跨项目软件缺陷预测方法研究综述

李 勇[1,2],刘战东[1],张海军[1]
(1. 新疆师范大学 计算机科学技术学院,新疆 乌鲁木齐 830054;
2. 新疆师范大学 数据安全重点实验室,新疆 乌鲁木齐 830054)

摘 要:软件缺陷预测是提高软件测试效率、保证软件可靠性的重要途径,已经成为目前实证软件工程领域的研究热点。在软件工程中,软件的开发过程或技术平台可能随时变化,特别是遇到新项目启动或旧项目重新开发时,基于目标项目数据的传统软件缺陷预测方法无法满足实践需求。基于迁移学习技术采用其他项目中已经标注的软件数据实现跨项目的缺陷预测,可以有效解决传统方法的不足,引起了国内外研究者的极大关注,并取得了一系列的研究成果。首先总结了跨项目软件缺陷预测中的关键问题。然后根据迁移学习的技术特点将现有方法分为基于软件属性特征迁移和软件模块实例迁移两大类,并分析比较了常见方法的特点和不足。最后探讨了跨项目软件缺陷预测未来的发展方向。

关键词:跨项目缺陷预测;迁移学习;软件属性特征;软件模块实例;模型训练
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2020)03-0098-06
doi:10.3969/j.issn.1673-629X.2020.03.019

Review on Cross-project Software Defects Prediction Methods

LI Yong[1,2],LIU Zhan-dong[1],ZHANG Hai-jun[1]
(1. School of Computer Science and Technology,Xinjiang Normal University,Urumqi 830054,China;
2. Key Laboratory of Data Security,Xinjiang Normal University,Urumqi 830054,China)

Abstract:Software defect prediction is an important way to improve the software testing efficiency and ensure software reliability,which has become a research hotspot in the field of empirical software engineering. In software engineering,the software development process or technology platform may change at any time. Especially when a new project is started or an old project is redeveloped,the traditional within-project software defect prediction method cannot meet the practical needs. Cross-project software defect prediction that using the cross-project labeled data and transfer learning technology can effectively solve the shortcomings of traditional method,which has attracted great attention of scholars at home and abroad,and produced a series of research findings. Firstly,the key problems of cross-project software defect prediction methods are summarized. Then,according to the technical characteristics of transfer learning,the existing methods are divided into two types,i. e.,the methods based on attribute characteristics and the methods based on software module instances,and the characteristics and shortcomings of common methods are analyzed and compared. Finally,the future development direction of cross-project software defect prediction is discussed.

Key words:cross-project defects prediction;transfer learning;software attribute characteristics;software module instance;model training

0 引 言

随着计算机软件规模和复杂度的日益增加,特别是大型系统对软件的强烈依赖,软件在运行过程中一旦失效可能导致严重的后果,有时甚至是致命的[1],导致软件失效的根本原因是系统中存在软件缺陷。通过软件缺陷预测技术对软件系统中可能存在缺陷的模块及其分布进行预测,可以有效提高软件测试的效率,对提高软件系统质量和保证软件可靠性具有重要意义[2]。

软件缺陷预测是指基于软件开发过程中积累的历史数据构建预测模型,对目标软件模块是否存在缺陷、缺陷严重程度或缺陷数量的分布等情况进行预测。通常情况下基于目标项目的历史数据,采用传统机器学习技术构建的模型可以获得理想的预测效果[3]。但在软件缺陷预测实践应用中,要进行预测的软件往往是新开发的项目,没有或只有较少的历史软件数据,而且

收稿日期:2019-05-03 **修回日期**:2019-09-09 **网络出版时间**:2019-12-05
基金项目:新疆自治区高校科研计划项目(XJEDU2017S031);新疆师范大学数据安全重点实验室招标课题(XJNUSYS102017B05)
作者简介:李 勇(1983-),男,讲师,博士,CCF 会员(20976G),研究方向为实证软件工程。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20191205.1104.018.html

进行数据标注的代价较高。虽然已有大量来自不同组织的软件缺陷数据集在互联网上公开,但不同软件项目间通常存在数据漂移问题,无法采用传统的学习算法训练模型[4]。

迁移学习是指可以采用与目标领域相关的数据训练模型,适用于目标领域没有历史积累数据的情况。基于迁移学习技术实现跨项目的软件缺陷预测是近几年该领域的研究热点之一,相关研究人员开展了大量工作,有力地促进了软件缺陷预测技术的发展。文中对该问题相关的研究文献进行综述。首先对相关的知识进行介绍,包括软件缺陷预测技术和迁移学习技术;然后对已有的研究进展进行分类评述;接着对研究中存在的问题以及未来研究工作进行展望;最后对全文进行总结。

1　相关研究背景

1.1　软件缺陷预测

软件缺陷预测是实证软件工程方向的一个活跃领域,通过对历史软件缺陷数据进行属性度量后,基于机器学习技术构建软件缺陷预测模型是目前研究者关注的热点。基于机器学习构建软件缺陷预测模型的前提是认为软件缺陷与软件内部度量属性之间存在某种关系,发现这些关系也成为软件度量发展的一个目标。现有研究文献在构建缺陷预测模型时使用的度量属性可以分为代码规模和复杂度度量、面向对象度量以及开发过程度量等。文献[5]指出各种常见的度量属性对于构建软件缺陷预测模型都是有效的,模型的性能取决于学习算法的选择。

随着机器学习技术的发展和应用,传统监督学习算法如朴素贝叶斯、决策树、随机森林、神经网络和集成算法等都被用来构建预测模型,各种算法在不同的应用环境中均取得了理想的预测性能[6]。也有文献针对软件缺陷预测中的特性问题进行深入的研究,如类数据的不平衡性[7]、代价敏感问题[8]和缺陷预测成本有效性[9]等。考虑到软件历史数据的标注代价问题,有研究提出了无监督或半监督的模型学习方式,如文献[10]将采用自适应阈值过滤的方法实现无监督的预测模型构建,文献[11]提出基于 LDS 算法的半监督模型构建方法。由于传统机器学习技术在构建模型时都假设训练数据与测试数据具有相同的特征空间和数据分布,所以现有研究中对模型的评价通常是基于目标项目数据构建模型,然后采用交叉验证的方式计算其准确率、召回率、AUC 等指标以实现模型的验证和比较。

然而在软件工程实践中,需要进行缺陷预测的往往是新开发项目的软件模块,而新开发项目一般没有或者只有少量的历史数据无法用于构建预测模型。现在互联网上已有许多专门用于软件缺陷预测研究的公开数据集,如创建于 2005 年的 PROMISE 软件工程预测模型数据库[12],其中的数据集均取自真实的软件项目。但是每个项目由于其不同的上下文环境导致软件缺陷数据的特征空间和分布不同,采用传统机器学习技术直接构建跨项目缺陷预测模型无法获得理想的预测效果[4]。

1.2　迁移学习

根据 NIPS 在 2005 年对迁移学习的定义,其目标是从相似但不同的领域直接进行知识的迁移,用于解决传统机器学习在遇到特征空间和分布变化时需要重新学习模型的问题[13]。由于迁移学习放宽了传统机器学习在构建模型时要求训练数据和测试数据独立同分布的假设,实现了目标领域缺乏训练数据时的模型构建。近年来,迁移学习技术获得了广泛的研究和关注。

在迁移学习中的领域 D 可以表示为 $D=\{\chi, P(X)\}$,其中 χ 为特征空间,$P(X)$ 为对应的边缘概率分布,$X=\{x_1,x_2,\cdots,x_n\}\in\chi$,$\boldsymbol{x}_i$ 为第 i 个实例的属性向量;学习任务表示为 $T=\{Y,f(\cdot)\}$,其中 Y 是标签空间,$f(\cdot)$ 是目标函数。迁移学习的目标是当 $D_s\neq D_t$ 或者 $T_s\neq T_t$ 时,使用 D_s 和 T_s 中有用的知识提高目标领域 D_t 的学习函数泛化性能,其中 D_s 表示源领域,D_t 表示目标领域,T_s 和 T_t 分别表示 D_s 和 D_t 对应的学习任务。在采用迁移学习进行模型构建时,D_s 中有足够的标注数据,而 D_t 中没有或者只有少量的标注数据。为了实现模型的迁移,可以通过特征迁移和实例迁移的方法实现[14]。基于特征迁移的方法是指通过特征变换的方式使源领域数据与目标领域数据分布最为接近,然后采用传统的学习算法训练模型,常见的特征迁移算法有 TPLSA[15]、MMDE[16] 和 TCA[17] 等。而实例迁移方法是根据源领域样例对目标领域的贡献程度形成新的训练数据用于目标领域模型的构建,常见的实例迁移算法有 TrBagg[18] 和 TrAdaBoost[19] 等算法。

目前迁移学习技术已经被广泛应用在各种跨领域学习任务中,如文本处理、图像处理和人工智能规划等[20]。文中关注的是软件缺陷预测领域,基于迁移学习的软件缺陷预测研究目标是解决跨项目软件缺陷预测模型的领域适应问题。在软件缺陷预测中,当 $D_s=D_t$ 且 $T_s=T_t$ 时,为传统的机器学习问题。当 $D_s\neq D_t$ 时,表示源项目和目标项目软件缺陷特征空间不同,如不同的开发上下文环境导致的 $X_s\neq X_t$ 或 $P_s(X)\neq P_t(X)$。对于 $T_s\neq T_t$,则表示缺陷预测任务不同,常见的软件缺陷预测任务包括预测模块是否存在缺陷、

缺陷的数量或缺陷的严重程度等。在跨项目的软件缺陷预测中,所关注的是学习任务相同 $T_s = T_t$,但领域特征分布不同,即 $D_s \neq D_t$ 的情况。也就是将从已有标注数据的源项目软件数据中学习到的知识迁移到与源项目的特征空间和缺陷分布不同的目标项目中实现缺陷的预测。

2 基于迁移学习的跨项目软件缺陷预测方法

2.1 跨项目软件缺陷预测的数据漂移问题

数据漂移是指在一定的模型评价指标下当训练数据和测试数据的联合分布改变时会导致预测模型性能发生变化,也就是说模型在不同领域数据间的预测性能具有很大差异[21]。在采用传统机器学习方法构建软件缺陷预测模型时,通常假设训练数据和测试数据服从相同的分布。例如训练集 $\{x_{\text{train}}, y_{\text{train}}\}$ 服从联合分布 $P(X,Y) = P(Y \mid X)P(X)$,然后通过某种学习算法构建模型得到 $\hat{P}(Y \mid X)$ 用于预测,为了验证该模型的性能一般采用未知标签的样例进行验证 $\hat{P}(y_{\text{test}} \mid x_{\text{test}})$。性能较好的模型要求 $\hat{P}(Y \mid X)$ 尽量逼近 $P(Y \mid X)$,可以通过多种评价指标进行计算。在一定的评价指标下,如果训练数据和测试数据来自不同的软件项目,会由于特征空间分布 $P(X)$ 或联合概率密度 $P(X,Y)$ 的不同影响到 $\hat{P}(Y \mid X)$,导致模型预测性能的下降。

在软件工程中,每个项目存在不同的上下文环境,如开发过程、开发者信息和程序语言等因素,导致软件缺陷数据的特征空间和分布不同,即跨项目的软件缺陷数据间存在数据漂移问题[22-23]。文献[4]采用 12 个软件项目的 622 个组合实现跨项目的预测,结果表明只有 3.4% 的组合可以获得较为理想的预测结果,进一步验证了软件项目间存在数据的漂移问题。因此,在跨项目的软件缺陷预测中必须对源项目和目标项目数据进行相应的处理才能获得理想的预测性能。采用迁移学习技术实现跨项目的软件缺陷预测流程如图 1 所示。下面将依据迁移学习的特点将现有研究文献中提出的方法分为软件属性特征迁移和软件模块实例迁移两类进行综述。

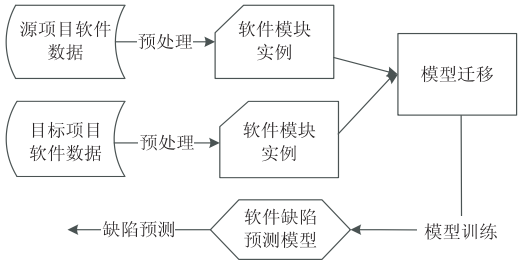

图 1 跨项目的软件缺陷预测流程

2.2 基于特征迁移的跨项目软件缺陷预测方法

现有研究文献在构建缺陷预测模型时使用的软件特征度量属性可以分为软件代码度量属性和开发过程度量属性两类。为了实现跨项目的缺陷预测,有研究者提出了基于特征迁移的跨项目软件缺陷预测方法。其主要思想是通过对软件缺陷数据属性特征进行相应的处理,使得源项目和目标项目在保证各自软件缺陷数据特性的同时其数据分布最为相似,然后采用传统的学习算法构建预测模型,实现跨项目的缺陷预测。根据对数据特征的处理方式不同,现有研究文献中提出的方法可归纳为特征选择、特征转换和特征映射三种。

2.2.1 基于特征选择的迁移方法

基于特征选择的方法是指找出源项目与目标项目软件缺陷数据的最小特征子集,该特征子集在保证源项目和目标项目软件缺陷特性的同时使得数据分布一致,从而实现软件缺陷预测模型的迁移。文献[24]在实验中通过迭代选择的方式获得使模型性能稳定的“最优属性子集”后,采用朴素贝叶斯和逻辑回归算法构建跨项目的缺陷预测模型,该模型在可接受指标范围(预测率大于 70%)内,实现了模型性能与模型构建成本的平衡。

基于特征选择的方法适用于源项目和目标项目软件缺陷数据采用相同的属性度量体系,但缺陷数据分布不同的情况。其优点是在模型构建中选择较少的度量属性,可以有效降低模型的成本。由于该方法实现模型迁移的前提要求源项目和目标项目数据存在潜在的属性子集,该属性子集对软件缺陷数据处理中度量属性的选择有指导意义。但该方法在实践中对源项目数据要求较高,而且由于所获取的属性子集对软件模块的缺陷特征描述有限,在实践应用中不一定会获得理想的性能。

2.2.2 基于特征转换的迁移方法

基于特征转换的模型迁移方法是指对源项目或目标项目的软件缺陷数据属性特征进行变换,使得跨项目的软件数据特征分布相似从而实现模型的迁移。在进行特征转换时,根据不同情况可以只对源项目或目标项目的缺陷数据属性特征进行转换,也可以对两者均进行转换。

文献[25]提出采用属性度量补偿的方式实现不同程序语言编写的跨项目软件缺陷预测。由于不同项目间的属性度量值范围不同,通过补偿的方式将源项目和目标项目的属性度量值调节至相似水平,结果表明可以有效提高预测模型的准确率和召回率。该方法要求源项目和目标项目的软件规模相似,该文献在研究中仅使用两个项目的数据集实验,说服力相对较弱。

而文献[26]则对源项目和目标项目软件数据属性特征均进行对数转换,使其取值范围相似,在转换的同时去除数据中的离群点,结果表明可以在跨项目软件缺陷预测中获得较好的预测结果。

基于特征转换的模型迁移方法适用于源项目和目标项目软件度量属性类似,但属性取值范围差异较大时的情况。而且在使用该方法的两篇文献中均采用相同开发组织的不同项目数据进行实验,当跨项目软件缺陷数据分布差异较大时,属性特征转换对数据分布的改变有限,因此该方法不适用于源项目和目标项目软件数据分布差异较大的情况。

2.2.3　基于特征映射的迁移方法

基于特征映射的迁移方法是指将源项目和目标项目的软件缺陷数据原始特征空间映射到某一潜在特征空间,在该潜在特征空间下,源项目和目标项目的数据分布最为接近,然后基于该潜在空间实现跨项目的软件缺陷预测模型迁移。该方法与特征选择和特征转换方法的区别在于映射得到的是全新的特征空间。

文献[27]提出基于迁移成分分析 TCA+的跨项目软件缺陷预测模型。TCA 是指在保证数据差异的约束条件下获得源项目和目标项目数据特征的线性映射函数,通过该函数将跨项目数据映射到潜在的特征空间实现模型的迁移。在实验中为了避免数据属性取值范围不同对模型性能的影响,提出 TCA+实现自动正规化算法,有效提高了预测模型的性能。文献[28]将跨项目软件缺陷预测模型迁移问题形式化为一个半正定矩阵,获得该半正定规划问题的解后采用 PCA 进行降维得到新的潜在特征空间,从而实现模型的迁移。

基于特征映射的迁移方法适用于源项目和目标项目的软件缺陷数据度量属性体系不同或差异较大时的情况。该方法的不足是在模型构建中没有使用软件缺陷信息,仅使用源项目和目标项目软件缺陷数据映射后的分布信息,对模型性能的提高有限。

2.3　基于实例迁移的跨项目软件缺陷预测方法

在软件缺陷预测中,待实现预测的软件程序单元可以在不同的粒度层次进行属性度量,如方法度量、类度量、包度量和文件度量等。文中将软件缺陷数据中的度量程序单元称为软件模块实例。基于实例的跨项目软件缺陷预测方法是指在软件模块实例层对源项目数据进行处理实现模型的迁移。根据对软件模块实例的处理方式不同,现有研究文献中提出的方法可以归纳为实例选择、实例权值和局部模型三种。

2.3.1　基于实例选择的迁移方法

基于实例选择的方法是指根据目标项目软件模块实例的特征从源项目中选择合适的标注实例构成模型数据,实现跨项目的模型迁移。从源项目数据中所选模块实例的质量决定着最终模型的预测性能。如果所选模块实例较少则不能充分反映目标项目的缺陷特征,容易导致预测率较低。但如果引入不相关的模块实例较多时,容易导致模型的误报率较高。

文献[29]提出 Burak 数据选择方法,对于目标项目中的每个软件模块实例使用 KNN 算法从多个源项目数据中寻找与其最接近的若干模块实例组成目标项目的训练数据。在该文献的实验中仅使用目标项目软件模块实例去引导源项目实例的选择,而没有关注源项目中的数据特性。当目标项目数据远小于源项目数据时,源项目数据可以提供更多的信息,即特征不相似的实例有可能包含对模型训练有用的信息。针对该问题文献[30]提出 Peters 数据选择方法,采用源项目数据寻找目标项目中与其最近邻的模块实例,将所有与目标项目最近邻的源项目实例构成模型训练数据,其结果要优于 Burak 方法,该方法的优点是在项目开发的早期阶段当目标项目数据较少时,也可以获得较好的预测性能。在这两篇文献中存在的不足是对于数据的选择均从源项目或目标项目的每个软件模块实例出发,当数据量增大时算法的运行时间呈指数增长,而且没有考虑数据的整体分布特征。

文献[31]提出在跨项目缺陷预测时,如果可以充分利用源项目和目标项目之间存在的分布特征相关性,从数据中获取先验知识指导源项目数据的选择可以很大程度提高模型的性能。为此文献[32]提出根据分布特征从大量源项目数据中获取目标项目的训练数据。该文献在实验中从项目数据层定义其特征属性,然后通过聚类的方式寻找多个源项目数据作为训练集,结果表明可以有效提高模型的预测率。但由于从项目数据层进行数据选择会引入不相关的软件模块实例,所以导致了模型综合评价指标仍然较低。也有研究者提出采用智能算法从多个源数据中寻找最优数据集的方法,如文献[33]使用遗传算法实现最优模型训练集的选择,然后采用集成的方式构建模型,有效降低了模型的误报率。

2.3.2　基于实例权值的迁移方法

基于实例权值的迁移方法是指根据源项目软件缺陷数据中每个模块实例对目标模型的作用不同分配权值,然后基于重新分配权值的数据实现模型的迁移,在现有研究文献中一般采用权值更新的方式。

文献[34]提出迁移贝叶斯 TNB 算法用于构建跨项目的软件缺陷预测模型。首先提取目标项目软件缺陷数据的分布特征,然后对源项目数据中的每个软件模块实例与目标项目数据特征进行比较,基于数据引力方法计算每个实例的权值,采用加权的训练数据构建模型。该方法的优势在于使用了源项目中所有软件

模块的信息,而且只需要计算一次目标项目数据的分布特征,降低了单个实例比较选择中运算量较大的问题,算法的时间复杂度是与数据量成线性关系。其不足是对目标数据信息考虑较少。类似的还有文献[35]提出代价敏感的 TrAdaBoost 算法实现模型迁移,该方法综合了软件缺陷数据的类分布不平衡以及不同误分代价的差异,结合实例迁移 TrAdaBoost 和代价敏感 AdaC2 算法实现。其优点是充分考虑到了不同误分代价,但在应用中代价值只能通过手工设置,不利于实践操作。

2.3.3 基于局部模型的迁移方法

基于实例选择和实例权值的迁移方法都是通过对源项目和目标项目软件模块实例的处理从而构建最终的全局模型。考虑到软件系统的复杂性,有研究提出在实现软件质量模型时,应该基于最小软件模块集合学习“专有规则”。基于局部模型的迁移方法是指根据相似性将目标项目软件数据分为多个簇,对每个簇选择相应的源项目数据构建局部模型,然后通过多个局部模型实现跨项目的缺陷预测。

文献[36]采用聚类算法形成源项目和目标项目软件模块实例簇,以簇为单位选择数据并构建局部模型,结果表明通过局部模型实现的预测性能要优于全局模型。文献[37]也取得了同样的结果,该文献从 38 个软件项目的 92 版本数据中实现相似项目的聚类,表明基于相似聚类簇训练的模型预测率较高于采用目标项目数据的模型。但也有研究者提出虽然局部模型的预测性能要优于全局模型,但多个局部模型性能平均后这种优势会抵消,仍然建议采用全局模型的方式实现预测[38]。

基于局部模型的跨项目预测方法结合了项目数据分布特征和实例特征,可以较好地反映目标项目的软件缺陷特征,通过多个局部模型进行预测可以获得较好的性能。但是该方法在每次进行缺陷预测时都要重新进行源项目和目标项目数据的聚类和局部模型训练,在实践应用中代价较高。

3 存在的不足及未来研究工作展望

3.1 跨项目软件缺陷预测模型性能的提高

基于特征迁移的方法在模型构建中对软件缺陷信息没有充分利用导致了模型性能提高有限,而基于实例迁移的方法在进行训练数据选择时对模块实例的数据分布特征考虑不足,容易造成算法运行效率低和模型预测率高但准确率低等问题。在未来的研究中应该将数据特征和实例选择结合起来,在充分考虑项目数据分布特征的前提下实现软件模块实例的选择,进一步提高最终模型的预测性能。另外如果在模型构建中可以结合目标项目软件模块的缺陷预测及修复实现部分数据的标注并有效利用,可以有效地避免现有模型方法容易造成过拟合的问题。

3.2 跨项目软件缺陷预测模型的评价

现有研究文献对跨项目软件缺陷预测模型的评价都是采用准确率、召回率和 F 值等传统的评价指标。文献[39]提出对于跨项目缺陷预测模型即使较低的准确率对软件测试和软件质量保证也能起到有效的作用。在未来的研究中针对软件缺陷预测实践中,如何使得模型构建成本与模型性能平衡,或者如何构建符合实践需求指标的模型直接决定着跨项目的预测模型构建方法和模型的评价,对于该问题需要进一步深入研究。

3.3 跨项目软件缺陷数据共享库构建

在实现跨项目数据驱动的软件缺陷预测中,数据的来源是必须考虑的问题。目前互联网上已经有许多公开的软件缺陷数据集用于实验研究,而且越来越多的研究者在公开自己采集处理后的软件缺陷数据。面对大量数据如何有效的组织管理并构建统一的软件缺陷数据共享库对实现跨项目软件缺陷数据选择和模型构建尤为重要。

3.4 跨项目软件缺陷知识的可解释性。

现有研究中实现跨项目软件缺陷预测的目的是为了对未知软件模块进行是否存在缺陷或者缺陷分布的预测。如果能从已有源项目软件缺陷数据中学习到具有可解释、可理解的规则,进行合理的组织选择并进行可视化后用于指导软件开发实践,不但可以更加有效地利用已积累的多源项目数据,而且可以找到提高软件质量的根本原因,对于软件工程的实践具有重要意义。

4 结束语

跨项目软件缺陷预测研究是目前实证软件工程领域的前沿方向。利用迁移学习技术实现跨项目的预测可以有效降低模型构建成本,从而提高软件测试效率和保证软件质量。文中对已有研究成果中提出的方法进行分类综述,并指出了其存在的不足和未来的研究方向,为完善跨项目的软件缺陷预测研究与应用提供了理论基础和技术参考。

参考文献:

[1] SELIYA N, KHOSHGOFTAAR T M, HULSE J V. Predicting faults in high assurance software[C]//2010 IEEE 12th international symposium on high assurance systems engineering. San Jose, CA: IEEE, 2010: 26-34.

[2] TIAN J. Software quality engineering: testing, quality assur-

ance, and quantifiable improvement[M]. [s. l.]: John Wiley & Sons, 2005.

[3] MENZIES T, MILTON Z, TURHAN B, et al. Defect prediction from static code features: current results, limitations, new approaches[J]. Automated Software Engineering, 2010, 17(4): 375-407.

[4] ZIMMERMANN T, NAGAPPAN N, GALL H, et al. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process[C]//Proceedings of the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering. New York: ACM, 2009: 91-100.

[5] MENZIES T, GREENWALD J, FRANK A. Data mining static code attributes to learn defect predictors[J]. IEEE Transactions on Software Engineering, 2007, 33(1): 2-13.

[6] HALL T, BEECHAM S, BOWES D, et al. A systematic literature review on fault prediction performance in software engineering[J]. IEEE Transactions on Software Engineering, 2012, 38(6): 1276-1304.

[7] 李　勇. 结合欠抽样与集成的软件缺陷预测[J]. 计算机应用, 2014, 34(8): 2291-2294.

[8] 李　勇, 黄志球, 房丙午, 等. 代价敏感分类的软件缺陷预测方法[J]. 计算机科学与探索, 2014, 8(12): 1442-1451.

[9] ARISHOLM E, BRIAND L C, JOHANNESSEN E B. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models[J]. Journal of Systems and Software, 2010, 83(1): 2-17.

[10] CATAL C, DIRI B. A fault detection strategy for software projects[J]. Tehnicki Vjesnik-Technical Gazette, 2013, 20(1): 1-7.

[11] CATAL C. A comparison of semi-supervised classification approaches for software defect prediction[J]. Journal of Intelligent Systems, 2014, 23(1): 75-82.

[12] MENZIES T, CAGLAYAN B, KOCAGUNELI E, et al. The promise repository of empirical software engineering data[DB/OL]. 2012-06-15. http://promisedata.googlecode.com.

[13] PAN S J, YANG Q. A survey on transfer learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.

[14] SHI Y, LAN Z, LIU W, et al. Extending semi-supervised learning methods for inductive transfer learning[C]//2009 ninth IEEE international conference on data mining. Miami, FL: IEEE, 2009: 483-492.

[15] PAN S J, KWOK J T, YANG Q. Transfer learning via dimensionality reduction[C]//Proceedings of the twenty-third AAAI conference on artificial intelligence. Chicago: AAAI Press, 2008: 677-682

[16] XUE G, DAI W, YANG Q, et al. Topic-bridged PLSA for cross-domain text classification[C]//Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. New York: ACM, 2008: 627-634.

[17] PAN S J, TSANG I W, KWOK J T, et al. Domain adaptation via transfer component analysis[J]. IEEE Transactions on Neural Networks, 2011, 22(2): 199-210.

[18] KAMISHIMA T, HAMASAKI M, AKAHO S. TrBagg: a simple transfer learning method and its application to personalization in collaborative tagging[C]//2009 ninth IEEE international conference on data mining. Miami, FL: IEEE, 2009: 219-228.

[19] DAI W, YANG Q, XUE G, et al. Boosting for transfer learning[C]//Proceedings of the 24th international conference on machine learning. New York: ACM, 2007: 193-200.

[20] 庄福振, 罗　平, 何　清, 等. 迁移学习研究进展[J]. 软件学报, 2015, 26(1): 26-39.

[21] QUIONERO-CANDELA J, SUGIYAMA M, SCHWAIGHOFER A, et al. Dataset shift in machine learning[M]. [s. l.]: MIT Press, 2009.

[22] TURHAN B. On the dataset shift problem in software engineering prediction models[J]. Empirical Software Engineering, 2012, 17(1): 62-74.

[23] WAHYUDIN D, RAMLER R, BIFFL S. A framework for defect prediction in specific software project contexts[C]//Proceedings of the third IFIP TC 2 central and east European conference on software engineering techniques. [s. l.]: [s. n.], 2008: 261-274.

[24] HE P, LI B, LIU X, et al. An empirical study on software defect prediction with a simplified metric set[J]. Information and Software Technology, 2015, 59: 170-190.

[25] WATANABE S, KAIYA H, KAIJIRI K. Adapting a fault prediction model to allow inter languagereuse[C]//Proceedings of international conference on software engineering. [s. l.]: [s. n.], 2008: 19-24.

[26] CRUZ A E C, OCHIMIZU K. Towards logistic regression models for predicting fault-prone code across software projects[C]//2009 3rd international symposium on empirical software engineering and measurement. Lake Buena Vista, FL: IEEE, 2009: 460-463.

[27] NAM J, PAN S J, KIM S. Transfer defect learning[C]//Proceedings of the 2013 international conference on software engineering. San Francisco, USA: IEEE, 2013: 382-391.

[28] 田　华, 蒲天银. 基于迁移学习的软件缺陷预测方法研究[J]. 西南师范大学学报: 自然科学版, 2014, 39(3): 90-95.

[29] TURHAN B, MENZIES T, BENER A B, et al. On the relative value of cross-company and within-company data for defect prediction[J]. Empirical Software Engineering, 2009, 14(5): 540-578.

[30] PETERS F, MENZIES T, MARCUS A. Better cross company defect prediction[C]//2013 10th working conference on mining software repositories (MSR). San Francisco, CA: IEEE, 2013: 409-418.

(下转第 121 页)