

# 基于代价敏感学习的软件缺陷预测方法

陆海洋<sup>1</sup>, 荆晓远<sup>2</sup>, 董西伟<sup>1</sup>, 刘 茜<sup>1</sup>  
(1. 南京邮电大学 计算机学院, 江苏 南京 210023;  
2. 南京邮电大学 自动化学院, 江苏 南京 210023)

**摘 要:** 软件缺陷预测是改善软件开发质量、提高测试效率的重要途径。文中分析了软件缺陷预测的特点, 同时针对当前软件缺陷预测中存在特征冗余问题和类不平衡问题进行了深入研究。首先为了解决软件模块中的特征冗余问题给软件缺陷预测造成困难, 提高对软件缺陷预测的准确率, 采用基于代价敏感的拉普拉斯特征映射方法 (CSLE) 对原样本空间进行降维, 改进拉普拉斯算法 (LE) 中的距离度量方式, 提高降维映射精度; 然后通过基于代价敏感的神经网络的方法 (CSB-PNN) 对软件模块进行分类, 调整 BP 神经网络的权值和偏置参数, 使 BP 神经网络对有缺陷软件模块的误分更加敏感, 进一步提高分类效果。在 NASA 软件缺陷标准数据集上与最新的几种软件缺陷预测方法相比, 文中提出的方法能够有效提高有缺陷样本的召回率和  $F$ -measure 值。

**关键词:** 软件缺陷预测; 代价敏感; 拉普拉斯特征映射; 神经网络

**中图分类号:** TP301

**文献标识码:** A

**文章编号:** 1673-629X(2015)11-0058-03

doi:10.3969/j.issn.1673-629X.2015.11.012

## Software Defect Prediction Based on Cost-sensitive Learning

LU Hai-yang<sup>1</sup>, JING Xiao-yuan<sup>2</sup>, DONG Xi-wei<sup>1</sup>, LIU Qian<sup>1</sup>  
(1. College of Computer, Nanjing University of Posts and Telecommunications,  
Nanjing 210023, China;  
2. College of Automation, Nanjing University of Posts and Telecommunications,  
Nanjing 210023, China)

**Abstract:** Software defect prediction is an important way to improve the quality of software development and raise the testing efficiency. In this paper, analyze the characteristics of software defect prediction and focus on the research of redundancy features and the imbalance class problem existed in current software defect. In order to solve the difficulty of software defect prediction caused by redundancy features in software modules, improving the accuracy for software defect prediction, adopt a new method named Cost-Sensitive Laplacian Eigenmaps (CSLE) to reduce the dimensionality of original sample space, improving the distance measurement method of Laplacian Eigenmaps (LE) to enhance the dimension reduction mapping accuracy. In addition, propose a new method named Cost Sensitive Back Propagation Neural Network (CSBPNN) to classify the software module, adjusting the weights and bias parameters of BP neural network, which makes the error of BP neural network to flawed software modules points more sensitive, further improving the classification effect. Compared with the latest several software defect prediction methods on NASA software datasets, prove that this method can improve the recall rate and  $F$ -measure value in software defect prediction.

**Key words:** software defect prediction; cost-sensitive; Laplacian Eigenmaps; neural network

## 0 引言

随着计算机软件技术的快速发展, 软件缺陷预测已经成为软件工程中一个非常重要的研究课题<sup>[1-2]</sup>。软件缺陷预测是指从已有软件模块中获得的历史数

据, 对新的软件模块进行缺陷预测, 从而判定它们是否存在缺陷, 为软件项目提供决策支持<sup>[3-4]</sup>。许多经典的机器学习方法, 如支持向量机<sup>[5]</sup>、决策树<sup>[6]</sup>、朴素贝叶斯<sup>[7]</sup>、集成学习<sup>[8-9]</sup>等, 都针对这一问题来建立预测

收稿日期: 2015-02-03

修回日期: 2015-05-05

网络出版时间: 2015-11-04

基金项目: 国家自然科学基金资助项目(61272273); 江苏省 333 工程项目(BRA2011175); 南京邮电大学校科研项目(XJKY14016)

作者简介: 陆海洋(1990-), 男, 研究生, 研究方向为机器学习; 荆晓远, 教授, 博士生导师, 研究方向为模式识别、图像与信号处理、信息安全、机器学习与数据挖掘。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0948.014.html>

模型,对新的软件模块进行分类,取得了不错的预测效果。

在 NASA 提供的软件缺陷数据库中,软件特征由 20 至 49 不等。Wang<sup>[10]</sup>等通过实验表明,只需要三个软件特征就能达到缺陷预测效果。因为在每一类软件特征中,除了基本特征是从源代码中直接抽取,其他的特征都是由这些基本特征计算得到。因此,同一类软件特征之间的相关性较大,存在较多的冗余特征。王培等<sup>[11]</sup>通过特征选择算法选出不同大小的最优属性子集用于软件预测,提高了软件预测模型的准确度。

在软件缺陷预测中,有两类错误分类<sup>[12]</sup>:第一,将有缺陷样本误分为无缺陷样本;第二,将无缺陷样本误分为有缺陷样本。在软件工程实践中,第一类误分代价要远远大于第二类误分代价。缪林松<sup>[13]</sup>提出一种基于阈值移动技术的代价敏感的神经网络算法。该算法在训练阶段不做代价敏感处理,将判断边界向较低的一类样本边界偏移,从而降低第一类误分风险。

文中在现有研究工作的基础上,采用拉普拉斯特征映射方法<sup>[14]</sup>(LE)对原始样本空间进行非线性降维,去除特征之间的冗余信息;进一步,为了避免将不同类的样本映射到较小的低维邻域中,尤其是将有缺陷样本映射到无缺陷样本邻域中,在 LE 算法计算样本点距离时引入代价敏感信息,以此来提高 LE 的映射精度。另外,文中选用 BP 神经网络作为分类器,并且在 BP 神经网络训练过程中再次引入代价敏感信息,然后对软件模块进行分类预测。不仅提高了算法的分类精度和有缺陷样本的召回率,而且与阈值移动技术相比,提高了分类器的泛化能力。在 NASA 数据库上的实验结果表明,文中提出的代价敏感学习算法(CSLEBP),可以很好地解决软件缺陷预测中的误分代价和类不平衡问题。

## 1 基于代价敏感学习的软件缺陷预测方法

### 1.1 基于代价敏感的拉普拉斯特征映射方法

LE 算法认为高维空间中距离很近的点投影到低维空间也应该离得很近,它的收敛性和鲁棒性较好,因而被广泛应用。

LE 算法具体步骤如下:

第一步:计算每个样本点  $x_i$  与其他样本点之间的欧氏距离,得到由  $x_i$  的最近的  $k$  个邻近点构成的邻域  $S$ 。

第二步:定义权重矩阵  $W$ :

$$W = \begin{cases} \exp(-\|x_i - x_j\|^2/t), & x_j \in S \\ 0, & x_j \notin S \end{cases} \quad (1)$$

其中,参数  $t$  为热核宽度。

第三步:求解  $Lf = \lambda Df$  广义特征值,获得低维嵌入

坐标。其中,  $D$  为对角矩阵,且  $D_{ii} = \sum_j W_{ji}$ ;  $L = D - W$  为邻域上的拉普拉斯算子。最终获得样本  $x_i$  的  $d$  维映射坐标,即  $L$  的第 2 个到第  $d+1$  个特征值对应的特征向量,记为  $\{x_i^1, x_i^2, \dots, x_i^d\}$ 。

然而,LE 算法是一种无监督学习方法,未能充分利用样本的类别信息以及代价信息,在降维的过程中很可能会把相距较远的点映射嵌入到一个较小的邻域内。因此,文中在 LE 算法能够保留高维数据的近邻结构的基础上,引入代价敏感信息,提出了一种基于代价敏感的拉普拉斯特征映射方法(CSLE),以提高降维映射精度。

在 CSLE 中,距离度量定义如下:

$$D(x_i, x_j) = c_{ab} d(x_i, x_j) \quad (2)$$

其中,  $c_{ab}$  是代价系数,表示将  $b$  类样本映射到  $a$  类样本邻域的代价;  $d(x_i, x_j)$  是样本  $x_i$  和  $x_j$  之间的欧氏距离。

### 1.2 基于代价敏感的神经网络算法

BP 神经网络是目前应用最广泛、最有影响的人工神经网络算法之一。理论证明,包含一个隐含层的 BP 网络可以实现任意非线性映射。

为了提高 BP 神经网络算法的泛化能力,文中使用的基于代价敏感的 BP 神经网络算法(CSBPNN)是在训练网络时,在误差函数中加入代价敏感信息,改变 BP 神经网络的误差参数,使 BP 神经网络代价敏感化。在 CSBP 中,误差函数定义如下:

$$E = \frac{1}{2} \sum_{i=1}^d \sum_{k=1}^n (c_{ab}(y_k - o_k))^2 \quad (3)$$

其中,  $d$  是经过 CSLE 降维后的特征维数;  $n$  是网络的神经元个数;  $y_k$  和  $o_k$  分别是第  $k$  个神经元的期望输出和实际输出;  $c_{ab}$  是代价敏感因子,表示将  $b$  样本判断为  $a$  类样本的惩罚代价。

CSBPNN 的输出层和隐含层的误分代价反向传播公式分别为:

$$\delta_j = (y_j - o_j) \cdot o_j \cdot (1 - o_j) \cdot c_{ab}^2 \quad (4)$$

$$\delta_j = o_j(1 - o_j) \cdot \sum_k \delta_k w_{kj} \quad (5)$$

### 1.3 基于代价敏感学习软件缺陷的算法步骤

CSLEBP 算法的实现过程总结如下:

输入:原始样本数据,样本维数为  $m$ ,样本个数为  $n$ ;

步骤 1:归一化,特征向量表示为  $\{x_i^1, x_i^2, \dots, x_i^m\}$ ,其中  $i = 1, 2, \dots, n$ ;

步骤 2:根据式(2)计算得到样本  $x_i$  的代价邻域  $S_i$ ;

步骤 3:根据 CSLE 算法得到  $d$  维特征向量  $\{x_i^1, x_i^2, \dots, x_i^d\}$  作为 CSBP 算法的输入;

步骤 4:根据式(4)和(5)训练 CSBP 神经网络。  
输出:分类器  $f$ 。

2 实 验

为了证明文中提出方法的有效性,本节将 CSLEBP 算法和其他对比算法在 NASA 数据库上进行对比验证。

2.1 数据库介绍

实验中用到的数据集是美国宇航局(NASA)项目的 10 个数据集,如表 1 所示,包含每个组成模块的静态代码度量和相应的缺陷标记数据。这些项目通过一个 bug 跟踪系统的记数来记录各模块的缺陷数。

表 1 NASA 数据集

数据集	缺陷样本数	总数	特征维数
CM1	42	344	38
JM1	1 759	9 593	22
KC1	325	2 096	22
KC3	36	200	40
MC2	44	127	40
MW1	27	264	20
PC1	61	759	20
PC3	140	1 125	20
PC4	178	1 399	38
PC5	503	17 001	39

2.2 评价指标

预测模型的评估指标有:召回率、精确度和  $F - \text{measure}$ 。召回率是指被正确预测为有缺陷的模块数与真实有缺陷的模块数的比值,软件缺陷预测模型的目的就是尽可能地找出有缺陷的模块。精确度是指正确预测的模块数与总模块数之比,该比值是所有数据机器学习分类器应用中广泛使用的指标。一个好的预测模型应达到较高的召回率和高精确度。然而,高召回率的获得往往以低精确度为代价,反之亦然。因此就需要将召回率与精确度结合起来评价,这就是  $F - \text{measure}$  指标,其定义为:

$$F - \text{measure} = 2 * \text{召回率} * \text{精确度} / (\text{召回率} + \text{精确度})$$

2.3 结果及分析

文中将基于代价敏感学习的方法(CSLEBP)与支持向量机(SVM)、决策树(C4.5)、朴素贝叶斯(BP)、集成方法(CEL)进行了对比。每种算法做 20 次随机实验得到  $F - \text{measure}$ ,并取其平均值,实验结果如表

2 所示。

表 2 几种算法对比结果

datasets	SVM	C4.5	NB	CEL	CSLEBP
CM1	0.20	0.25	0.32	0.27	0.34
JM1	0.29	0.34	0.33	0.33	0.38
KC1	0.29	0.39	0.38	0.36	0.42
KC3	0.38	0.38	0.38	0.33	0.40
MC2	0.52	0.48	0.45	0.49	0.56
MW1	0.27	0.27	0.31	0.27	0.37
PC1	0.35	0.32	0.28	0.32	0.38
PC3	0.28	0.29	0.29	0.36	0.39
PC4	0.47	0.49	0.36	0.48	0.52
PC5	0.16	0.48	0.33	0.36	0.49

从最终的实验结果中可以看出,与现今效果比较好的几种软件缺陷预测算法相比,文中提出的 CSLEBP 方法得到了最好的  $F - \text{measure}$ 。这一结果展示了该方法能够更好地处理软件缺陷预测问题。

3 结束语

文中针对软件缺陷预测代价敏感问题和类不平衡问题,首先在拉普拉斯特征映射中引入代价敏感信息,提高降维后的映射精度。然后在神经网络算法中再次引入代价敏感信息,提高算法分类精度和泛化能力。在 NASA 数据库上的实验结果表明,文中提出的方法可以很好地解决软件缺陷预测中代价敏感问题和类不平衡问题。

参考文献:

[1] Lyu M R. Software reliability engineering;a roadmap[C]//Proc of future of software engineering. Minneapolis,MN:IEEE Computer Society,2007:153-170.

[2] Seliya N,Khoshgoftaar T M,van Hulse J. Predicting faults in high assurance software[C]//Proc of 2010 IEEE 12th international symposium on high-assurance systems engineering. San Jose:IEEE,2010:26-34.

[3] Catal C,Diri B. A systematic review of software fault prediction studies[J]. Expert Systems with Applications,2009,36(4):7346-7354.

[4] 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.

[5] Elish K O,Elish M O. Predicting defect-prone software mod-

(下转第 66 页)

构阵列机上实现了算法的并行化设计。在统一图形渲染管线中,两种并行处理方式很好地提高了数据处理的速度;同样利用像素级图像处理数据量大、数据相关性小等特点,在阵列机上很好地实现了图像算法的并行化,并且获得了较高的加速比,实现了图像处理加速。通过文中的研究表明,PAAG 阵列机在并行处理图形图像算法及其相关应用,提高程序执行效率,降低功耗等方面具有独特的优势。在以后的研究工作中,将针对阵列机的整体设计、通信机制等方面作进一步优化,以追求其在高性能并行计算方面取得更多的突破。

#### 参考文献:

- [1] Tomas A M, Haines E, Hoffman N. Real-time rendering[M]. 2nd ed. 夏文宇, 胡艳祥, 译. 北京: 清华大学出版社, 2002.
  - [2] Macedonia M. The GPU enters computing's mainstream[J]. Computer, 2003, 36(10): 106-108.
  - [3] Rainey E, Gautam S. The OpenVX™ Specification Version 1.0 [S]. USA: Khronos Vision Working Group, 2014.
  - [4] Barnes G H, Brown R M, Kato M, et al. The ILLIAC IV computer[J]. IEEE Transactions on Computers, 1968, C-17(8): 746-757.
  - [5] Lindholm E, Nickolis J, Oberman S, et al. NVIDIA Tesla: a unified graphics and computing architecture[J]. IEEE Micro, 2008, 28(2): 39-55.
  - [6] Li T. PAAG: a polymorphic array architecture for graphics and image processing[C]//Proceedings of the 2012 fifth international symposium on parallel architectures, algorithms and programming. [s. l.]: IEEE Computer Society, 2012: 242-249.
  - [7] Moreno M C C, Auzinger T. General-purpose graphics processing units in service-oriented architectures [C]//Proc of IEEE 6th international conference on service-oriented computing and applications. Koloa, HI: IEEE, 2014: 260-267.
  - [8] Kato S, Lakshmanan K, Rajkumar R, et al. TimeGraph: GPU scheduling for real-time multi-tasking environments [C]//Proceedings of the USENIX annual technical conference. [s. l.]: USENIX, 2011.
  - [9] Angel E. Interactive computer graphics[M]. Beijing: Tsinghua University Press, 2006.
  - [10] Ao Qian, Chen Rongguan, Ning Ning, et al. High-definition image processing algorithm and digital platform design [C]//Proc of IEEE 12th international conference on computer and information technology. Chengdu: IEEE, 2013: 798-800.
  - [11] Dong Fuguo, Li Yiling. Study on fast algorithm of median filtering based on DC and ICS method [C]//Proc of international conference on wireless communications & signal processing. Nanjing: IEEE, 2009: 1-5.
  - [12] Trahanian P E, Venetsanopoulos A N. Color image enhancement through 3-D histogram equalization [C]//Proc of IAPR international conference on pattern recognition. The Hague: IEEE, 1992: 545-548.
  - [13] Chiuchisan I. A new FPGA-based real-time configurable system for medical image processing [C]//Proc of e-health and bioengineering conference. Iasi: IEEE, 2014: 1-4.
  - [14] Prajapati H B, Vij S K. Analytical study of parallel and distributed image processing [C]//Proc of international conference on image information processing. Himachal Pradesh: IEEE, 2011: 1-6.
- 
- (上接第 60 页)
- ules using support vector machines[J]. Journal of Systems and Software, 2008, 81(5): 649-660.
  - [6] Wang J, Shen B, Chen Y. Compressed C4.5 models for software defect prediction [C]//Proc of 2012 12th international conference on quality software. Washington D C: IEEE, 2012: 13-16.
  - [7] Wang T, Li W. Naive Bayes software defect prediction model [C]//Proc of 2010 international conference on computational intelligence and software engineering. [s. l.]: [s. n.], 2010: 1-4.
  - [8] Song Q, Jia Z, Shepperd M, et al. A general software defect-proneness prediction framework [J]. IEEE Transactions on Software Engineering, 2011, 37(3): 356-370.
  - [9] Sun Z, Song Q, Zhu X. Using coding-based ensemble learning to improve software defect prediction [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2012, 42(6): 1806-1817.
  - [10] Wang H, Khoshgoftaar T M, Seliya N. How many software metrics should be selected for defect prediction? [C]//Proc of FLAIRS. Palm Beach: [s. n.], 2011.
  - [11] 王 培, 金 聪, 葛贺贺. 面向软件缺陷预测的互信息属性选择方法[J]. 计算机应用, 2012, 32(6): 1738-1740.
  - [12] Lanubile F, Visaggio G. Evaluating predictive quality models derived from software measures: lessons learned [J]. Journal of Systems and Software, 1997, 38(3): 225-234.
  - [13] 缪林松. 基于代价敏感神经网络算法的软件缺陷预测[J]. 电子科技, 2012, 25(6): 75-78.
  - [14] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation [J]. Neural Computation, 2003, 15(6): 1373-1396.

基于代价敏感学习的软件缺陷预测方法

作者：[陆海洋](#)，[荆晓远](#)，[董西伟](#)，[刘茜](#)，[LU Hai-yang](#)，[JING Xiao-yuan](#)，[DONG Xi-wei](#)，[LIU Qian](#)

作者单位：[陆海洋, 董西伟, 刘茜, LU Hai-yang, DONG Xi-wei, LIU Qian\(南京邮电大学 计算机学院, 江苏 南京, 210023\)](#)，[荆晓远, JING Xiao-yuan\(南京邮电大学 自动化学院, 江苏 南京, 210023\)](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015, 25(11)

引用本文格式：[陆海洋. 荆晓远. 董西伟. 刘茜. LU Hai-yang. JING Xiao-yuan. DONG Xi-wei. LIU Qian 基于代价敏感学习的软件缺陷预测方法](#)[期刊论文]-[计算机技术与发展](#) 2015(11)