

表 4 给出各度量指标数据取最小值/最大值类占总类数比例情况的统计,其中,第 2 列给出了 R_{min} 在 112 个

项目中的平均情况,可以看到,LCOM2p 由于本身定义的问题,112 个项目平均来说有超过 40%的类取最小值(即 0),TCC、LCC、CC、LSCC 和 SCOM 等指标也都有接近四分之一的类取最小值.这个表的第 4 列给出了 R_{\min} 大于 10%的项目数,可以看到,对于指标 LCOM2p,所有项目取最小值的类数占比都超过 10%,表的第 5 列给出了 R_{\min} 的最大可能阈值,指标 LCOM2p 的最大可能阈值为 44.23%,这意味着只有 47 个项目的 R_{\min} 值小于等于 44.23%,也即有 65 个项目的 R_{\min} 值大于 44.23%,这个比例相当高,说明面向对象软件的类在 LCOM2p 指标取最小值 0 的情况相当常见.TCC、LCC、CC、LSCC 和 SCOM 等指标的 R_{\min} 最大可能阈值也都超过 20%,说明面向对象软件的类在这些指标上取最小值 0 的情况也比较常见.

表 4 的第 5 列~第 8 列给出了面向对象软件类在这些指标上取最大值的情况,可以看到,指标 DCI 的 R_{\max} 的均值和最大可能阈值都超过 30%,表明类在指标 DCI 取最大值 1 的情况很常见,这是因为 DCI 基于间接方法调用和间接字段引用计算类的内聚度,使得不少类的内聚度达到最大理论值 1.指标 LCOM1p 和 LCC 的 R_{\max} 的均值和最大可能阈值也都超过 20%,表明类在这两个指标上取最大值的情况也比较常见.

显然,当一个度量指标的数据取最大值和最小值的数据个数比较多时,这个数据的分布不可能是对数正态分布或幂律分布,而是呈一种类似 U-型分布,这种分布很难确定数据中的离群值,也即数据呈 U-型分布的内聚度量指标很难找到合适的标准去确定一个类的内聚度是否太高或太低.同时,当一个度量指标的数据取最大值和最小值的数据个数比较多时,该度量指标的区分能力也受到影响,它无法区分都是取最大值或最小值的那些类的内聚度.Al Dallal 的研究表明,DCI 和 LCC 的区分度比 DCD 和 TCC 要低,而 LCOM2 的区分度比 LCOM1 低^[20].上面也发现 DCI 和 LCC 的数据取最小值或最大值的比例比 DCD 和 TCC 的数据要高,这从某种角度解释了它们之间区分度的差别.

对于至多包含一种方法或没有字段(数据成员)的特殊类,我们按照 Al Dallal 的研究^[28]指定这些特殊类的内聚度量指标值.由于这些特殊类所指定的内聚度量指标值要么是最小值 0,要么是最大值 1,因此,这使得内聚度量数据的分布更倾向于 U-型分布.按照 Al Dallal 的统计,这种特殊类可能在一个项目中多达 30%,在我们考察的 112 个项目总共 100 151 个类中也有 25 926 个这样的特殊类,占比达到 25.9%.分项目来看,特殊类占比最少的项目是 jpf-1.5.1,全部 93 个类中有 5 个特殊类,占 5.4%,最多的是 jgraphpad-5.10.02,全部 291 个类中有 192 个特殊类,占比为 65.9%.112 个项目特殊类占比的平均值也是 25.1%,说明特殊类在项目确实占有很大比例.

为了更好地理解内聚度量数据的分布情况,我们为每个项目挑出非特殊类(即方法数大于 1 且字段数大于 0 的类)的度量数据进行分析.表 5 给出上述规范化内聚度量指标非特殊类的数据取最小值/最大值类占总非特殊类数比例情况的统计.与表 4 相比,可以看到,排除特殊类之后,在各指标取最小值和最大值的非特殊类占比显著下降,这表明,特殊类的存在是导致规范化内聚度量数据呈 U-型分布的关键因素.从表 5 还可以看到,即使排除特殊类,指标 LCOM2p 取最小值 0 的类占比仍然超过 30%,而指标 DCI 取最大值 1 的类占比也超过 30%,说明这两个度量指标的定义存在缺陷.

Table 5 Ratios of non-special classes with minimal or maximal values in normalized cohesion metrics

表 5 规范化内聚度量指标非特殊类数据最小值/最大值类占总类数比例情况统计

度量	R_{\min} 均值	R_{\min} 方差	$R_{\min}>10\%$ 的项目数	R_{\min} 最大可能阈值 T_{\max}	R_{\max} 均值	R_{\max} 方差	$R_{\max}>10\%$ 的项目数	R_{\max} 最大可能阈值 T_{\max}
LCOM1p	0.119 7	0.076 5	57	0.096 3	0.146 0	0.093 5	72	0.108 1
LCOM2p	0.326 1	0.100 8	112	0.301 8	0.106 9	0.078 9	47	0.076 9
LCOM5	0.059 3	0.047 8	13	0.041 9	0.020 1	0.032 4	3	0.009 7
Coh	0.043 1	0.048 0	8	0.024 0	0.059 2	0.047 8	13	0.041 9
TCC	0.136 8	0.090 2	67	0.104 8	0.133 5	0.077 3	74	0.107 8
LCC	0.136 8	0.090 2	67	0.104 8	0.218 5	0.095 5	102	0.186 2
DCD	0.068 6	0.056 2	21	0.047 3	0.155 5	0.081 6	89	0.130 1
DCI	0.068 6	0.056 2	21	0.047 3	0.369 6	0.119 5	112	0.331 3
CAMC	0.029 9	0.045 5	6	0.013 6	0.018 2	0.033 5	4	0.008 1
NHD	0.079 4	0.066 3	26	0.057 4	0.038 4	0.048 7	7	0.020 5
CC	0.146 0	0.093 5	72	0.108 1	0.078 4	0.061 7	19	0.056 8
LSCC	0.135 8	0.091 1	66	0.102 1	0.063 0	0.048 4	15	0.045 1
SCOM	0.146 0	0.093 5	72	0.108 1	0.071 0	0.053 4	21	0.052 7
SCC	0.075 7	0.056 5	28	0.051 0	0.008 3	0.011 7	0	0.003 0

表 6 给出了对每个项目非特殊类的度量数据分布进行拟合的 p 值统计情况,与表 3 比较可以看到:

- (1) 规模度量和非规范化内聚度量仍可适用幂律分布和对数正态分布模型进行拟合;
- (2) 指标 SCC 的对数正态分布拟合的 Fisher 检验为 1,这表明,可一致地认为 112 个项目的 SCC 数据都符合对数正态分布.而且指标 SCC 的幂律分布拟合虽然 Fisher 检验的 p 值小于 0.05,但其最大显著水平阈值为 0.498,远大于 0.05,表明多数项目(超过 65 个项目)的幂律分布拟合 p 值大于 0.498,说明,如果只考虑非特殊类,SCC 的数据符合幂律分布的情况比较常见;
- (3) 其他基于相似度的内聚度量,包括 NHD、CC、LSCC 和 SCOM 的对数正态分布拟合的最大显著水平阈值都远大于 0.05,说明,如果只考虑非特殊类,这些指标的多数项目数据都可使用对数正态分布拟合.

Table 6 Results of fitting the data of non-special classes into power-law and lognormal distributions

表 6 各个度量指标的非特殊类数据分布模型拟合的 p 值统计

度量指标	有效 p 值数	对数正态分布拟合 p 值大于 0.05 的项目数	对数正态分布拟合的 α_{max}	对数正态分布拟合 Fisher 汇总 p 值	幂律分布拟合 p 值大于 0.05 的项目数	幂律分布拟合的 α_{max}	幂律分布拟合 Fisher 汇总 p 值
IMPMTD	112	111	0.954 253 1	1	111	0.859 324 0	1
FLD	109	109	0.992 203 4	1	108	0.909 526 7	1
ELOC	112	111	0.932 904 4	1	109	0.785 576 1	0.999 999 8
LCOM1	112	111	0.926 301 3	1	111	0.759 545 4	1
LCOM2	110	110	0.910 423 3	1	109	0.776 784 0	1
LCOM1p	112	52	0.000 915 5	1.2348E-296	12	0.000 000 0	0
LCOM2p	110	54	0.016 975 0	1.4786E-214	15	0.000 000 0	0
LCOM3	110	106	0.984 289 7	1	108	0.890 352 1	1
LCOM4	101	95	0.970 242 4	0.999 998 1	98	0.942 182 9	1
LCOM5	112	74	0.099 552 6	5.3226E-82	86	0.442 464 8	4.9779E-35
Coh	112	80	0.231 865 5	1.8182E-42	45	0.005 642 6	2.2058E-237
TCC	112	58	0.016 234 3	4.1526E-149	27	0.000 074 0	0
LCC	111	32	0.000 016 1	0	10	0.000 000 0	0
DCD	112	48	0.004 137 0	1.0680E-197	23	0.000 000 8	0
DCI	110	10	0.000 000 0	0	2	0.000 000 0	0
TCCp	111	109	0.846 710 7	1	110	0.742 032 5	1
LCCp	111	106	0.658 948 1	0.977 356 6	108	0.613 348 0	0.999 910 6
DCDp	112	109	0.818 024 7	1	111	0.763 725 7	1
DCIp	112	106	0.610 713 6	0.961 600 3	108	0.644 280 6	0.999 973 3
CAMC	112	62	0.033 926 0	4.8020E-105	35	0.000 018 3	0
NHD	112	103	0.769 751 7	0.034 844 7	10	0.000 000 0	0
ICH	110	110	0.956 495 0	1	109	0.842 511 7	1
CC	112	90	0.281 772 0	5.3163E-30	53	0.011 595 6	1.0847E-166
LSCC	112	91	0.428 716 3	2.5988E-21	63	0.028 493 4	4.2735E-161
SCOM	112	88	0.321 980 6	1.6577E-31	58	0.026 104 8	5.5577E-185
SCC	112	108	0.895 736 3	1	96	0.498 429 9	0.000 128 7

简单地说,在排除特殊类后,即使都是规范化的内聚度量指标,基于相似度的内聚度量指标(包括 NHD、CC、LSCC、SCOM 和 SCC)的数据则会倾向于对数正态分布,而基于连通性的内聚度量指标(包括 TCC、LCC、DCD 和 DCI)则仍不能使用对数正态分布或幂律分布进行拟合.

图 1 汇总了各度量数据幂律分布和对数正态分布拟合 p 值大于 0.05 的项目数情况,其中的横线代表项目数达到 65.可以看到,软件规模度量指标和所有非规范化内聚度量指标的所有类和只考虑非特殊类的数据进行两种分布的拟合 p 值大于 0.05 的项目数都远大于 65.对于规范化内聚度量指标,在各种情况下,SCC 的数据分布拟合 p 值大于 0.05 的项目数都大于 65,因此,SCC 的数据符合对数正态分布或幂律分布的可能性最大.

图 2 汇总了各度量数据幂律分布和对数正态分布拟合 p 值的最大可能阈值(这意味着拟合 p 值大于该阈值的项目数仍大于等于 65)的情况.从图 2 可以得到与图 1 类似的结论,进一步可以看到,对于软件规模度量指标和所有非规范化内聚度量指标,对数正态分布的拟合 p 值大于幂律分布的拟合 p 值,说明这些指标数据的分布更适合使用对数正态分布拟合,而且从总体情况看,对于所有度量指标而言,只考虑非特殊类的数据的对数正态分布拟合具有最佳的效果.

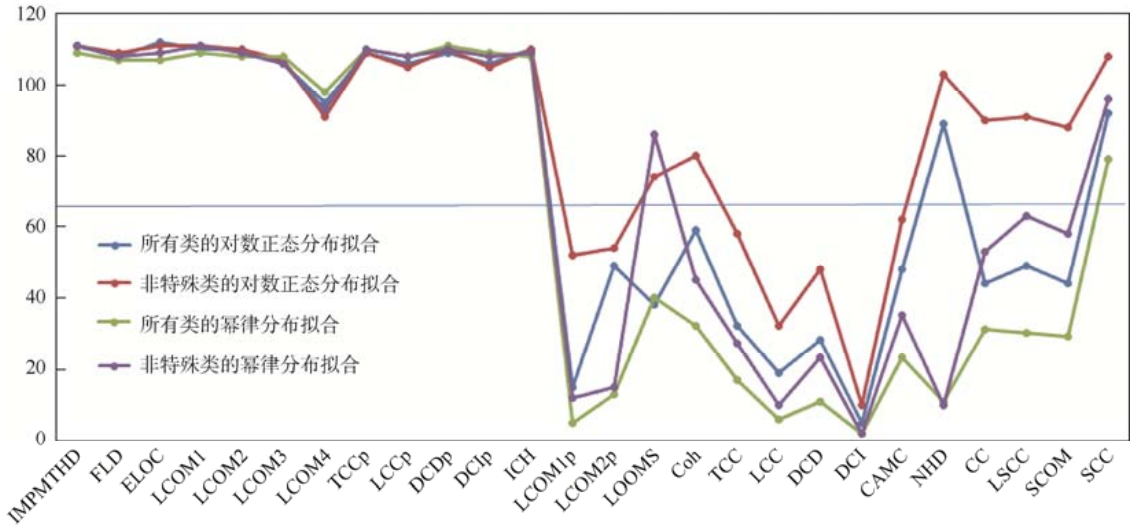


Fig.1 The number of projects with the fitting p -value greater than 0.05

图 1 各度量数据分布拟合 p 值大于 0.05 的项目数情况汇总

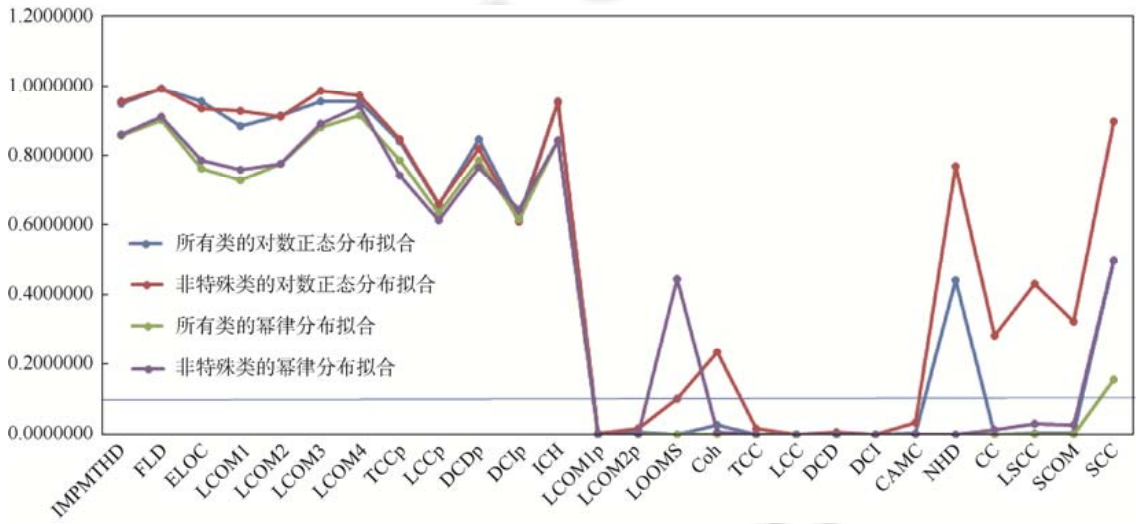


Fig.2 Maximal possible thresholds of fitting p -value

图 2 各度量数据分布拟合 p 值最大可能阈值情况汇总

通过对比非规范化的内聚度量指标 LCOM1、LCOM2、TCCp、LCCp、DCDp 和 DCIp 以及它们对应的规范化内聚度量指标 LCOM1p、LCOM2p、TCC、LCC、DCD 和 DCI 的分布拟合情况,我们发现,非规范化的度量指标数据都呈现幂律分布或对数正态分布这种重尾分布,但规范化的度量指标数据都没有明确的分布(我们使用 EasyFit 软件对一些项目的这些指标的数据进行了探索性研究,没有发现有合适的分布能够拟合这些指标的数据,EasyFit 所支持的 50 多种分布的拟合 p 值都远小于 0.05),而这些指标都是使用方法对总数进行规范化,这是否说明影响相应非规范化指标数据呈重尾分布的主要因素是类的方法数本身呈重尾分布?进一步地,已有文献发现不少耦合度量数据的分布也呈重尾分布,是否也是源于某种软件规模的效应?这些是在软件度量分布方面值得进一步探讨的问题。

3.4 实证研究结果总结

我们对3个软件规模度量以及21个内聚度量指标(包括 LCOM1 和 LCOM2 的规范化版本,以及 TCC 等非规范化版本)在112个Java开源项目的数据分布情况进行了分析,使用R软件程序包 `powerLaw` 对每个度量指标的每个项目的数据使用离散或连续幂律分布和对数正态分布,在估计最佳参数的情况下进行拟合,使用KS统计量计算得到的 p 值评估拟合的效果.我们对每个度量指标的112个项目数据拟合得到的 p 值进行了汇总分析,使用 Fisher 方法检验是否可以得到所有项目数据都符合所拟合的理论分布,并使用符号来检验每个度量指标的数据符合所拟合的理论分布是否常见.

拟合与检验的结果总结如下.

- (1) 软件规模度量数据都可使用幂律分布和对数正态分布模型进行拟合;
- (2) 非规范化的内聚度量指标 LCOM1、LCOM2、LCOM3、LCOM4 和 ICH 的数据都可使用幂律分布和对数正态分布拟合;
- (3) 规范化的内聚度量指标数据都没有通过对数正态分布和幂律分布拟合 p 值的 Fisher 检验,不过,符号检验的结果表明,度量指标 NHD 和 SCC 的数据符合对数正态分布的情况比较常见;
- (4) 软件项目中存在许多方法数小于等于1或字段数等于0的特殊类导致规范化内聚度量指标数据中存在许多取理论最小值0或最大值1的类,我们看到,多数规范化内聚度量指标取最小值0的类占比超过20%,其中,LCOM2p 取最小值0的类占比甚至达到40%,而指标 DCI 取最大值1的类占比也超过30%;
- (5) 我们对排除特殊类之后的每个度量指标的每个项目的数据进行拟合,发现软件规模度量数据和非规范化的内聚度量指标数据的幂律分布和对数正态分布模型拟合仍可通过 Fisher 检验,而且基于相似性定义的规范化内聚度量指标,包括 NHD、CC、LSCC、SCOM 和 SCC 的数据符合对数正态分布的情况更为常见,其中,SCC 对对数正态分布拟合甚至可通过 Fisher 检验;
- (6) 我们对基于连通性定义的内聚度量指标 TCC、LCC、DCD 和 DCI 分析它们的非规范化版本(即不除以类的方法对总数)各个项目数据的幂律分布和对数正态分布拟合情况,发现它们像 LCOM1、LCOM2 一样,如果不进行规范化,则幂律分布和对数正态分布拟合都可通过 Fisher 检验.

3.5 实证研究结果应用

了解度量指标数据分布有助于人们对度量指标的理解和使用,例如可选择更合适的度量指标用于构建预测软件构件易错性、易变性或可重用性的模型.基于本文对内聚度量指标数据分布的实证研究结果,这里讨论在内聚度量阈值确定和在检测软件设计缺陷时内聚度量指标选择方面的初步应用.

内聚度量阈值计算.确定软件度量阈值常用方法,包括基于开发人员经验确定以及基于统计方法确定.基于统计方法确定度量阈值的基本原理与确定样本数据中离群值(outlier)类似,基于度量数据的均值与标准差,或者分位数确定^[2].例如,以均值加减1倍、1.5倍、2倍、3倍标准差的值作为阈值,又或者以20%、80%分位数作为阈值^[3,5].实证研究表明,基于统计方法所确定的类的一些度量阈值在类的易错性预测方面有一定的作用^[3,5].

基于均值和标准差计算阈值需要假定度量数据呈正态分布,由于实证研究表明多数软件度量数据呈幂律分布或对数正态分布,Shatnawi 等人提出将度量数据做对数变换后计算阈值^[5].本文实证研究则表明,规范化内聚度量数据通常不能使用幂律分布或对数正态分布拟合,只有在排除方法数小于等于1或字段数等于0的特殊类后,基于相似性定义的内聚度量的数据才趋向于对数正态分布.因此我们建议,在使用对数变换^[5]确定内聚度量阈值,或确定内聚度量相对阈值^[4]时,对于基于相似性定义的内聚度量指标 CC、LSCC、SCOM 和 SCC,应该排除方法数小于等于1或字段数等于0的特殊类,而对于基于连通性定义的内聚度量指标 TCC、LCC、DCD 和 DCI,应该针对它们的非规范化的指标数据进行分析.

具体到度量阈值的计算,这里以文献[5]给出的方法为例:对于可使用对数正态分布拟合的度量指标数据,将数据进行对数变换,然后计算其均值 μ 与标准差 σ ,以 $\mu \pm 2\sigma$ 或 $\mu \pm 3\sigma$ 为判断内聚度偏高或偏低的标准(统计学结果表明,正态分布的数据以少于5%的概率落在均值加减两个标准差范围之外).

以项目 JHotdraw-7.5.1 为例,表 7 给出其非特殊类的 4 个基于相似性内聚度量指标数据的分布.这个项目总共有 532 个类,排除特殊类(方法数小于等于 1 或字段数等于 0 的类)后有 440 个类,但是由于只能对大于 0 的数据做对数变换,因此每个度量指标的有效类数(可进行对数变换的数据数)并不相同.表中最小值、 Q_1 、中位数、 Q_3 、最大值、均值和标准差是基于对数变换后的数据计算,但为便于比较,已经经过指数变换回到原数据范围(即 0~1 之间).可以看到,每个指标数据的中位数与均值都很接近,说明对数变换后的数据呈对称分布.表中的列“阈值 1”给出均值减两个标准差算出的值,“阈值 2”给出均值减 3 个标准差算出的值(注意是表中均值和标准差的自然对数相减,然后做指数变换得到).“类数 1”给出小于等于“阈值 1”的类的个数,“类数 2”给出小于等于“阈值 2”的类的个数.由于实践中通常关注内聚度偏低的类,这里只计算均值减标准差之后给出的阈值.

Table 7 The distributions on similarity-based cohesion metrics in JHotdraw-7.5.1

表 7 项目 JHotdraw-7.5.1 的基于相似性内聚度量指标数据的分布及阈值

度量	有效类数	最小值	Q_1	中位数	Q_3	最大值	均值	标准差	阈值 1	类数 1	阈值 2	类数 2
CC	347	0.001 7	0.060 5	0.136 4	0.318 1	1	0.130 4	3.243 9	0.012 4	10	0.003 8	3
LSCC	348	0.000 4	0.022 8	0.055 6	0.200 0	1	0.062 1	4.704 0	0.002 8	6	0.000 6	1
SCOM	347	0.000 8	0.038 5	0.094 4	0.316 7	1	0.095 9	4.284 5	0.005 2	13	0.001 2	1
SCC	404	0.001 2	0.030 6	0.050 4	0.107 4	0.6	0.053 0	2.757 5	0.007 0	12	0.002 5	3

从表 7 可以看到,JHotdraw-7.5.1 中内聚度偏低的类并不太多,实际上,分别只有 10 个、6 个、13 个和 12 个类在 CC、LSCC 和 SCOM 取值小于“阈值 1”,合计只有 31 个不同的类在这 4 个指标的某一项或多项指标小于“阈值 1”,而只有 6 个不同的类在这 4 个指标的某一项或多项指标小于“阈值 2”,其中,类 org.jhotdraw.color.ColorSliderUI 在 CC、LSCC 和 SCOM 上都取表中的最小值,SCC 值也只有 0.039 8,可以说是从这几个指标看内聚度最低的类,值得开发人员进一步关注.

我们也对其他项目的内聚度量阈值进行了初步计算,发现:(1) 对于其他规范化度量,包括 TCC 等基于连通性的度量,由于它们的数据通常不能使用对数正态分布拟合,取最小值和最大值的数据个数占比很大,其阈值很难确定;(2) 对于非规范化度量,使用对数变换往往只能给出上界阈值(下界阈值往往小于 0),因此,由于实践中通常关注内聚度偏低的类而只适用于 LCOM 等内聚度缺乏度量指标,而适用于 TCCp 等基于连通性度量指标的非规范化版本.

内聚度量指标的选择.通过对度量指标数据的分布以及取最大值最小值的分析,还可以看到,从具有好的统计性质角度来看,指标 LCOM1 比 LCOM2 要好,TCC 比 LCC 要好,而 DCI 比 DCD 更好.这个结论与现有的考察度量在类的易错性^[6]、易变性^[7]以及对内聚度量指标的区分能力^[20]方面所得到的结论相一致.在对内聚度量指标数据的分析中我们还发现,基于相似性定义的内聚度量指标 CC、LSCC、SCOM、SCC 通常具有很小的中位数和均值,例如,在多数项目中,指标 SCC 数据的中位数小于 0.15,这意味着多数项目中的多数类具有很小的 SCC 值,这无疑会影响这些指标的区分能力.

Lanza 和 Marinescu 使用类内聚度量指标 TCC 检测“上帝类”^[20],它是指在系统中集许多功能于一身的类,由于承担太多的职责(因此也很难有高的内聚度),被认为是一种软件设计缺陷.Lanza 和 Marinescu 将 TCC 值小于 1/3 作为检测“上帝类”的规则之一,但根据我们上面的分析,这个阈值太大,在实践中基本上不会起任何作用.从表 4 可以看到,我们所研究的 112 个项目中有 107 个项目的 TCC 取最小值 0 的类占总类数比超过 10%.具体到 JHotdraw-7.5.1,532 个类的 TCC 值中位数才 0.144 2,也即超过 50%的类的 TCC 值远小于 1/3.Lanza 和 Marinescu 假定 TCC 的数据呈正态分布,但根据我们上面的分析,这个假定并不成立.

因此我们建议,选择统计性能更好的基于相似度的内聚度量,如 CC、LSCC、SCOM 或 SCC 用于检测软件设计缺陷,这样可根据软件项目内聚度量数据的分布自适应地确定类内聚度是否偏低的阈值(而不是固定的 1/3).例如,可选择基于对数变换计算出的均值作为阈值.由于这些度量指标的数据通常符合对数正态分布,因此这可保证小于该阈值的类确实具有偏低的内聚度.

开源工具 iPlasma(<http://www.loose.upt.ro/reengineering>)使用 Lanza 和 Marinescu 提出的规则检测“上帝类”,我们对其检测出的项目 JHotdraw-7.5.1 中的 25 个疑似“上帝类”的 TCC 值和 CC 值进行了初步分析:(1) 这些类

的 TCC 值均小于 0.3,其中有两个类的值为 0,在大于 0 的类中,类 `org.jhotdraw.color.ColorSliderUI` 也取最小的 TCC 值;(2) 这些类的 CC 值除两个类外,其他均小于表 7 给出的均值,而这两个类的 TCC 值均大于 0.25,其中,类 `org.jhotdraw.draw.liner.SlantedLiner` 只有 7 种方法 1 个字段,很可能并不是“上帝类”,另外一个类也只有 30 种方法 2 个字段.因此,基于初步分析结果我们认为,内聚度量 CC 比 TCC 更适用于检测内聚度偏低的类.当然,如何选择和利用内聚度量检测软件设计缺陷还需要我们作进一步的研究.

3.6 实证研究的局限性

影响我们实证研究结论有效性的最主要因素是所分析的软件项目集——Qualitas Corpus,这个集合中只包含 Java 开源项目集,除了少数几个项目规模比较大之外,主要由中等规模和小规模的 Java 软件构成.这个集合中的项目是否能作为 Java 软件的典型代表需要进一步研究,但 Qualitas Corpus 是目前不少实证研究工作的数据集^[4,11,15-17,31],使用这个集合中的项目进行研究有一定合理性.我们的结论对于该集合中的项目有效,是否对于其他的 Java 项目乃至其他语言的项目也有效有待进一步考察.目前只有少数对不同语言的软件度量分布进行比较^[17,33],不同语言的软件的度量数据是否具有显著不同仍需进一步加以实证研究.

Zhang 等人对软件项目的应用领域、编程语言、存在年限、开发周期、系统规模、系统变化数、系统下载数这 6 个因素对软件度量分布的影响进行了研究^[34].这些研究的内聚度量包括 LCOM、ICH、TCC 和 LCC,发现项目的应用领域对内聚度量影响最大.Qualitas Corpus 包含软件开发平台、游戏、软件分析工具、应用服务器构件等多个领域的 Java 软件,因此在应用领域方面有一定的代表性.但它不包含基于 Android 平台的移动应用程序,这种项目由于运行平台的限制,在界面设计、运行能耗管理等方面有显著不同,在编码实现方面也更依赖 Android 平台提供的构件库,因此在模块内聚度方面与其他应用可能不同.分不同应用领域考察软件度量数据的分布情况是另外一个需要进一步研究的问题.

我们的分析基于 Java 软件的源代码(而不是二进制代码)的度量数据,这样操作可能存在的问题是一个开源软件项目的源代码范围有时并不容易确定,从 GitHub 或 SourceForge 下载的项目(特别是一些工具性项目,如 Ant,antlr 等)的源代码往往包含一些演示代码(demo)、例子程序(example)、测试用例(test case)等.我们基于 Qualitas Corpus 提供的元数据(meta data)排除了这些代码,而只分析项目的“核心”源代码.

度量数据计算的准确性也影响实证研究结论的有效性.由于没有开源工具实现本文所要研究的所有内聚度度的计算,因此我们基于 Eclipse 平台提供的抽象语法树生成功能实现了 Java 程序的这些内聚度量数据以及其他众多度量(包括规模、耦合度和继承度量)数据的收集.为保证度量计算尽可能地准确,我们将该工具收集到的代码规模数据与 Qualitas Corpus 元数据进行了对比,也利用一些小的程序进行了人工验证.经过对比和测试,我们认为该工具对度量数据的计算是准确的.

4 相关工作

人们对软件度量(或软件性质)在实际软件中的分布一直都有研究,最早可追溯到 20 世纪 70 年代 Knuth 对 FORTRAN 语言程序的实证研究.近 10 年来,随着复杂网络理论的兴起,人们对幂律分布等重尾(heavy-tailed)分布是否存在于软件中有非常浓厚的兴趣.

Wheeldon 和 Counsell 对 3 个 Java 开源系统(JDK 1.4.1、Apache Ant 1.5.3 和 Tomcat 4.0)的类的字段数、方法数、构造方法数以及类之间的 5 种耦合关系(继承、接口实现、聚合、参数类型、返回类型)进行了实证研究,发现字段数、方法数、构造方法数、子类个数、实现的接口数、接口被实现的类数、类被引用为成员数(即被其他类用作字段声明)、类的对象成员数(即字段为其他类类型的成员个数)、类作为方法参数类型数、参数为类型的引用数、将类作为返回类型的引用数和返回类的方法数等都符合幂律分布.本文使用回归分析,给出了这 3 个系统中 12 个幂律分布的幂律指数及其置信区间^[10].

Baxter 等人扩展了 Wheeldon 和 Counsell 的工作,使用幂律分布、对数正态分布和两参数的 Weibull 分布对 Qualitas Corpus 最早收集的 56 个 Java 开源系统的 18 个度量(除了文献[10]所考察的度量外,还包括 DO 一个类型依赖多少个类型才能被编译、DOinv 一个类型出现在多少个类型的实现、公有方法数、程序包包含的类

型数、方法字节码大小)的数据进行加权最小二乘拟合,发现其中多数度量支持幂律分布或对数正态分布^[11]。

Concas 等人对一个大型 Smalltalk 软件的类方法数、类实例变量数、类包含继承的实例变量数、类直接子类数、类所有后代数、实例变量名字数、方法名字数、方法调用次数、方法代码行数、类代码行数、类依赖图的入度、类依赖图的出度和规范化后的出度这 10 个性质进行了研究,发现它们符合幂律分布或对数正态分布^[12]。之后,Concas 对 2009 年 2 月发布的 Qualitas Corpus 中的 100 个 Java 开源项目中的 94 个(因其中 6 个只有 byte code),以及另外 7 个版本的 Eclipse 软件和 10 个版本的 Netbeans 进行了实证研究,对源代码行数、类依赖图的扇入、类依赖图的扇出以及 6 个 CK 度量和 9 个来自社会网络分析的软件度量的数据进行了分析,发现它们多数符合对数正态分布或幂律分布,并且认为不同项目的同一度量数据具有十分类似的分佈^[15]。

Ferreira 等人对从 SourceForge 下载的 40 个 Java 开源系统,总数超过 26 000 个类的耦合因子(COF)、公有字段数、公有方法数、LCOM、继承数深度(DIT)以及相当于类依赖图入度的 afferent couplings 这 6 个度量指标的数据使用 EasyFit 软件进行分布拟合,发现其中 DIT 可使用泊松分布进行拟合,而其余 5 个度量数据则可使用同样是重尾分布的双参数 Weibull 分布拟合^[3]。Ferreira 等人还在考察度量数据分布的基础上,探讨了度量阈值的确定。

Louridas 等人对包括 J2SE SDK、Eclipse、OpenOffice、BEA Weblogic 等在内的 Java 软件,以及 Linux、Tex、Ruby 等系统的类依赖图的出度和入度进行了考察,发现幂律分布在这些软件中广泛存在^[13]。Zhang 等人以 Eclipse 为实证研究对象,考察了 Java 文件源代码行数 LOC 的分布,发现它符合对数正态分布^[14]。Herraiz 等人对 Qualitas Corpus 的 20101126r 版本中的 69 个 Java 开源项目的 6 个 CK 度量的数据使用极大似然估计(MLE)和 KS 拟合优度检验进行分析,发现 WMC、CBO 和 RFC 更适合使用双 Pareto 分布描述(小于某个阈值 x_{min} 的数据符合对数正态分布,而大于该阈值的数据符合幂律分布),而 NOC 和 LCOM 适合使用幂律分布描述,DIT 既不能使用幂律分布也不能使用对数正态分布拟合^[16]。

最近,Destefanis 等人使用 10 个来自 Qualitas Corpus 的 Java 开源软件和 10 个 Python 软件对类的个数、声明的实例方法数、声明的实例变量数、局部方法数、方法总数、代码行数、注释行数、语句数、继承树深度这 9 个度量的数据分布进行了研究,并重点基于局部方法数、方法总数和语句数这 3 个度量的数据分布比较了 Java 软件和 Python 软件的不同,发现这 3 个度量的数据都可使用对数正态分布拟合,并认为分布的不同参数可用于区分 Java 软件和 Python 软件^[17]。

可以看到,当前虽然已经存在不少对面向对象软件度量指标数据分布的研究,但迄今为止,除了对类的内聚度缺乏度 LCOM 的数据分布有研究之外,对于多数内聚度量指标数据的分布都没有相关的研究。但不少有关面向对象软件度量在预测类的易错性、易变性等方面的实证研究都表明 LCOM(包括 LCOM1、LCOM2、LCOM3、LCOM4 和 LCOM5)不是好的内聚度量指标,因此,研究其他内聚度量指标数据的分布情况很有必要。Al Dallal 对许多内聚度量指标的可用性(applicability)^[28]、区分度(discriminative power)^[20]等进行了深入研究,但却没有报告这些内聚度量指标数据的分布情况。

本文工作除了对内聚度量数据的分布进行研究之外,与上述对度量数据分布的研究还有一个不同点就是,我们使用 R 软件程序包 powerLaw 的每个度量的每个项目的数据使用极大似然估计法来估计最佳参数,以使用最佳参数的对数正态分布和幂律分布做拟合并计算 KS 检验统计量及其 p 值,并对每个度量的 112 个项目数据的拟合 p 值使用荟萃分析中的 p 值汇总的 Fisher 检验进行检验,以及使用符号检验对项目数据符合所拟合的理论分布是否常见进行了检验。我们的研究以大量项目的拟合作为基础,并进行了汇总分析,从而可对度量数据是否符合所拟合的理论分布得到更可靠的结论。

5 结论与展望

了解度量数据的分布对于理解和使用软件度量有重要意义,特别地,在基于软件度量评测和预测软件质量时,确定度量的阈值或参考值非常重要,而度量阈值和参考值的确定需要基于度量数据的分布。目前已经存在许多关于面向对象软件规模、耦合度和继承维度的软件度量数据分布的研究,可以发现,它们往往符合幂律分布

或对数正态分布,不少确定软件度量阈值的方法也基于度量数据符合这种重尾分布的假定.然而,对于面向对象软件内聚度量,人们甚少报告除 CK 度量中的 LCOM 外的内聚度量数据的分布情况,而 LCOM 被不少实证研究证明并不是好的内聚度量指标,因此研究更多的内聚度量数据的分布很有必要.

我们以 Qualitas Corpus 中的 112 个 Java 开源软件项目作为实证研究对象,考察了包括内聚缺乏度、基于连通性的内聚度和基于相似性的内聚度共 21 个内聚度量指标数据的分布情况,使用 R 软件程序包 `powerLaw` 估计最佳参数,并以使用最佳参数的对数正态分布和幂律分布进行拟合,计算 KS 检验统计量及其 p 值.对每个度量的 112 个项目数据拟合的 p 值使用荟萃分析中的 p 值汇总的 Fisher 检验进行检验,以确定是否能够一致地认为所有项目数据都符合所拟合的理论分布.对每个度量的 112 个项目数据拟合的 p 值是否大于给定显著水平 0.05 做符号检验,以确定 p 值大于给定显著水平 0.05 是否在这些项目中更为常见.

我们的分析表明,对每个项目的所有类的内聚度量数据而言,非规范化的内聚度量都可使用幂律分布和对数正态分布进行拟合,但规范化的内聚度量指标,除 NHD 和 SCC 使用对数正态分布拟合的 p 值大于 0.05 更为常见外,其他规范化度量指标数据均不能认为符合幂律分布或对数正态分布.我们分析了这些规范化度量指标数据的取值情况,发现在软件项目中这些指标取最小值或最大值的情况相当常见,尤其是 LCOM2 取最小值和 DCI 取最大值的类在绝大多数项目中都占总类数 30% 乃至 40% 以上.由于方法数小于等于 1 或字段数为 0 的类通常被赋予最小值或最大值,因此我们对每个项目中排除这些特殊类之后的每个度量数据分布进行了拟合,发现基于相似性的内聚度量指标(包括 CC、LSCC、SCOM 和 SCC)数据在这种情况下更倾向于对数正态分布,但基于连通性的内聚度量指标(包括 TCC、LCC、DCD 和 DCI)数据仍不符合对数正态分布或幂律分布.进一步地,我们也对这些基于连通性内聚度量指标的非规范化版本的数据进行了拟合,发现它们可以使用对数正态分布或幂律分布进行拟合.

基于这些实证研究结果,我们建议,在确定内聚度量数据的阈值时,如果采用的方法需要假定度量数据符合重尾分布,则对于基于连通性内聚度量指标,应该针对其非规范化版本的数据进行分析,而对于基于相似性的内聚度量指标,应该针对排除特殊类之后的度量数据进行分析.

下一步的工作可从几个方面展开:(1) 对更多的软件项目和更多的内聚度量指标进行实证研究,特别地,可分项目应用领域进行实证研究,以进一步证实本文的结论;(2) Qualitas Corpus 也提供了一些 Java 开源项目的多个版本,可基于一个项目的多个版本研究内聚度量指标数据分布的演化情况;(3) 本文使用符号检验对多个项目的度量数据的统计量(如最大值、最小值等)的性质进行了检验,可以针对更多的度量指标(不限于内聚度量指标)的更多的统计量的性质作类似的研究,以更好地了解这些度量指标数据分布的性质.

References:

- [1] Fenton NE, Bieman J. *Software Metrics: A Rigorous and Practical Approach*. 3rd ed., CRC Press, 2015.
- [2] Lanza M, Marinescu R. *Object-Oriented Metrics in Practice*. Springer-Verlag, 2006.
- [3] Ferreira KAM, da Silva Bigonha MA, da Silva Bigonha R, Mendes LFO, Almeida HC. Identifying thresholds for object-oriented software metrics. *Journal of Systems and Software*, 2012,85(2):244–257.
- [4] Oliveira P, Valente MT, Lima FP. Extracting relative thresholds for source code metrics. In: *Proc. of the 2014 Software Evolution Week—IEEE Conf. on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE 2014)*. Antwerp, 2014. 254–263.
- [5] Shatnawi R. Deriving metrics thresholds using log transformation. *Journal of Software: Evolution and Process*, 2015,27(2):95–113.
- [6] Briand LC, Wüst J, Daly JW, Victor Porter D. Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software*, 2000,51(3):245–273.
- [7] Lu H, Zhou Y, Xu B, Leung H, Chen L. The ability of objectoriented metrics to predict change-proneness: A meta-analysis. *Empirical Software Engineering*, 2012,17(3):200–242.
- [8] Chidamber SR, Kemerer CF. A metrics suite for object oriented design. *IEEE Trans. on Software Engineering*, 1994,20(6): 476–493.

- [9] Succi G, Pedrycz W, Djokic S, Zuliani P, Russo B. An empirical exploration of the distributions of the Chidamber and Kemerer object-oriented metrics suite. *Empirical Software Engineering*, 2005,10(1):81–104.
- [10] Wheeldon R, Counsell S. Power law distributions in class relationships. In: *Proc. of the 3rd IEEE Int'l Workshop on Source Code Analysis and Manipulation (SCAM 2003)*. Amsterdam, 2003. 45–54.
- [11] Baxter G, Freen MR, Noble J, Rickerby M, Smith H, Visser M, Melton H, Tempero ED. Understanding the shape of Java software. In: *Proc. of the 21st Annual ACM SIGPLAN Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2006)*. Portland, 2006. 397–412.
- [12] Concas G, Marchesi M, Pinna S, Serra N. Power-Laws in a large object-oriented software system. *IEEE Trans. on Software Engineering*, 2007,33(10):687–708.
- [13] Louridas P, Spinellis D, Vlachos V. Power laws in software. *ACM Trans. on Software Engineering and Methodology*, 2008,18(1).
- [14] Zhang H, Tan HBK, Marchesi M. The distribution of program sizes and its implications: An Eclipse case study. *CoRR*, abs/0905.2288, 2009.
- [15] Concas G, Marchesi M, Murgia A, Pinna S, Tonelli R. Assessing traditional and new metrics for object-oriented systems. In: *Proc. of the 2010 ICSE Workshop on Emerging Trends in Software Metrics (WETSoM 2010)*. Cape Town, 2010. 24–31.
- [16] Herraiz I, Rodríguez D, Harrison R. On the statistical distribution of object oriented system properties. In: *Proc. of the 3rd Int'l Workshop on Emerging Trends in Software Metrics (WETSoM 2012)*. Zurich, 2012. 56–62.
- [17] Destefanis G, Ortu M, Porru S, Swift S, Marchesi M. A statistical comparison of Java and Python software metric properties. In: *Proc. of the 7th Int'l Workshop on Emerging Trends in Software Metrics (WETSoM@ICSE 2016)*. Austin, 2016. 22–28.
- [18] Briand LC, Daly JW, Wüst J. A unified framework for cohesion measurement in object-oriented systems. *Empirical Software Engineering*, 1998,3(1):65–117.
- [19] Al-Dallal J, Briand LC. An object-oriented high-level design-based class cohesion metric. *Information & Software Technology*, 2010,52(12):1346–1361.
- [20] Al-Dallal J. Measuring the discriminative power of object-oriented class cohesion metrics. *IEEE Trans. on Software Engineering*, 2011,37(6):788–804.
- [21] Al-Dallal J, Briand LC. A precise method-method interaction-based cohesion metric for object-oriented classes. *ACM Trans. on Software Engineering Methodology*, 2012,21(2):8.
- [22] Al-Dallal J. Fault prediction and the discriminative powers of connectivity-based objectoriented class cohesion metrics. *Information & Software Technology*, 2012,54(4):396–416.
- [23] Al-Dallal J, Morasca S. Predicting object-oriented class reuse-proneness using internal quality attributes. *Empirical Software Engineering*, 2014,19(4):775–821.
- [24] Basili VR, Briand LC, Melo WL. A validation of object-oriented design metrics as quality indicators. *IEEE Trans. on Software Engineering*, 1996,22(10):751–761.
- [25] Tempero ED, Anslow C, Dietrich J, Han T, Li J, Lumpe M, Melton H, Noble J. The Qualitas Corpus: A curated collection of Java code for empirical studies. In: *Proc. of the 17th Asia Pacific Software Engineering Conf. (APSEC 2010)*. Sydney, 2010. 336–345.
- [26] Hartung J, Knapp G, Sinha BK. *Statistical Meta-Analysis with Applications*. John Wiley & Sons, Inc., 2008.
- [27] Gibbons JD, Chakraborti S. *Nonparametric Statistical Inference*. 4th ed., Revised and Expanded, Marcel Dekker, Inc., 2003.
- [28] Al-Dallal J. Improving the applicability of object-oriented class cohesion metrics. *Information & Software Technology*, 2011,53(9):914–928.
- [29] Al-Dallal J. Effects of incorporating special methods into cohesion measurement on class instantiation reuse-proneness prediction. *IET Software*, 2014,8(6):285–295.
- [30] Al-Dallal J. Identifying refactoring opportunities in object-oriented code: A systematic literature review. *Information & Software Technology*, 2015,58:231–249.
- [31] Fontana FA, Ferme V, Zanoni M, Yamashita A. Automatic metric thresholds derivation for code smell detection. In: *Proc. of the 6th IEEE/ACM Int'l Workshop on Emerging Trends in Software Metrics (WETSoM 2015)*. Florence, 2015. 44–53.
- [32] Gillespie GS. Fitting heavy tailed distributions: The poweRlaw package. *Journal of Statistical Software*, 2015,64(2):1–16.

- [33] Wu D, Chen L, Zhou Y, Xu B. A metrics-based comparative study on object-oriented programming languages. In: Proc. of the 27th Int'l Conf. on Software Engineering and Knowledge Engineering (SEKE 2015). Pittsburgh: Wyndham Pittsburgh University Center, 2015. 272–277.
- [34] Zhang F, Mockus A, Zou Y, Khomh F, Hassan AE. How does context affect the distribution of software maintainability metrics? In: Proc. of the 2013 IEEE Int'l Conf. on Software Maintenance (ICSM 2013). Eindhoven, 2013. 350–359.



周晓聪(1971—),男,湖南常宁人,博士,副教授,CCF 高级会员,主要研究领域为软件工程,包括软件工程中的实证研究,软件测试与分析,软件工程的理论基础.



温剑丰(1975—),男,博士,讲师,主要研究领域为软件度量,软件估算,经验软件工程.



赖蔚(1992—),男,硕士,主要研究领域为软件度量,软件工程中的数据挖掘.

www.jos.org.cn

www.jos.org.cn