

基于 Adaboost 算法的软件缺陷预测模型

熊婧 高岩 王雅瑜

(工业和信息化部电子第五研究所软件质量工程研究中心 广州 510610)

摘要 将 Adaboost 算法应用到软件缺陷预测模型中是软件缺陷预测的一种新思路, Adaboost 算法原理通过训练多个弱分类器构成一个更强的级联分类器, 有效地避免了过拟合问题。通过采用美国国家航空航天局(NASA)的软件缺陷数据库的仿真实验, 分别对原始 BP 神经网络算法和 Adaboost 算法进行分析对比, 其中 Adaboost 的弱分类器采用神经网络。实验结果表明, Adaboost 级联分类器有效地提高了软件缺陷预测模型的预测性能。

关键词 软件缺陷, 软件缺陷预测, BP 神经网络, Adaboost, 级联分类器

中图分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.7.034

Software Defect Prediction Model Based on Adaboost Algorithm

XIONG Jing GAO Yan WANG Ya-yu

(Software Quality Engineering Research Center, CEPREI, Guangzhou 510610, China)

Abstract A new software defect prediction method was proposed in this paper, which used Adaboost cascade classifier as its prediction model. The principle of Adaboost algorithm is to train multiple weak classifiers and combine them into another stronger cascade classifier, which can avoid over-fitting problem effectively. In this paper, comparative experiments based on NSNA software defect data sets are carried out between the original BP network and Adaboost with the weak classifier of BP network. The experimental results show that, the software defect prediction model based on Adaboost cascade classifier can improve the prediction performance significantly.

Keywords Software defect, Software defect prediction, BP neural network, Adaboost, Cascade classifier

1 引言

随着信息技术的发展, 软件复杂程度不断提高, 软件规模不断增大, 良好的软件缺陷控制和预测机制可以帮助企业开发出高质量的软件产品; 防止由于软件系统的缺陷而导致的严重后果^[1]; 降低产品生产和维护的成本; 提高客户满意度; 建立良好的企业形象和增强企业在市场上的竞争力。因此, 软件质量越来越受重视, 如何预测软件缺陷及提高软件质量成为当今研究的热点之一。

国内外对软件缺陷预测模型进行了深入的研究并取得了一定的成果。其中, 较为主流的分类方法是基于统计学方法的传统软件缺陷预测模型和基于神经网络的软件缺陷预测模型, 传统的软件缺陷预测模型所采用的方法多是基于统计学的线性或者非线性回归^[2,3]及聚类分析^[4]等。基于统计学方法的软件缺陷预测模型不具备自学习的能力, 且具有一定的局限性, 无法全面考虑影响软件缺陷的各方面因素。常用的人工神经网络包括: 反向传播神经网络(BP)、学习矢量量化神经网络(LVQ)、模糊自适应学习控制网络(FALCON)。其中基于 BP 神经网络的软件缺陷预测模型的研究^[5-7]最为成熟, 目前已有较为成熟的理论与实践成果。而集成方法也逐渐在

软件缺陷预测中得到广泛应用, 包括 Bagging、随机森林、轮转森林以及随机子空间等。Stefan Lessmann 等人^[8]对传统的软件缺陷预测模型的建立方法(包括统计分类、最近邻方法、神经网络、支持向量机、决策树以及集成方法)进行了预测准确度的对比研究, 指出不存在最优分类方法, 认为几种方法分类预测准确度间不存在显著差异。Baljinder Ghotra 等人^[9]在 Stefan Lessmann 的研究基础上, 通过实验指出不同的预测方法对于特定的软件缺陷模型的建立存在统计学上的显著差异, 特别是集成方法的预测结果比单个简单方法的预测结果更为准确。

在软件缺陷预测研究中, 样本数据的不均衡分布是一个亟待解决的问题。在软件缺陷预测过程中, 由于软件失效的概率比正常的概率低, 实际得到的数据集基本上都是不均衡的, 即软件失效样本占样本总数较少。而这少数类的识别率往往更重要。传统上, 在软件缺陷预测中, 人们会通过抽样来均衡样本数据以提高失效样本在训练学习过程中的比重, 从而解决数据集不均衡的问题以及影响。Shuo Wang 等人^[10]使用不同的方法来解决数据集不均衡的问题, 包括重新抽样、阈值偏移以及集成方法等。Jing 等人^[11]则采用稀疏表达方法处理并消除由于数据集不均衡而带来的影响。Chris Se-

到稿日期: 2015-05-20 返修日期: 2015-09-16 本文受 2012“核高基”科技专项: 基于国产 CPUOS 的办公信息系统应用方案评测及规范研究(2012ZX01045-006-003)资助。

熊婧(1985—), 女, 硕士生, 工程师, 主要研究方向为软件可靠性及安全性测评技术、软件可靠性工程技术与质量保证技术、基础软件测评技术, E-mail: winning615@126.com; 高岩(1986—), 男, 博士生, 工程师, 主要研究方向为数据挖掘、机器学习、模式分类; 王雅瑜(1987—), 女, 硕士生, 助理工程师, 主要研究方向为软件工程、软件测试。

iffert 等人^[12]对不均衡数据下的分类方法和抽样方法做了细致的研究,并认为若在不均衡情况很严重时,先做抽样后做分类会大大改善分类结果;若不均衡情况并不是特别严重,则做抽样会使分类结果变差。Mingxia Liu 等人^[13]首次提出使用二级代价敏感方法来解决软件缺陷预测中的不均衡问题。对于传统软件缺陷预测模型中应用最广泛的神经网络方法,其同样倾向于对整体有较高的识别率,对少数类的识别率却很低。少数类的识别率对整体识别率的影响较多数类的小,在对比测试过程中往往被忽略;且神经网络具有陷入局部极小的缺点。虽然研究人员对神经网络进行了深入的研究和改进,在 BP 神经网络的基础上提出了改进的算法^[14-16],但样本不均衡导致的识别率低问题仍然存在。因此,如何建立有效的软件缺陷预测模型成为当前该领域工作的重中之重。

本文提出了一种新的解决思路:利用基于 Adaboost 和神经网络的算法来建立软件缺陷预测模型。Adaboost 算法本质上是通过组合弱分类器得到强分类器,同时具有分类错误率上界随着训练增加而稳定下降、不会过拟合等性质,特别是在处理不均衡样本上,对少数类样本有较好的泛化能力,应该说是一种很适合于在各种分类场景下应用的算法。如今,Adaboost 算法主要应用于模式识别领域,如目前比较热门的人脸识别^[17,18]。Adaboost 本质上是处理分类问题,根据软件缺陷预测中的样本不均衡分布特点,Adaboost 完全可以用于软件缺陷预测模型。同时,采用发展成熟的神经网络作为 Adaboost 的弱分类器,将两者的优势相结合,这些特性都使得 Adaboost 适用于软件缺陷预测建模中。

2 软件缺陷预测

2.1 软件缺陷预测架构

软件缺陷预测模型的构建分为 4 个部分,包括:质量数据采集、质量数据预处理、预测模型构建和训练、预测模型评估测试。软件缺陷预测模型的主要框架如图 1 所示。

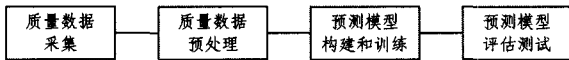


图 1 软件缺陷预测模型

首先采集质量数据,即是将一个软件对象的源代码转化为可以统计分析处理的样本集。通过对软件模块进行度量数据的测量得到原始数据集。然后对所采集的原始质量数据集进行预处理,包括软件属性选择、归一化、消零、主成分分析(PCA)等,以去除冗余属性,提高软件数据的质量。接下来在框架的核心部分对输入的训练样本进行建模和训练,具体过程依所选用的不同分类器的建模算法而有所不同。最后对每个软件模块是否存在缺陷进行预测。

2.2 软件度量数据

在软件缺陷预测模型中,软件模块的度量数据通常作为模型的主要输入。软件度量是指对软件属性的定量研究,主要分为面向结构和面向对象的软件度量。面向结构的软件度量是传统的度量方法,主要针对函数的内部结构复杂度和函数间的联系,如模块长度、内聚性、模块耦合、数据耦合等。面向对象的软件度量针对的是面向对象软件开发,强调的是对等对象实体间的关系,如继承、抽象、封装和多态等,传统的度量对面向对象的特征是无法适用的,也不足以刻画和评估面向对象系统的质量。

2.3 存在问题

软件缺陷预测中的数据分布是不均衡的,软件失效数据样本通常远远少于软件有效数据样本。但是,在软件缺陷数据库中,作为少数类的软件失效数据才是软件缺陷预测的重点,提高少数类的分类性能才是目前软件缺陷预测算法亟待解决的问题。

主流的软件缺陷预测算法主要包括基于统计学方法的传统软件缺陷预测模型和基于神经网络的软件缺陷预测模型。这些传统的软件缺陷预测模型没有全面地考虑影响软件质量的众多不确定性的因素,大多都是只针对一两个影响软件质量的因素进行研究和建模的,这样不能较好地描述软件内部特性和软件所表现出来的缺陷之间的不确定性的关系,得出的软件缺陷预测模型不能较好地表示软件质量的真实情况。总体而言,对于目前大部分常用的软件缺陷预测算法,由于其性能的评价准则、归纳偏置等内在问题,使得其以总体最大分类精度作为目标,在两类训练样本数据相关悬殊时,分类明显偏向多数类训练样本一方,小样本类的预测精度会被忽略,从而产生偏向性。因此,在软件缺陷预测领域有必要寻求一种新的分类算法,使其能在数据分布不均衡的条件下对少数类、多数类进行准确分类。

3 基于 Adaboost 的神经网络算法

3.1 人工神经网络算法

人工神经网络是受生物神经网络功能的运作而产生的,它模拟并学习人脑的结构特点以及对信息的处理机制的一种数学模型。神经网络是由大量的神经元相互联接而成的网络。两个神经元之间的联接由连接前一个神经元的加权值到后一个神经元构成,类似于生物神经网络中的记忆功能。

BP 神经网络(back Propagation Neural Network)是神经网络中的一种重要模式,是应用最广泛的一种神经网络算法,也是软件缺陷预测的常用方法之一。BP 神经网络的学习过程包括信号的正向传播和误差的反向传播,整个网络由输入层、隐藏层、输出层组成,其结构如图 2、图 3 所示。

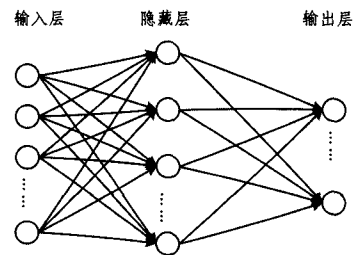


图 2 BP 神经网络结构图

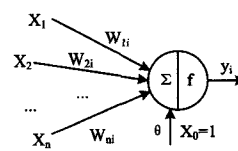


图 3 神经元结构模型

每一层神经元的输出通过联接权值 w 加权后传送到下一层。每个神经元的激励程度由输入 x 、激励函数 $f(\cdot)$ 和阈值决定。

$$net = x_1 w_1 + x_2 w_2 + \dots + x_n w_n \quad (1)$$

$$y = f(net) \quad (2)$$

输出层的实际输出与期望输出的误差平方和产生的误差信号反向传播到输出层、隐藏层和输入层单元,不断调整网络的权值和阈值,使网络的期望输出更加接近实际输出的值,直至两者输出误差的平方和达到最小。

3.2 Adaboost 算法

Adaboost 是一种基于级联分类模型的分类器,全称为 Adaptive Boosting^[19],其级联分类模型如图 4 所示。

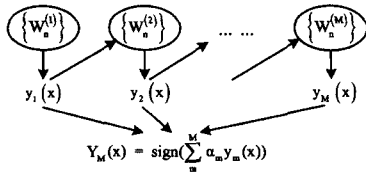


图 4 Adaboost 级联分类模型

Adaboost 算法对同一个训练集训练不同的弱分类器,把多个不同的弱分类器集合起来构成一个最终的强分类器^[20]。使用 BP 神经网络作为 Adaboost 弱分类器,BP 神经网络有多个参数,主要通过选择合适的迭代次数和隐藏层神经元个数来训练弱分类器,防止出现过拟合。

Adaboost 算法本身是通过调整每个样本对应的权重实现的,初始时每个样本对应的权重是相同的,在此样本分布下训练得到第一个基本弱分类器,下一层分类器都根据前一层分类器训练集中的每个样本的分类是否正确以及上次的总体分类的准确率,来更新样本的权值分布,增大弱分类器分类效果较差的样本权重,减小弱分类器分类效果较好的样本权重,将修改权值的新样本分布送给下层分类器进行训练。同时根据错分的情况对每次训练得到的分类器按一定的权重叠加,通过加权投票集成实现最后的决策分类器。

根据 Adaboost 算法的思想,其实现步骤如下。

假设 X 表示样本空间, Y 为样本类别标记集合,则样本训练集为 $S = \{(x_i, y_i) | i = 1, 2, \dots, m\}$, $x_i \in X, y_i \in Y$ 。

Step1 初始化 m 个训练样本的权值,初始样本权值分布为均匀分布; $D_1(i) = 1/m, D_t(i)$ 表示第 t 次迭代中赋给样本的权值;

Step2 T 表示迭代次数,对于第 $t(t = 1, \dots, T)$ 次迭代:

1) 在给定训练样本的概率分布 D_t 下,训练弱分类器 h_t ;

2) 计算弱分类器 h_t 的加权错误率 $\epsilon_t = \text{Pr}_{r \sim D_t} [h_t(x_i) \neq y_i]$;

3) 令 $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$;

4) 更新每个样本的权值:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t}, & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

其中, Z_t 是归一化因子,使得 $\sum_i D_{t+1}(i) = 1$ 。

Step3 最终的强分类器预测输出结果为: $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$ 。

4 实验

4.1 实验准备

为了验证 Adaboost 算法在软件缺陷预测中的有效性,实

验数据选自多组美国宇航局 NASA^[21] 的软件度量数据库。该数据库的数据是从实际的软件系统项目中收集而来。数据集每个样本对应于每个软件模块的度量数据表征面向对象软件的内部属性,以这些内部属性作为缺陷预测模型的输入变量,以软件模块是否存在缺陷作为软件缺陷预测模型的输出。目前,许多算法在使用该数据库时,为了使算法达到良好的效果,会对软件度量数据进行人工的或者自动的属性挑选,以达到最优属性子集的目的。为了全面地评价测试 Adaboost 性能,本文保留完整的软件内部属性。

文中集成方法的弱分类器全部采用了 BP 神经网络分类器。为构造弱分类的 BP 神经网络,将 BP 神经网络的隐藏层数设置为最简单的单层,隐藏层神经元个数设置为输入数据库维度的 1/5,迭代次数为 5,保证每个弱分类器对训练集的正确率达到 50% 或者以上。所采用的对比算法为 BP 神经网络,其采用三层网络,隐藏层神经元个数 m 与输入层维数 n 之间的关系设置为理论上效果较好的。

4.2 评价标准

软件缺陷预测模型的评价指标主要包括准确率 (Accuracy)、查全率 (Recall)、特异性 (Specificity) 和 G-Mean, 这些指标都是基于二分类数据集的混合矩阵,如表 1 所列。

表 1 混合矩阵

	预测正样本数	预测负样本数
实际正样本数	TP	FN
实际负样本数	FP	TN

准确率反映的是预测正确的样本占总样本的百分比,查准率计算的是正确预测为正类的样本数 (True Positive, TP) 与所有真正样本数的比率。G-Mean 是查全率 (Recall) 和特异性 (Specificity) 之间的一种均衡度量,只有当查全率和特异性的值都大时,G-Mean 才会大。在评价软件缺陷预测模型时,需要综合考虑这些评价指标。具体定义如下:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Specificity = \frac{TN}{FP + TN} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$G\text{-Mean} = \sqrt{Recall \times Specificity} \quad (4)$$

4.3 实验结果及分析

本文选用具有代表性的 BP 神经网络作为对比学习方法。BP 神经网络算法是直接利用 matlab 的神经网络工具箱来创建、训练和预测的,隐藏层神经元的激励函数使用 S 型对数函数 logsig。在 matlab 上仿真实现 Adaboost 分类函数,其中 Adaboost 的弱分类器同样使用 matlab 的神经网络工具箱创建、训练和预测的 BP 神经网络,隐藏层神经元的激励函数也是使用 S 型对数函数 logsig。另外,在本文对比实验中的 Bagging 和 Random forest 方法在 matlab 上仿真实现,弱分类器同样采用了 matlab 的神经网络工具中的 BP 神经网络,其中弱分类器个数和神经网络参数与 Adaboost 设置一致,以保证实验的可比性。

对数据集采用随机选择的方法,将每个数据集随机分为 5 份,取其中 4 份作为训练集,1 份作为测试集,对每个数据集采用 5 次实验取平均值的方法得到最终结果,以防止随机数据的影响。在训练之前,对训练数据和测试数据进行归一化预

处理。将处理后的数据输入到分类器中进行训练和分类。

首先,对比了基于 Adaboost 算法的神经网络与传统神经网络的预测性能,如表 2 所列。从表 2 中可以看出,通过在多个数据集的实验对比分析,基于 Adaboost 的软件缺陷预测模型不断提高弱分类器错误分类样本的预测性能,由于弱分类器错误分类样本主要是有缺陷样本,因此可以看到其在查准率方面比传统模型有明显提升,而对于无缺陷样本的预测性能略微受到影响,查全率、准确率等评价指标多有下降。尽管如此,综合评价指标 G-Mean 仍然要比基于 BP 神经网络的软件缺陷预测模型实验结果更好,有效地解决了 BP 神经网络

在小样本下容易过拟合和的问题。

其次,对 Adaboost、Bagging 和 Random Forest 等多种集成方法进行了实验对比,如表 3 所列。从表 3 中可以看出,各个集成方法都在一定程度上提升了对传统预测模型的预测性能,Adaboost 算法预测准确率略高,但相差不大,这也说明集成方法可以有效解决软件缺陷预测中样本分布不均衡的问题。另外,值得注意的是,对于 MW1、PC1 和 PC4 3 个数据集,集成方法的性能都低于传统预测模型,这一结果也验证了之前有研究提出的不同的预测方法对于特定的软件缺陷模型的建立存在统计学上的显著差异的论述。

表 2 基于 Adaboost 算法的软件缺陷预测模型与传统模型对比实验结果

	BP 方法				Adaboost+BP			
	Accuracy	Recall	Specificity	G-Mean	Accuracy	Recall	Specificity	G-Mean
CM1	0.827±0.018	0.894±0.029	0.375±0.021	0.579±0.022	0.782±0.018	0.836±0.014	0.416±0.024	0.590±0.043
KC1	0.720±0.022	0.799±0.026	0.414±0.022	0.575±0.022	0.738±0.022	0.824±0.019	0.403±0.028	0.577±0.029
KC3	0.718±0.019	0.849±0.021	0.142±0.016	0.343±0.017	0.700±0.019	0.828±0.015	0.142±0.025	0.348±0.025
MC1	0.845±0.064	0.900±0.120	0.444±0.063	0.632±0.067	0.778±0.064	0.813±0.070	0.518±0.090	0.649±0.085
MC2	0.496±0.058	0.562±0.102	0.375±0.062	0.459±0.062	0.608±0.058	0.666±0.054	0.500±0.058	0.577±0.071
MW1	0.767±0.033	0.800±0.095	0.533±0.031	0.653±0.035	0.773±0.033	0.814±0.032	0.466±0.046	0.616±0.047
PC1	0.886±0.036	0.921±0.105	0.417±0.047	0.619±0.044	0.863±0.036	0.901±0.026	0.361±0.045	0.570±0.040
PC2	0.844±0.039	0.944±0.131	0.111±0.039	0.324±0.038	0.833±0.039	0.917±0.036	0.222±0.055	0.451±0.055
PC3	0.810±0.090	0.845±0.194	0.500±0.063	0.650±0.065	0.807±0.090	0.835±0.066	0.564±0.109	0.686±0.138
PC4	0.864±0.037	0.918±0.067	0.523±0.055	0.693±0.053	0.867±0.037	0.932±0.038	0.457±0.056	0.652±0.070
PC5	0.754±0.054	0.934±0.122	0.280±0.051	0.511±0.060	0.753±0.054	0.912±0.044	0.333±0.075	0.551±0.085

表 3 多种集成方法的对比实验结果

	Adaboost		Bagging		Random Forest	
	Accuracy	G-Mean	Accuracy	G-Mean	Accuracy	G-Mean
CM1	0.782±0.018	0.590±0.043	0.815±0.021	0.580±0.018	0.821±0.035	0.588±0.030
KC1	0.738±0.022	0.577±0.029	0.725±0.022	0.575±0.021	0.732±0.025	0.577±0.025
KC3	0.700±0.019	0.348±0.025	0.737±0.016	0.454±0.017	0.705±0.025	0.451±0.023
MC1	0.778±0.064	0.649±0.085	0.816±0.063	0.633±0.067	0.798±0.087	0.645±0.067
MC2	0.608±0.058	0.577±0.071	0.581±0.062	0.474±0.066	0.611±0.073	0.503±0.098
MW1	0.773±0.033	0.616±0.047	0.767±0.031	0.613±0.033	0.755±0.051	0.602±0.037
PC1	0.863±0.036	0.570±0.040	0.852±0.047	0.565±0.039	0.868±0.045	0.576±0.045
PC2	0.833±0.039	0.451±0.055	0.848±0.039	0.433±0.040	0.827±0.051	0.407±0.048
PC3	0.807±0.090	0.686±0.138	0.811±0.063	0.650±0.072	0.805±0.098	0.674±0.113
PC4	0.867±0.037	0.652±0.070	0.869±0.055	0.667±0.057	0.857±0.057	0.670±0.072
PC5	0.753±0.054	0.551±0.085	0.753±0.051	0.543±0.057	0.749±0.063	0.539±0.083

结束语 本文提出了一种利用 Adaboost 级联分类器对软件缺陷预测进行建模的新方法,同时将 Adaboost 和神经网络相结合进行实验验证。从实验结果分析可以看出,在软件缺陷预测的通用架构下,Adaboost 方法能够有效提升软件缺陷预测的性能,尤其是样本不均衡的情况下,预测性能明显优于传统的 BP 神经网络,并且具有比 BP 神经网络更强的泛化能力。本文创新性地将 Adaboost 算法应用到软件缺陷预测中,未来的研究工作将对基于 Adaboost 算法的软件缺陷预测模型进行更深入的分析 and 全面的评估,并研究 Adaboost 算法与更多方法的结合,如决策树、SVM、其他神经网络等。

参考文献

[1] Wang Qing, Wu Shu-jian, Li Ming-shu. Software Defect Prediction[J]. Journal of Software, 2008, 19(7): 1565-1580 (in Chinese)
王青,伍书剑,李明树. 软件缺陷预测技术[J]. 软件学报, 2008, 19(7): 1565-1580

[2] Khoshgoftaar T M, Seliya N. Tree-based software quality estimation models for fault prediction[C]// IEEE Symposium on Metrics. 2002; 203-214

[3] Khoshgoftaar T M, Allen E B, Jones W D, et al. Accuracy of software quality models over multiple releases[J]. Annals of Software Engineering, 2000, 9(1/2): 103-116

[4] Khoshgoftaar T M, Liu Y, Seliya N. A multi-objective module-order model for software quality enhancement[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(6): 593-608

[5] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 2010, 232(6088): 533-536

[6] Gupta N, Singh M P. Estimation of software reliability with execution time model using the pattern mapping technique of artificial neural network[J]. Computers & Operations Research, 2005, 32(1): 187-199

[7] Khoshgoftaar T M, Pandya A S. A neural-network approach for predicting software development faults[J]. IEEE Computer, 1992, 17(8): 83-88

[8] Lessmann S, Baesens B, Mues C, et al. Benchmarking classification models for software defect prediction: A proposed framework and novel findings[J]. IEEE Transactions on Software Engineering, 2008, 34(4): 485-496

[9] Ghotra B, McIntosh S, Hassan A E. Revisiting the impact of

classification techniques on the performance of defect prediction models[C]// Proceedings of the International Conference on Software Engineering. 2015

- [10] Wang S, Yao X. Using class imbalance learning for software defect prediction[J]. IEEE Transactions on Reliability, 2013, 62(2): 434-443
- [11] Jing X Y, Ying S, Zhang Z W, et al. Dictionary learning based software defect prediction[C]// Proceedings of the International Conference on Software Engineering. 2014: 414-423
- [12] Seiffert C, Khoshgoftaar T M, Van Hulse J, et al. An empirical study of the classification performance of learners on imbalanced and noisy software quality data[J]. Information Sciences, 2014, 259: 571-595
- [13] Liu M X, Miao L S, Zhang D Q. Two-stage cost-sensitive learning for software defect prediction[J]. IEEE Transactions on Reliability, 2014, 63(2): 676-686
- [14] He Yong, Xiang Li-guo. Study and Application of BP Neural Network Model Based on Fuzzy Clustering[J]. Systems Engineering-theory & Practice, 2004, 24(2): 79-82(in Chinese)
何勇, 项利国. 基于模糊聚类的 BP 神经网络模型研究及应用[J]. 系统工程理论与实践, 2004, 24(2): 79-82
- [15] Thwin M M T, Quah T S. Application of neural networks for

software quality prediction using object-oriented metrics[J]. Journal of Systems and Software, 2005, 76(2): 147-156

- [16] Liu Tian-shu. The Research and Application on BP Neural Network Improvement[D] Harbin: Northeast Forestry University, 2011(in Chinese)
刘天舒. BP 神经网络的改进研究及应用[D]. 哈尔滨: 东北林业大学, 2011
- [17] Ma S, Du T. Improved Adaboost face detection[C]// International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). 2010: 434-437
- [18] Zhang H, Xie Y, Xu C. A classifier training method for face detection based on Adaboost[C]// International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE). 2011: 731-734
- [19] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[C]// Computational learning theory. Berlin Heidelberg, 1995: 23-37
- [20] Zhang Z, Xie X. Research on Adaboost M1 with random forest [C]// International Conference on Computer Engineering and Technology (ICCET). 2010: 647-652
- [21] NASA. The NASA Metrics Data Program[OL]. <http://mdp.ivv.nasa.gov>

(上接第 179 页)

的移动网络环境下,通过跟踪、学习用户的兴趣、偏好以及性格特征等信息,实时、准确地发现不同用户对各种移动网络服务的需求,并对其变化做出适应和调整。对于移动业务流程的识别与分析,随着互联网业务的移动化,业务识别越来越受到重视。文献[9]中李磊等人提出了一种业务识别技术,即通过深度检查分析业务流量从数据链路层到应用层的报文,根据协议类型、端口号、特征字符串和流量行为特征等参数来确定业务类型、业务状态、业务内容和用户行为等信息,并进行分类统计和存储。同时,在文献[10]中王楠等人提出了一种基于目标的业务流程概念模型框架,通过目标驱动的方法,并以目标-行为的细化分层关系为基础,自底向上生成与各个子目标相匹配的流程片段。不同于上述研究工作,本文采取需求驱动的方式来分析识别需要移动化的业务流程,即首先分析在移动环境下用户需求的变化,然后根据用户需求的变化来改进现有的业务流程。

结束语 本文提出了一种目标驱动的方法来帮助分析人员识别和改进企业的业务流程以适应移动互联网的发展所带来的环境变化。该方法用目标来表达用户的需求,通过建立企业的业务流程模型与用户需求模型以及二者之间的对应关系,然后分析和识别移动环境下用户需求的变化,来改进企业的用户需求模型以及业务流程模型。

虽然本文以一个医院为案例来说明所提方法的可行性,但是还需要更多的案例研究来验证其有效性。特别是在对企业相关传统业务以及涉及到安全业务的移动化分析与改进上,还需要进一步的研究。

参 考 文 献

- [1] Jureta I, Borgida A, Ernst N, et al. The requirements problem for adaptive systems[J]. Transactions on Management Information Systems(TMIS), 2014, 5(3): 17-50

- [2] Aguilar-Saven R. Business process modeling: review and framework[J]. International Journal of Production Economics, 2004, 90(2): 129-149
- [3] White S. Process modeling notations and workflow patterns[J]. Workflow Handbook, 2004, 1(1): 265-294
- [4] Simpson R. An xml representation for crew procedures[J]. Department of Rehabilitation Science and Technology, 2005, 1(2): 18-26
- [5] Rao N M, Seetharam P. An Intelligent Software Workflow Process Design for Location Management on Mobile Devices[J]. International Journal of Advanced Computer Science & Applications, 2012, 1(1): 43-49
- [6] Reijers H, Mansar S. Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics[J]. The International of Management Science, 2005, 33(4): 283-306
- [7] Kevin B. The reengineering handbook: a step-by-step guide to business transformation [J]. Journal for Healthcare Quality, 1995, 24(1): 86-87
- [8] Meng Xiang-bin, Wang Fan, Shi Yan-cui, et al. Being mobile users demand technology and its application [J]. Journal of Software, 2014, 25(3): 439-456(in Chinese)
孟祥斌, 王凡, 史艳翠, 等. 移动用户需求获取技术及应用[J]. 软件学报, 2014, 25(3): 439-456
- [9] Li Lei. The research of Mobile Internet business identification method [J]. Mobile Communications, 2012, 36(S1): 128-132(in Chinese)
李磊. 移动互联网业务识别方法研究[J]. 移动通信, 2012, 36(S1): 128-132
- [10] Wang Nan, Sun Shan-wu, Ouyang Dan-tong. Business process modeling based on the concept of target [J]. Computer Science, 2014, 41(10): 220-224(in Chinese)
王楠, 孙善武, 欧阳丹彤. 基于目标的业务流程概念建模[J]. 计算机科学, 2014, 41(10): 220-224