类不平衡稀疏重构度量学习软件缺陷预测

史作婷[1],吴　迪[2],荆晓远[2,3],吴　飞[3]
(1. 南京邮电大学 计算机学院,江苏 南京 210003;
2. 武汉大学 计算机学院 软件工程国家重点实验室,湖北 武汉 430072;
3. 南京邮电大学 自动化学院,江苏 南京 210003)

摘　要:软件缺陷预测是提升软件质量的重要手段。为了改善缺陷预测性能,目前许多机器学习领域的最新成果已经引入到软件缺陷预测中。但是,软件缺陷预测数据通常存在类别分布不平衡的问题,这会影响预测效果。针对这个问题,提出了类不平衡稀疏重构距离度量学习软件缺陷预测方法。该方法首先在度量学习中加入代价敏感因素,学习距离度量特征矩阵并解决软件缺陷预测中分类错误代价不同的问题。其次,通过在目标函数中加入权重来进一步提高小类样本距离度量学习的准确性。最后,为了解决预测阶段数据集的类别不平衡问题,采用了改进加权 KNN 算法预测测试样本标签。在 NASA 软件缺陷预测标准数据集上的实验结果证明了该方法能提高召回率与 F -measure 值,改善分类性能。

关键词:软件缺陷预测;类不平衡;改进加权 KNN;度量学习
中图分类号:TP31　　**文献标识码**:A　　**文章编号**:1673-629X(2018)06-0125-04
doi:10.3969/j.issn.1673-629X.2018.06.028

Prediction of Defect of Class-imbalance Sparse Reconstruction Metric Learning Software

SHI Zuo-ting[1], WU Di[2], JING Xiao-yuan[2,3], WU Fei[3]
(1. School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;
2. State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China;
3. School of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Software defect prediction (SDP) is an important method to improve the quality of software. Currently many latest results from machine learning has been applied to improve the performance of defect prediction. However, imbalance of class distribution usually exists in SDP dataset, which might affect the prediction performance. For this, we propose a novel software defect prediction method termed class-imbalance sparse reconstruction metric learning (CSRML). In CSRML, by introducing cost-sensitive factor into metric learning, a feature matrix of distance metric can be learned and the problem of different cost of misclassification can also be solved. And weight parameter is added in objective function to further improve the accuracy of the small class samples distance metric learning. Finally, improved weighted KNN (IWKNN) method is employed to predict the label of test sample for tackling class imbalance in prediction phase. Experiment on the NASA SDP dataset demonstrates that the proposed method can improve the recall rate, F -measure value and classification performance.

Key words: software defect prediction; class-imbalance; IWKNN; metric learning

0　引　言

软件缺陷预测是软件工程领域的一个重要研究方向,对于发现程序错误、保障软件质量有重要的意义。已有的软件缺陷预测技术借助于机器学习等方法,通过分析软件代码,提取与软件缺陷相关的度量元,构建模型,找出项目中潜在的缺陷程序模块[1]。

研究表明,软件系统中只有极少数的模块是有缺陷的,这说明数据集的类别分布不平衡。常用的机器学习算法直接用于不平衡数据集分类时会倾向于把有缺陷样本错分到无缺陷类中。但是,将有缺陷样本错

收稿日期:2017-08-11　　**修回日期**:2017-12-13　　**网络出版时间**:2018-02-24
基金项目:国家自然科学基金(61272273)
作者简介:史作婷(1992-),女,硕士研究生,研究方向为信息安全;荆晓远,教授,博导,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。

误预测与将无缺陷样本错误预测的代价是不同的[2]。为了解决软件缺陷预测中数据不平衡的问题,目前常用的方法是采样法(包括上采样、下采样等)、代价敏感学习方法以及集成学习方法。

现在已经有很多典型的分类方法运用在软件缺陷预测领域中,比如 SVM[3]、朴素贝叶斯[4]、决策树[5]、神经网络[6]等算法。最近,一些机器学习领域最新的研究成果,例如字典学习、稀疏表示[7]等也引入到软件缺陷预测中,并且取得了良好的预测效果。代价敏感鉴别字典学习(cost-sensitive discriminative dictionary learning,CDDL)[8]结合稀疏表示字典学习以及代价敏感因子,在提升预测性能的同时也解决了类不平衡问题。代价敏感局部协同表示(cost-sensitive local collaborative representation,CLCR)[9]利用协同表示为给定的一个测试模块找出训练集中的邻居模块,然后使用这些邻居模块的线性组合表示测试模块。

文献[10]提出了基于局部稀疏重构的度量学习方法(local sparse reconstruction based metric learning,LSRML),首次将距离度量学习方法运用到软件缺陷预测中,在稀疏表示的基础上引入距离度量学习技术。但是该方法并没有针对样本集类别不平衡的问题进行处理,影响了算法性能。因此文中提出一种类不平衡稀疏重构度量学习软件缺陷预测方法(class-imbalance sparse reconstruction metric learning software defect prediction,CSRML),在学习特征矩阵的阶段融入代价敏感因子,同时加入类别权重提高小类样本距离度量学习的准确性,并在分类预测阶段使用改进加权 KNN(improved weighted KNN,IWKNN)算法预测标签。最后在 NASA 数据集上验证该方法的有效性。

1 相关工作

1.1 代价敏感学习方法

软件缺陷领域中样本错误分类的代价是不同的,通常将无缺陷样本预测为有缺陷样本的情况称为Ⅰ类错分,将有缺陷样本预测为无缺陷样本的情况称为Ⅱ类错分,而Ⅱ类错分的代价远大于Ⅰ类错分。代价敏感学习提高Ⅱ类错分的惩罚,可以生成分类模型来使错误分类的代价最小。例如代价敏感多层感知机(CSMLP)[11]在目标函数中加入一个代价参数来区分不同类错误带来的代价。

1.2 稀疏表示

给定训练样本集 $X=[X_1,X_2,\cdots,X_c]\in\Re^{m\times n}$,其中 c 表示类别数,第 i 类训练样本集为 $X_i=[x_{i,1},x_{i,2},\cdots,x_{i,N_i}]$,$N_i$ 表示第 i 类样本的样本数。对于一个测试样本 y,可以看作是训练样本的联合稀疏表示[7],公式表示为:

$$\arg\min_{\boldsymbol{\alpha}}\{\|y-X\boldsymbol{\alpha}\|_2^2+\mu\|\boldsymbol{\alpha}\|_1\} \tag{1}$$

其中,$\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\cdots,\alpha_c]^{\mathrm{T}}$ 表示稀疏系数向量;μ 为使 $\boldsymbol{\alpha}$ 保持稀疏性的正则化系数。

1.3 基于局部稀疏重构的距离度量学习

式 1 中的稀疏重构误差 $\|y-A\boldsymbol{\alpha}\|_2^2$ 使用欧氏距离度量计算,为了更好地体现出样本间的鉴别信息,LSRML 算法使用距离度量学习替换稀疏重构技术中的欧氏距离度量方式。

使用马氏距离代替欧氏距离,重新定义稀疏重构误差,即:$d_M^2(y,X)=(y-X\boldsymbol{\alpha})^{\mathrm{T}}\boldsymbol{M}(y-X\boldsymbol{\alpha})$,因此可以得到新的稀疏表示公式:

$$\arg\min_{\boldsymbol{\alpha}}d_{\boldsymbol{M}}^2(y,X)+\mu\|\boldsymbol{\alpha}\|_1 \tag{2}$$

因此算法的主要工作就是学习一个能很好地反映样本之间关系的矩阵,使得同类样本之间距离更小,异类样本之间距离更大[12]。为了使学习到的距离度量矩阵的鉴别性更强,LSRML 算法构建了类内稀疏重构项 $R(x_i,A)$ 和类间稀疏重构项 $R(x_i,B)$,其中 A 表示与 x_i 同类的训练样本集,B 表示与 x_i 异类的训练样本集。同时为稀疏系数加入局部权重 D_i 与 $\tilde{D}_i$,使同类近邻样本求得的稀疏表示系数更大。

局部权重定义为:

$$D_i=\exp\left(\frac{\|x_i-x_j\|^2}{r_1}\right),x_j\in A \tag{3}$$

$$\tilde{D}_i=\exp\left(\frac{\|x_i-x_k\|^2}{r_2}\right),x_k\in B \tag{4}$$

其中,$r_1>r_2$。

最终得到的稀疏重构项为:

$$R(x_i,A)=(x_i-A\beta)^{\mathrm{T}}\boldsymbol{M}(x_i-A\beta)+\sigma\|D_i\beta\|_1=d_{1,M}^2(x_i,A)+\sigma\|D_i\beta\|_1 \tag{5}$$

$$R(x_i,B)=(x_i-B\gamma)^{\mathrm{T}}\boldsymbol{M}(x_i-B\gamma)+\sigma\|\tilde{D}_i\gamma\|_1=d_{2,M}^2(x_i,B)+\sigma\|\tilde{D}_i\gamma\|_1 \tag{6}$$

其中,β 表示 x_i 关于样本集 A 的稀疏表示系数;γ 表示 x_i 关于样本集 B 的稀疏表示系数。

虽然 LSRML 方法可以学习到鉴别性更好的度量矩阵,但是该方法没有考虑到软件缺陷预测中数据不平衡的问题与代价敏感问题。因此文中在度量矩阵学习阶段引入代价敏感因子来减小样本错分代价,同时更加注重小类样本距离度量学习的准确性,并在测试样本分类阶段使用改进加权 KNN 算法。

2 类不平衡稀疏重构度量学习软件缺陷预测

文中提出的 CSRML 方法分为两个阶段:距离度量矩阵学习阶段与使用 IWKNN 算法分类阶段。第一

个阶段可以学习到距离度量矩阵 $\boldsymbol{M}$。在这一阶段中引入代价敏感因子,并为不同类别的样本赋予不同的权重,提高小类样本计算的准确性,目的是使学习到的矩阵能更适用于类别不平衡的数据集。第二个阶段使用 IWKNN 分类算法进行样本标签预测,使用该算法的目的是在进行分类任务时也考虑样本类别不平衡的问题。

2.1　距离度量矩阵学习

为了解决软件缺陷预测中的数据不平衡问题与样本错分代价不同的问题,在类间稀疏重构项中加入代价敏感因子,得到新的类间稀疏重构项公式:

$$R(x_i,B)=\sum_{j=1}^{c}\mathrm{cost}(i,j)\times d_{2,M}^2(x_i,B)+\sigma\parallel\tilde{D}_i\gamma\parallel_1 \tag{7}$$

其中, $\mathrm{cost}(i,j)$ 表示样本错分的惩罚值,将有缺陷样本预测为无缺陷样本时惩罚值更大。$\mathrm{cost}(i,j)$ 的取值为表 1 中的代价矩阵。

表 1　代价矩阵

	预测为无缺陷	预测为有缺陷
实际有缺陷	0	cost(1,2)
实际无缺陷	cost(2,1)	0

对于软件缺陷预测来说,少数的存在缺陷的样本需要被检测出来。所以在学习过程中除了增加对错分的惩罚之外,也应该提高小类样本距离度量学习的准确性,可以为不同类别的样本赋予不同的权值,使小类样本之间距离更近。权重计算公式为:

$$w_i=\frac{N}{N_i},i=1,2,\cdots,c \tag{8}$$

其中, N 表示样本总数; N_i 表示第 i 类样本个数; c 表示样本类别数,文中 c 取值为 2。

因此可以得到 CSRML 算法的目标函数:

$$\begin{aligned}J_{(M,\beta,\gamma)}=\underset{(M,\beta,\gamma)}{\mathrm{argmin}}\{&\sum_i w_i d_{1,M}^2(x_i,A)+\mu\sum_i\xi_i+\\&\sigma\sum_i(\parallel D_i\beta_i\parallel_1+\parallel\tilde{D}_i\gamma_i\parallel_1)\}\\ \text{s.t. }&\sum_{j=1}^{c}\mathrm{cost}(i,j)\times d_{2,M}^2(x_i,B)-\\&d_{1,M}^2(x_i,A)\geqslant 1-\xi_i,\xi_i\geqslant 0,\boldsymbol{M}\geqslant 0\end{aligned} \tag{9}$$

式 9 可以分成两个子问题来求解:固定 $\boldsymbol{M}$ 更新 β 与 γ;固定 β 与 γ 更新 $\boldsymbol{M}$。目标函数的优化过程就是迭代更新 $\boldsymbol{M}$ 与 β、γ 的过程。

2.2　改进加权 KNN 算法

K 近邻算法是一种经典的分类算法,其基本思想是将测试样本标记为其 K 个近邻样本中类别数最多的那一类[13]。文中在度量矩阵学习阶段没有改变原始样本中数据不平衡分布的情况,因此,在后续的分类任务中还需要考虑不同类别间样本数目差别较大对算法分类性能的影响。

文中引入了 IWKNN,为训练样本赋予不同权重,在分类阶段统计测试样本与近邻样本的相似度之和,以此作为样本分类的判决准则。IWKNN 算法的步骤如下:

(1)为有缺陷与无缺陷的训练样本赋予不同的权重,使得有缺陷样本具有更大的权重,同一类样本权重相同,如式 8 所示。

(2)计算测试样本与所有训练样本的马氏距离,公式为:

$$d(y,x_j)=\sqrt{(y-x_j)^{\mathrm{T}}\boldsymbol{M}(y-x_j)},\ j=1,2,\cdots,N \tag{10}$$

(3)找出式 10 中求出的 K 个距离测试样本最近的训练样本。

(4)分别计算测试样本与 K 个近邻样本的相似度,距离越近,相似度越大,计算公式为:

$$\mathrm{sim}(y,x_j)=\frac{1}{1+d(y,x_j)},j=1,2,\cdots,k \tag{11}$$

(5)计算测试样本与 K 个近邻样本中每类样本的加权相似度之和,计算公式为:

$$\mathrm{sim}(y,X_i)=\sum_{j=1}^{\tilde{N}_i}w_i\times\mathrm{sim}(y,x_j) \tag{12}$$

其中, $i=1,2,\sum_i\tilde{N}_i=k$。

(6)测试样本类别指定为与其加权相似度最大的类,计算公式为:

$$C(y)=\arg\max_i\mathrm{sim}(y,X_i) \tag{13}$$

结合距离度量矩阵学习阶段以及 IWKNN 算法分类阶段,可得到 CSRML 算法的具体步骤:

(1)初始化鉴别矩阵 $\boldsymbol{M}$。初始化矩阵 $\boldsymbol{M}$ 为欧氏度量矩阵 $\boldsymbol{M}=\boldsymbol{I}$。

(2)更新稀疏系数 β 与 γ。固定矩阵 $\boldsymbol{M}$,依次求得 β 与 γ。

(3)更新半正定矩阵 $\boldsymbol{M}$。固定稀疏系数 β 与 γ,更新矩阵 $\boldsymbol{M}$。

(4)分解半正定矩阵。$\boldsymbol{M}=\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$,更新训练样本 $x_i=\boldsymbol{W}^{\mathrm{T}}x_i$。

(5)输出矩阵 $\boldsymbol{M}$。返回步骤 2,直到连续两次迭代求得的目标函数值 $J_{(M,\beta,\gamma)}$ 足够接近或者达到最大的迭代次数。

(6)利用 IWKNN 算法求出测试样本的标签。

3　实　验

3.1　数据库

使用 NASA 数据库[14-15]中的五个项目作为实验

数据,每个项目代表一个 NASA 的软件系统或子系统,由静态代码度量指标和样本缺陷标签组成。表 2 列出了数据集的详细信息。从表中可以看出,这五个数据集中有缺陷样本数所占比例都低于 13%,有缺陷样本与无缺陷样本这两类样本的数目之间差别较大。

表 2 NASA 数据集详细信息

项目	维数	样本总数	缺陷样本数	缺陷样本占/%
CM1	38	344	42	12.21
MW1	20	264	27	10.23
PC1	20	759	61	8.04
PC3	20	1125	140	12.44
PC4	38	1399	178	12.72

3.2 评价指标

采用四个常用的度量指标来评价该算法的实验效果,四个度量指标包括召回率(recall (pd))、假阳率(false positive rate (pf))、F - measure 和曲线下面积(area under curve (AUC))。这些度量指标由表 3 所示的四种预测结果定义。

表 3 四种预测结果

	预测为缺陷样本	预测为无缺陷样本
缺陷样本	A	B
无缺陷样本	C	D

其中,pd 表示被正确分类的缺陷样本占所有缺陷样本的比例,pf 表示被错分的无缺陷样本占所有无缺陷样本的比例。计算公式为:

$$pd = A/(A + B) \tag{14}$$

$$pf = C/(C + D) \tag{15}$$

精准度(precision)用来评价预测模型的准确度,公式为:

$$precision = A/(A + C) \tag{16}$$

F - measure 指标用于平衡 precision 和 recall,计算公式为:

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{17}$$

AUC 表示 ROC 曲线下的面积,ROC 曲线是由一列 pf、pd 数据对得到的一条曲线。

3.3 实验结果

CSRML 算法主要与软件缺陷预测中的几种典型方法作对比,包括 SVM、CC4.5、NB。此外,文中算法是对 LSRML 算法做的改进,因此也要将其作为对比算法。

CSRML 算法与其他方法的对比结果如表 4 所示。

表 4 实验结果

数据集	度量指标	SVM	CC4.5	NB	LSRML	CSDML
CM1	pd	0.15	0.26	0.44	0.66	0.71
	pf	0.04	0.11	0.18	0.17	0.20
	F - measure	0.20	0.25	0.32	0.51	0.55
	AUC	0.45	0.43	0.52	0.72	0.80
MW1	pd	0.21	0.29	0.49	0.67	0.68
	pf	0.15	0.17	0.19	0.15	0.18
	F - measure	0.27	0.27	0.31	0.51	0.60
	AUC	0.42	0.43	0.41	0.68	0.71
PC1	pd	0.66	0.38	0.36	0.70	0.78
	pf	0.10	0.09	0.11	0.22	0.25
	F - measure	0.35	0.32	0.28	0.53	0.60
	AUC	0.59	0.50	0.46	0.76	0.73
PC3	pd	0.64	0.34	0.28	0.78	0.79
	pf	0.19	0.09	0.09	0.27	0.26
	F - measure	0.28	0.29	0.29	0.64	0.68
	AUC	0.53	0.52	0.47	0.72	0.75
PC4	pd	0.72	0.49	0.39	0.75	0.77
	pf	0.41	0.08	0.13	0.29	0.24
	F - measure	0.47	0.49	0.36	0.61	0.66
	AUC	0.52	0.54	0.45	0.71	0.72

从表中可以看出,与传统的算法相比,文中算法有比较明显的优势。而与 LSRML 算法相比,CSRML 方法获得了较高的 pd 值与 F - measure 值。通过进一步分析可以得出,CSRML 算法的 pd 平均值比 LSRML 高 0.026,F - measure 的平均值比 LSRML 算法的高 0.058。这说明了在度量学习软件缺陷预测中考虑数据集不平衡问题的必要性。

4 结束语

在稀疏重构度量学习的基础上,为度量矩阵学习阶段引入代价敏感因子来减小样本错分代价,同时加入权重来提高小类样本距离度量学习的准确性,并在测试样本分类阶段使用 IWKNN 算法。在 NASA 上 5 个项目的实验结果证明了文中算法的有效性。

参考文献:

[1] 陈 翔,贺 成,王 宇,等. HFS:一种面向软件缺陷预测的混合特征选择方法[J]. 计算机应用研究,2016,33(6):1758-1761.

[2] 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.

[3] ELISH K O,ELISH M O. Predicting defect-prone software modules using support vector machines[J]. Journal of Systems & Software,2008,81(5):649-660.

[4] WANG Tao,LI Weihua. Naive Bayes software defect prediction model[C]//International conference on computational intelligence and software engineering. Wuhan,China:IEEE,2010:1-4.

(下转第 136 页)

程中像素面积的误差值。实验数据如表 1 所示。

表 1 图像面积与误差

序号	像素面积/迭代值下限	像素面积/迭代值上限	像素面积差	误差/%	实际偏差/cm²
1	18 046/27	18 504/45	458	2.54	7.16
2	17 140/27	17 561/45	421	2.46	6.58
3	14 528/27	14 892/45	364	2.51	5.69
4	10 479/27	10 695/45	216	2.06	3.38
5	13 626/27	13 937/45	311	2.28	4.86

4 结束语

提出了一种基于图像识别技术的炉衬风口侵蚀面积识别方法。利用高精度的拍摄系统对炉衬样品进行图像采集,然后采用滤波、边缘检测、图像分割等技术对拍摄到的图片进行处理,获取图像特征,最终实现对侵蚀面积的识别。由于图像检测边缘的模糊性和闭运算结构元素选择的局限性,图像处理存在一定范围内的误差,有待于进一步改善。炉衬风口侵蚀面积检测为以后研究炉衬侵蚀面积与侵蚀程度(深度)数据库研究提供了理论依据。

参考文献:

[1] 张 毅,伍从应,王琳松,等. 100t 转炉炉衬侵蚀因素的分析和长寿炉龄的工艺实践[J]. 特殊钢,2015,36(4):23-27.

[2] 史江涛,蒋仁全,马跃庆. 延长高碳铬铁电炉炉衬寿命的实践[J]. 铁合金,2003,34(4):28-31.

[3] GONZALEZ R C, WOODS R E. Digital image processing [M]. 3rd ed. Beijing: Electronic Industry Press, 2011.

[4] 包晓敏,张云华,汪亚明,等. 基于离散高斯滤波器的纺织品图像增强[J]. 纺织学报,2005,26(4):121-123.

[5] 魏伟波,芮筱亭. 图像边缘检测方法研究[J]. 计算机工程与应用,2006,42(30):88-91.

[6] 周心明,兰 赛,徐 燕. 图像处理中几种边缘检测算法的比较[J]. 现代电力,2000,17(3):65-69.

[7] 靳鹏飞. 一种改进的 Sobel 图像边缘检测算法[J]. 应用光学,2008,29(4):625-628.

[8] NIXON M, AGUADO A S. Feature extraction and image processing[M]. [s.l.]: Academic Press, 2008.

[9] 陈 亮,丁国辉,郭 雷. 基于直方图互确认的图像阈值化分割[J]. 红外与毫米波学报,2011,30(1):80-84.

[10] KAPUR J N, SAHOO P K, WONG A K C. A new method for gray-level picture thresholding using the entropy of the histogram[J]. Computer Vision, Graphics & Image Processing, 1985, 29(3): 273-285.

[11] WANG Qing, CHI Zheru, ZHAO Rongchun. Image thresholding by maximizing the index of nonfuzziness of the 2-D grayscale histogram[J]. Computer Vision & Image Understanding, 2002, 85(2): 100-116.

[12] 朱齐丹,荆丽秋,毕荣生,等. 最小误差阈值分割法的改进算法[J]. 光电工程,2010,37(7):107-113.

[13] 贺建峰,符 增,相 艳,等. 基于灰度空间相关性最大类间方差的图像分割[J]. 计算机工程,2015,41(11):280-286.

[14] 景晓军,李剑峰,刘郁林. 一种基于三维最大类间方差的图像分割算法[J]. 电子学报,2003,31(9):1281-1285.

[15] 毛星云,冷雪飞. OpenCV3 编程入门[M]. 北京:电子工业出版社,2015:327-333.

(上接第 128 页)

[5] WANG Jun, SHEN Beijun, CHEN Yuting. Compressed C4.5 models for software defect prediction [C]//International conference on quality software. Xi'an, Shaanxi, China: IEEE, 2012: 13-16.

[6] ZHENG Jun. Cost-sensitive boosting neural networks for software defect prediction[J]. Expert Systems with Applications, 2010, 37(6): 4537-4543.

[7] WRIGHT J, YANG A Y, GANESH A, et al. Robust face recognition via sparse representation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2008, 31(2): 210-227.

[8] JING Xiaoyuan, YING Shi, ZHANG Zhiwu, et al. Dictionary learning based software defect prediction[C]//International conference on software engineering. [s.l.]: IEEE, 2014: 414-423.

[9] WU Fei, JING Xiaoyuan, DONG Xiwei, et al. Cost-sensitive local collaborative representation for software defect prediction[C]//International conference on software analysis, testing and evolution. Kunming, China: IEEE, 2016: 102-107.

[10] 王 晴,荆晓远,朱阳平,等. 基于局部稀疏重构度量学习的软件缺陷预测[J]. 计算机技术与发展,2016,26(11):54-57.

[11] CASTRO C L, BRAGA A P. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(6): 888-899.

[12] 刘江涛. 距离度量学习中的类别不平衡问题研究[D]. 南京:东南大学,2016.

[13] 王超学,潘正茂,马春森,等. 改进型加权 KNN 算法的不平衡数据集分类[J]. 计算机工程,2012,38(20):160-163.

[14] SHEPPERD M, SONG Qinbao, SUN Zhongbin, et al. Data quality: some comments on the NASA software defect datasets[J]. IEEE Transactions on Software Engineering, 2013, 39(9): 1208-1215.

[15] MENZIES T, GREENWALD J, FRANK A. Data mining static code attributes to learn defect predictors[J]. IEEE Transactions on Software Engineering, 2006, 33(1): 2-13.