

DOI: 10.7652/xjtuxb201312001

软件更改可能性预测中的重要性度量研究

朱晓燕, 宋擒豹, 张本文
(西安交通大学计算机系, 710049, 西安)

摘要: 针对已有软件度量在对类进行衡量时没有考虑软件的整体结构的问题, 考虑到类在整个软件项目中的重要性, 提出了 2 个新的软件度量指标, 并以此构造了一种预测效果更好的软件更改可能性预测模型。新的软件度量指标将软件项目用图进行表示, 其中节点表示项目中的类, 边表示类之间的依赖关系; 在整个软件项目构建的图中对类的重要性进行研究, 从而构造预测软件更改可能性的重要性度量。选用了 6 个开源软件项目进行实验验证, 实验数据包括: ① 项目中每个类的软件度量, 包括 4 个基于大小的软件度量指标, 6 个基于复杂度的软件度量指标和 2 个基于重要性的软件度量指标; ② 项目中每个类在选定的历史阶段的更改行数。实验结果表明, 新的软件度量指标可使 5 个项目的精度得到提高, 平均提高 1.16%, 并使 6 个项目的曲线下面积得到提高, 平均提高 3.65%。

关键词: 软件度量; 更改可能性; 预测模型
中图分类号: TP311.5 **文献标志码:** A **文章编号:** 0253-987X(2013)12-0001-06

Research on Significance Metrics for Software Change-Proneness Prediction

ZHU Xiaoyan, SONG Qinbao, ZHANG Benwen
(Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Since existing software metrics do not consider the whole structure of a software project in evaluating classes, two new static metrics based on significance are proposed to build a better software change-proneness prediction model. The new metrics represent a project as a graph with classes being vertices and dependencies between classes being edges. The importance of a class is evaluated in the whole structure of the project, and the significance metrics for software change-proneness prediction are then proposed. Six open source software projects are used to validate the new metrics. Data used in experiments include two parts. The first part is metric of each class, including 4 volume metrics, 6 complexity metrics and 2 significance metrics, and the second part is the number of changed lines of each class during the selected history period. Experimental results show that when the new metrics are combined with other metrics, the accuracy increases on 5 projects by 1.16% on average, and the area under ROC curve increases on 6 projects by 3.65% on average.

Keywords: software metrics; change-proneness; prediction model

为了增加新功能、修复缺陷或者适应新环境, 软件被不断地更改, 变得越来越复杂, 需要花费越来越多的维护成本。因此, 对软件维护的有效方法的研究引起了很多学者的关注, 如预测维护成本^[1]、评估

可更改性^[2],以及在预测软件中可能发生更改的部分^[3]等。

预测更改可能性可以使软件开发者和维护者更有效地利用诸如同行评审、测试和检验等资源^[4],因此预测更改可能性是一种维护软件项目的有效方法。研究者已经提出了很多预测更改可能性的方法^[1,3-6]。在预测更改可能性的多种方法中,使用软件度量构造更改可能性预测模型是一种广泛使用的方法^[4,7]。静态软件度量不需要运行软件就可以直接得到,因此引起了很多研究者的关注^[1,8-10]。

然而,已有的软件度量只考虑类自身的属性或者类与类之间的直接关系对类进行衡量,没有在软件的整体结构中考虑类的重要性对类进行衡量。在一个软件项目中,某些部分包含比其他部分更重要的或者更显式的特征,因此这些部分比其他部分发生更改的可能性更大^[11],而一个类在整个软件项目中的重要性可能和它的更改可能性相关。忽略类在整个软件项目中的重要性会影响软件更改可能性预测的效果。

本文在研究中充分考虑了类在整个软件项目中的重要性构造新的软件度量指标,从而能够构造预测效果更好的软件更改可能性预测模型。

1 基于重要性的软件度量

1.1 定义

软件度量指标是对软件某一部分属性进行衡量的量化指标。传统软件度量指标包括基于大小和基于复杂度的软件度量指标,表 1 给出了这些软件度量指标及每个指标的含义。

表 1 基于大小和基于复杂度的软件度量指标

软件度量指标	含义
CountLine	总行数
CountLineCode	代码行数
CountLineComment	注释行数
CountLineBlank	空行数
CBO	对象类之间的耦合
NOC	子类的个数
RFC	一个类的响应集
DIT	继承树的深度
LCOM	类内聚缺乏度
WMC	每个类的加权方法数

从表 1 可以看出,基于大小和基于复杂度的软件度量指标主要从类自身的属性和类与类之间的直接关系对类进行度量,而没有考虑软件的整体结构对类进行度量。

在面向对象的软件中,一个类可能依赖于其他类,也可能被其他类依赖。因此,使用面向对象语言,如 Java 编写的项目可以用一个图或者网络表示,将软件项目中的类视为节点,类之间的关系视为连接节点的边。Kleinberg^[12]提出的 hubness 和 authority 反映了网页在网络中的重要性。本文将 hubness 和 authority 应用到软件构造的图中,反映类在整个软件项目中的重要性,从而得到了 2 个新的基于重要性的软件度量指标: H 和 A 。

因此,与传统的软件度量指标相比,新的软件度量指标 H 和 A 从新的角度对类进行度量,更好地反映类在整个软件系统中的重要性。

1.2 收集方法

基于重要性的软件度量指标收集方法包括以下 2 个步骤。

(1)构建图。Understand 工具提供了类之间依赖关系的信息。根据这些依赖信息,可以构建一个图。在构建的图中,节点表示类,节点之间的边表示类之间的依赖。

(2)计算每个类的 H 和 A 的值。使用 JUNG 工具^[13]提供的 HITS 算法计算图中每个节点的 H 和 A 的值。

首先给每个类 i 赋 2 个值 a_i 和 h_i ,分别表示类 i 的 A 和 H 的值。用 $i \rightarrow j$ 表示有一条类 i 到类 j 的依赖。 a 和 h 之间的递归关系可以表示为

$$h_i = \sum_{i \rightarrow j} a_j \tag{1}$$

$$a_j = \sum_{i \rightarrow j} h_i \tag{2}$$

开始,所有类的 h 和 a 都初始化为 1。然后重复使用式(1)、式(2)更新 h 和 a 的值。每次更新之后, h 和 a 的值都进行归一化。这个过程最终收敛于一个稳定的 h 和 a 的集合。

2 实 验

2.1 实验数据

在实验中,选用了 6 个开源软件项目:Ant、Hibernate、jEdit、Eclipse、iText 和 Framework。这 6 个开源软件项目的信息如表 2 所示。

表 2 开源软件项目信息

项目	起始版本	历史阶段
Ant	ANT_170	2010.02.02-2011.12.23
Hibernate	3.2	2010.10.13-2011.11.07
jEdit	Jedit-4-3-pre10	2007.07.11-2011.12.25
Eclipse	v_800	2009.02.24-2011.06.30
iText	iText_5_0_0	2008.03.27-2012.01.05
Framework	v3.0.0.M1	2009.12.16-2011.12.23

本文使用的实验数据包括 2 部分:一是每个类的软件度量,包括 4 个基于大小的软件度量指标,6 个基于复杂度的软件度量指标和 2 个基于重要性的软件度量指标;二是每个类在选定的历史阶段的更改行数。

本文使用 Understand 工具收集基于大小和基于复杂度的软件度量指标。对于更改行数,首先使用 MininGit 工具从软件版本控制库中提取信息并存入 MySQL 数据库中。然后从 MySQL 数据库中提取软件补丁,对每一个补丁统计更改的行数,包括增加、删除和更改的行数。更改的行在软件版本控制库中表示为先删除一个旧行再增加一个新行。增加的行是以“+”开头的行,删除的行是以“-”开头的行。因此,在一个补丁中,以“+”和以“-”开头的行的总行数即对应的类在该补丁中更改的行数。这样对每一个类,在选定的历史阶段中,它所对应的所有补丁中更改的行数的总和就是该类的总更改行数。这样就得到了一个项目中每个类在选定的历史阶段中的总更改行数。

2.2 数据预处理

收集到的数据并不能直接用于构造类的更改可能性预测模型,必须先进行如下 2 步的预处理。

(1)删除具有相同文件名的数据。如果使用文件的路径名加文件名结合起来作为文件的唯一标识符,那么在项目被重构时,文件名就发生了变化,这些文件之后的更改信息就丢失了。因此,只能使用文件名作为文件的唯一标识符。然而,有些文件有相同的文件名,如果使用文件名作为唯一标识符,这些文件就会被视为同一个文件,这些文件的更改行数的和就会被视为一个文件的更改行数。因此,需要删除那些具有相同文件名的数据,而只使用剩余的数据构造预测模型。

(2)分配类标签。对每一个项目,使用盒式图的方法找出更改行数的分界点。

设一个项目中所有类的更改行数的集合为 L , 首先将 L 从大到小排列,然后找出下四分位数 Q_1 和上四分位数 Q_2 ,接下来计算四分位间距 $I=Q_2-Q_1$,最后计算异常值截断点 $s=Q_2+1.5I$ 。 s 即为分界点,更改行数大于 s 的类的类标签为 change-prone,其他类的类标签为 not-change-prone。

2.3 新软件度量验证

2.3.1 相关分析 相关分析可以评估自变量和因变量之间的相关关系,它是一种验证自变量在因变量预测中作用的有效方法。Kaudell 的 tau 相关分析可以适应多个数据点取相同值的数据,而且分析不依赖于数据的分布^[7],因此本文使用 Kaudell 的 tau 相关分析评估软件度量和类的更改可能性之间的关系。

分析时,对每一个项目的数据集进行相关分析并计算 p 值。为了测试软件度量和类的更改可能性之间关系的重要性,使用 Holm-Bonferroni 方法^[14],并设显著性水平为 0.05。下面介绍具体的检验方法。

首先计算所有软件度量指标的 p 值,并对得到的 p 值按照降序进行排列,然后将每个软件度量指标的 p 值与 $0.05/k$ 进行对比,其中 k 是对应的软件度量指标在 p 值排序中对应的序号。对一个软件度量指标而言,原假设是它与类的更改可能性之间的关系是不重要的。如果一个软件度量指标的 p 值小于 $0.05/k$,那么拒绝原假设,接受备择假设,即该软件度量指标与类的更改可能性之间的关系是重要的。

对 6 个项目的软件度量与类的更改可能性相关关系的重要性测试结果如图 1 所示。为了使图看起来更清晰,所有大于 0.05 的 p 在图中都表示为 0.055。

从图 1 可以看出,大部分软件度量指标与类的更改可能性相关关系的 p 值都小于 $0.05/k$,每个项目中最多只有 2 个软件度量指标与类的更改可能性相关关系分析的 p 值大于 $0.05/k$ 。这说明大部分软件度量指标与类的更改可能性之间的相关关系都是重要的。

这些与类的更改可能性相关关系不具有统计重要性的软件度量指标,可能包含基于大小和基于复杂度的软件度量指标,也可能包含基于重要性的软件度量指标。为了具体分析这些软件度量指标中有多少是基于重要性的软件度量指标,表 3 给出了每个项目中与类的更改可能性相关关系不重要的软件度量指标及个数。

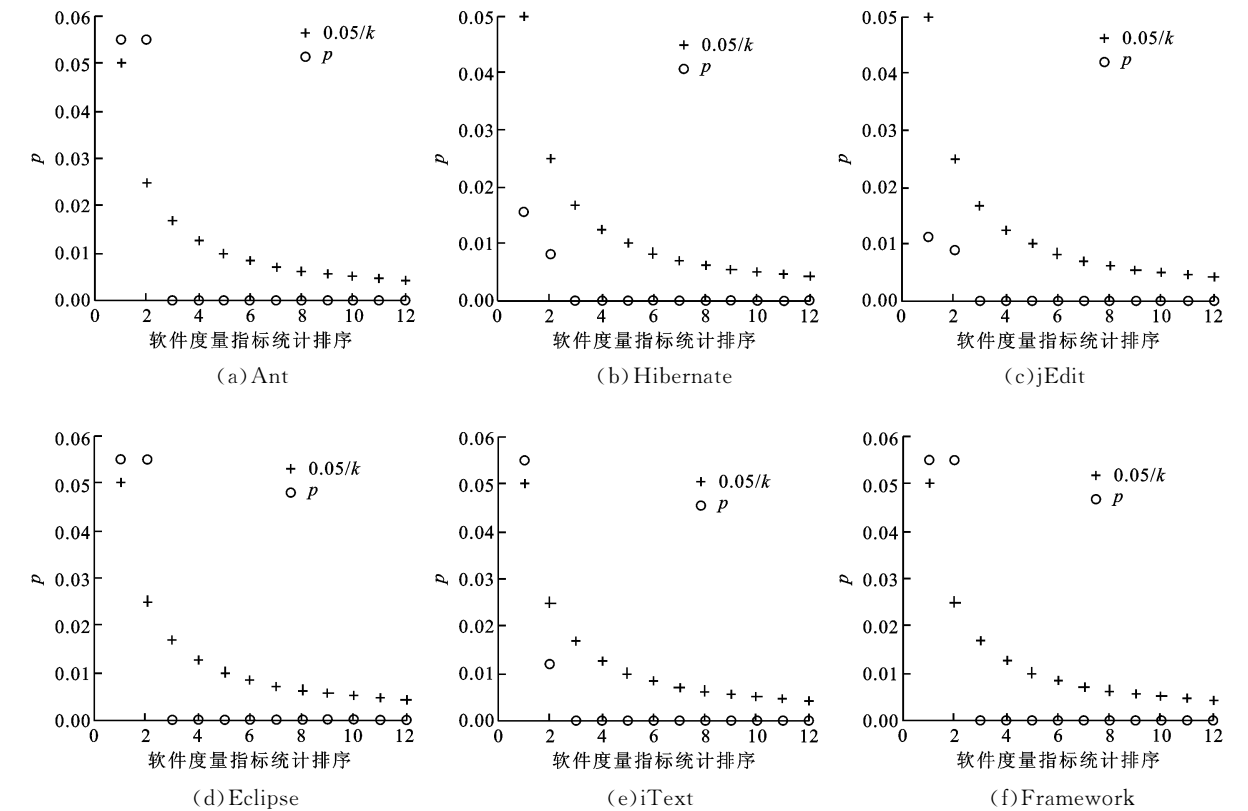


图 1 6 个项目的软件度量与类的更改可能性相关关系测试结果

表 3 与更改可能性关系不重要的软件度量指标

项目	软件度量指标	基于重要性的软件度量指标数	其他软件度量指标数
Ant	NOC, DIT	0	2
Hibernate		0	0
jEdit		0	0
Eclipse	NOC, DIT	0	2
iText	DIT	0	1
Framework	NOC, DIT	0	2

从表 3 可以看出:在选定的 6 个项目上,新的软件度量指标与类的更改可能性之间的相关关系都是重要的。除了 Hibernate 和 jEdit 外,在其他的项目上都至少有一个基于大小或者基于复杂度的软件度量指标与类的更改可能性之间的相关关系不重要。

表 4 属性选择算法选出的软件度量指标

项目	基于重要性的软件度量指标	其他软件度量指标	基于重要性的软件度量指标数	其他软件度量指标数
Ant	H	NOC, RFC, CountLineCode, DIT, LCOM	1	5
Hibernate	H	CBO, RFC, CountLine, CountLineCode, DIT, LCOM	1	6
jEdit	H, A	NOC, CountLineBlank, DIT, LCOM	2	4
Eclipse	H	RFC, CBO, CountLineCode, WMC	1	4
iText	H, A	RFC, CountLine, CountLineBlank, DIT, WMC	2	5
Framework		CountLine, CountLineBlank, CountLineCode	0	3

2.3.2 属性选择 属性选择可以选出对构造分类模型最有效的属性子集。选出的属性子集中的属性包含了因变量最重要的信息。如果某个软件度量指标在属性选择中被选中,说明该软件度量指标对类的更改可能性预测是有效的。

本文使用包装的属性选择方法,并使用 best first 搜索算法进行属性选择。对每一个项目选出的软件度量指标及个数如表 4 所示。

从表 4 可以看出:项目 jEdit 和 iText 的 2 个新的软件度量指标都被选出来了;项目 Ant、Hibernate 和 Eclipse 的指标 H 被选出来了;只有在 Framework 上,没有新的软件度量指标被选出来。属性选择的结果表明,基于重要性的软件度量指标在类的更改可能性预测中对 5 个项目是有效的。

2.3.3 对预测结果的影响 本文使用 Bagging 算法构建预测模型,并使用 C4.5 作为 Bagging 的基分类器。构建预测模型的过程如下:

- (1)使用重复抽样方法从训练数据集中生成 10 个新的训练数据集;
- (2)对每一个新的训练数据集,使用 C4.5 构建一个分类模型,共构建了 10 个分类模型;
- (3)对一个新的类,使用第 2 步中构建的 10 个分类模型分别预测它的类标签为 change-prone 或者 not-change-prone;
- (4)根据 10 个分类模型的预测结果,预测新类的类标签为 5 个以上分类模型的预测结果,如果有 5 个分类模型预测为 change-prone,其他 5 个预测为 not-change-prone,则预测新实例的类标签为设定的默认值 change-prone。

本文使用不同的软件度量指标构造了 2 个预测模型:model-0 和 model-1。model-0 使用基于大小的软件度量指标和基于复杂度的软件度量指标构造。model-1 使用基于重要性的软件度量指标、基于大小的软件度量指标和基于复杂度的软件度量指标构造。

为了对比上述 2 个模型,运行了 10 次十折交叉验证来获得更可靠的对比结果。为了对 2 个模型的分类结果进行评价,本文采用精度(P)和单位正方形曲线下面积(S)来描述分类结果,在每一折,计算预测模的 P 和 S 值。

表 5 给出了每个项目上每个模型获得的 100 个 P 和 S 的平均值 $\bar{P}$ 和 $\bar{S}$ 。

表 5 model-0 与 model-1 的预测结果				
项目	$\bar{P}$		$\bar{S}$	
	model-0	model-1	model-0	model-1
Ant	0.787 1	0.796 3	0.797 3	0.799 8
Hibernate	0.922 3	0.924 2	0.879 5	0.896 6
jEdit	0.851 1	0.881 8	0.774 5	0.833 6
Eclipse	0.873 7	0.880 6	0.829 1	0.875 2
iText	0.908 5	0.906 9	0.885 4	0.912 4
Framework	0.816 7	0.817 0	0.754 6	0.755 1

从表 5 可以看出:model-1 在所有项目上都比 model-0 获得更高的平均曲线下面积,平均提高了 3.65%;除了 iText 外,model-1 在其他项目上都比 model-0 获得更高的平均精度,平均提高了 1.16%。因此,可以说在大部分项目上,model-1 比 model-0

获得了更好的预测效果。
为了验证 model-1 与 model-0 预测效果之间的差异是否具有统计重要性,进行 wilcoxon 符号秩检验,结果如表 6 所示。

表 6 model-0 与 model-1 的预测效果的假设检验结果		
项目	精度假设检验值	曲线下面积假设检验值
Ant	0.000 5	0.22
Hibernate	0.054 1	0
jEdit	0	0
Eclipse	0.002	0
iText	0.798 8	0
Framework	0.614 9	0.893 9

从表 6 可以看出:对于 Ant、jEdit 和 Eclipse 项目,model-1 比 model-0 获得了更高的平均精度,并且它们之间的差异具有统计重要性;对于 Hibernate、jEdit、Eclipse 和 iText 项目,model-1 比 model-0 获得更高的平均曲线下面积,并且它们之间的差异具有统计重要性;在 iText 上,model-0 比 model-1 获得更高的平均精度,但是它们之间的差异不具有统计重要性。表 6 说明,加入新的软件度量指标可以在 3 个项目上提高精度,在 4 个项目上提高曲线下面积,并且提高具有统计重要性,在一个项目上降低了精度,但降低不具有统计重要性。因此,可以说加入新的软件度量指标可以获得更好的预测效果。

3 总 结

本文提出了 2 个新的用于软件更改可能性预测的软件度量指标,并且通过实验对这 2 个软件度量指标进行验证。首先给出了这 2 个新的软件度量指标的定义和收集方法,然后选定了 6 个开源软件项目对这 2 个新的软件度量指标进行验证。验证时使用了 3 种不同的方法:相关分析、属性选择以及对预测结果的影响。实验结果表明,新的软件度量指标与软件更改可能性之间的相关关系具有统计重要性,并且在属性选择时在大部分项目上都能被选出来。另外,通过加入新的软件度量指标构造的预测模型能够获得更好的预测效果。

参考文献:

[1] LI Wei, HENRY S. Object-oriented metrics which predict maintainability [J]. Journal of Systems and Soft-

- ware, 1993, 23(2): 111-122.
- [2] CHAUMUM M A, KABAILI H, KELLER R K, et al. A change impact model for changeability assessment in object-oriented software systems [C] // Proceedings of 3rd European Conference on Software Maintenance and Reengineering. Los Alamitos, CA, USA: IEEE Computer Society, 1999: 130-138.
- [3] AMOUI M, SALEHIE M, TAHVILDARI L. Temporal software change prediction using neural networks [J]. International Journal of Software Engineering and Knowledge Engineering, 2009, 9(7): 995-1014.
- [4] KORU A G, LIU Hongfang. Identifying and characterizing change-prone classes in two large-scale open-source products [J]. Journal of Systems and Software, 2007, 80(1): 63-73.
- [5] ZIMMERMANN T, ZELLER A, WEISSGERBER P, et al. Mining version histories to guide software changes [J]. IEEE Transactions on Software Engineering, 2005, 31(6): 429-445.
- [6] LINDVALL M, SANDAHL K. How well do experienced software developers predict software change? [J]. Journal of Systems and Software, 1998, 43(1): 19-27.
- [7] HAN A R, JEON S U, BAE D H, et al. Measuring behavioral dependency for improving change-proneness prediction in UML-based design models [J]. Journal of Systems and Software, 2010, 83(2): 222-234.
- [8] ZHOU Yuming, LEUNG H. Predicting object-oriented software maintainability using multivariate adaptive regression splines [J]. Journal of Systems and Software, 2007, 80(8): 1349-1361.
- [9] 陈振强, 徐宝文. 一种基于依赖性分析的类内聚度量方法 [J]. 软件学报, 2003, 14(11): 1849-1856.
- CHEN Zhenqiang, XU Baowen. An approach to measurement of class cohesion based on dependence analysis [J]. Journal of Software, 2003, 14(11): 1849-1856.
- [10] 阮彤, 沈备军, 居德华. 面向对象软件度量工具的软件结构 [J]. 计算机研究与发展, 2000, 37(4): 401-406.
- RUAN Tong, SHEN Beijun, JU Dehua. A flexible architecture in a measurement tool towards programs written in OO languages [J]. Journal of Computer Research and Development, 2000, 37(4): 401-406.
- [11] KITCHENHAM B A, PICKARD L M, LINKMAN S J. An evaluation of some design metrics [J]. Software Engineering Journal, 1990, 5(1): 50-58.
- [12] KLEINBERG J M. Authoritative sources in a hyper-linked environment [J]. Journal of the ACM, 1999, 46(5): 604-632.
- [13] O'MADADHAIN J, FISHER D, SMYTH P, et al. Analysis and visualization of network data using JUNG [J]. Journal of Statistical Software, 2005, 10(2): 1-25.
- [14] HOLM S. A simple sequentially rejective multiple test procedure [J]. Scandinavian Journal of Statistics, 1979, 6(2): 65-70.

[本刊相关文献链接]

- 鲍军鹏, 宋楠, 陶斌. 依赖知识的服务组合算法. 2013, 47(8): 1-6. [doi:10.7652/xjtuxb201308001]
- 吕正, 陈昊, 陈峰. 一种 ARM 存储模型的快速检测方法. 2013, 47(6): 68-72. [doi:10.7652/xjtuxb201306012].
- 蒋曹清, 应时, 文静, 等. 面向服务软件中异常处理的形式化建模方法. 2013, 47(4): 118-124. [doi:10.7652/xjtuxb201304020]
- 史旻, 邱劲锋, 侯迪, 等. 一种敏捷服务组合方法模型的研究与设计. 2013, 47(2): 1-6. [doi:10.7652/xjtuxb201302001]
- 李健, 黄庆佳, 刘一阳, 等. 云计算环境下的大规模图状数据处理任务调度算法. 2012, 46(12): 116-122. [doi:10.7652/xjtuxb201212020]
- 张峰, 韩燕波, 魏永山, 等. 一种用户为中心的即时服务组合方法. 2012, 46(12): 123-130. [doi:10.7652/xjtuxb201212021]
- 杜立新, 孟祥旭, 刘士军, 等. 基于服务提供者获益的服务响应选择方法. 2012, 46(11): 112-119. [doi:10.7652/xjtuxb201211021]
- 张曼, 段振华. 真并发等价性下的流程模型转换方法. 2012, 46(10): 42-47. [doi:10.7652/xjtuxb201210008]
- 邵必林, 边根庆, 张维琪, 等. 采用 k-均值聚类算法的资源搜索模型研究. 2012, 46(10): 55-59. [doi:10.7652/xjtuxb201210010]
- 聂艳明, 李战怀, 陈群. 针对不确定射频识别数据流的改进概率推导方法. 2011, 45(12): 45-52. [doi:10.7652/xjtuxb201112009]
- 陈峰, 李伟华, 陈昊, 等. 采用模型检测器的软件安全模型验证方法. 2011, 45(02): 15-20. [doi:10.7652/xjtuxb201102004]
- 李建勋, 解建仓, 陈田庆, 等. 计算网格下的高性能 MODIS 蒸散发反演研究. 2010, 44(10): 36-41. [doi:10.7652/xjtuxb201010007]
- 王森, 郝忠孝. 采用定性坐标的位置表达及主方向关系推理. 2010, 44(08): 36-42. [doi:10.7652/xjtuxb201008008]
- 冯少荣, 张东站. 高效的用户访问预测新算法. 2010, 44(04): 28-33. [doi:10.7652/xjtuxb201004007]

(编辑 刘杨 武红江)