基于局部稀疏重构度量学习的软件缺陷预测

王 晴[1],荆晓远[1,2],朱阳平[3],吴 飞[3],董西伟[1],程 立[2]
(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:随着计算机技术的不断发展,如何准确地预测出软件中潜在的缺陷显得至关重要。近年来,研究者们尝试把一些机器学习方法应用到软件缺陷预测领域中,但是这些方法在分类过程中大多使用了传统的欧氏距离。距离度量学习方法通过挖掘训练样本集的特征信息和标记信息,学习得到有效的距离度量,让样本在基于度量矩阵的新特征空间中具有更好的鉴别可分性。将距离度量学习方法引入到软件缺陷预测中,同时融入了局部稀疏重构信息,提出一种新的软件缺陷预测方法,即局部稀疏重构度量学习方法(LSRML)。该方法学习得到的距离度量具有很好的鉴别性,并有效地解决了噪声敏感问题。在软件工程 NASA 数据库上的实验结果表明,提出的方法具有较好的缺陷预测效果。

关键词:度量学习;软件缺陷预测;稀疏表示;局部信息;鉴别性
中图分类号:TP181 **文献标识码**:A **文章编号**:1673-629X(2016)11-0054-04
doi:10.3969/j.issn.1673-629X.2016.11.012

Software Defect Prediction of Metric Learning Based on Local Sparse Reconstruction

WANG Qing[1], JING Xiao-yuan[1,2], ZHU Yang-ping[3], WU Fei[3], DONG Xi-wei[1], CHENG Li[2]
(1. College of Computer, 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. College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: With the development of computer technology, how to predict the potential defects in software project preciously is an important topic. Recently, researchers have introduced some machine learning methods into the software defect prediction field. However, they usually utilize the traditional Euclidean metric in classification phase. Distance metric learning can learn an effective distance metric by exploiting the feature and label information of training sets, which makes the original samples hold better discriminability in the new feature space. The distance metric learning is introduced into the software defect prediction field, and a novel software defect prediction approach called Local Sparse Reconstruction based Metric Learning (LSRML) is proposed. It incorporates the local sparse reconstruction information into the distance metric learning scheme. The learned distance metric not only has favorable discriminability, but also effectively handles the noise problem. The experiment results on the NASA projects demonstrate the effectiveness of the proposed approach.

Key words: distancemetric learning; software defect prediction; sparse representation; local information; discriminability

0 引 言

随着软件在各个领域中的开发规模不断增长,由于软件故障导致巨大损失的事件频有发生,因此如何准确地预测出软件中是否存在潜在缺陷的问题变得十分重要[1-3]。软件缺陷预测(Software Defect Prediction, SDP)技术可以根据软件的基本属性,以及软件模

收稿日期:2016-01-24 **修回日期**:2016-05-11 **网络出版时间**:2016-10-24
基金项目:国家自然科学基金资助项目(61272273)
作者简介:王 晴(1993-),女,研究生,研究方向为软件工程、机器学习与数据挖掘;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.20161024.1114.050.html

块中的历史缺陷数据等信息,来预测开发的软件模块中是否存在缺陷。它对于提高软件质量、缩短开发周期和控制软件开发成本方面有着重要的意义。

目前,软件缺陷预测技术主要分为动态缺陷预测技术和静态缺陷预测技术两种。动态缺陷预测技术是基于缺陷产生的时间,对软件在生命周期或某些阶段的时间关系的缺陷分布进行预测的技术;静态缺陷预测技术是利用软件中已经存在的缺陷以及能够度量缺陷的度量元,结合一些机器学习模型,预测软件中潜在的缺陷[4-5]。文中主要研究静态缺陷预测技术。

静态缺陷预测方法的关键在于如何充分挖掘已有的缺陷数据,构造出更为精确有效的预测模型。目前,已有研究者将传统的机器学习方法成功地应用在软件缺陷预测领域,例如,压缩 C4.5 模型(Compressed C4.5,CC4.5)[6]、朴素贝叶斯模型(Naïve Bayes,NB)[7]、支持向量机模型(Support Vector Machine,SVM)[8]、神经网络模型(Neural Networks,NN)[9-10]等。近年来,一些较新的机器学习方法,如稀疏表示、字典学习等,已经被成功运用到软件缺陷预测中。代价敏感字典学习(Cost-sensitive Discriminative Dictionary Learning,CDDL)[11]融合了字典学习和代价敏感技术,解决了缺陷预测中的类不平衡和错误分类代价问题。协同表示分类模型(Collaborative representation classification based SDP,CSDP)[12]使用协同表示技术代替了稀疏表示应用在缺陷预测中,有效降低了计算复杂度,提高了分类器的性能。

尽管现有的软件缺陷预测方法融入了一些机器学习算法的优点,但是预测效果仍有较大的提升空间。现有相关方法在训练模型阶段或预测阶段中,往往使用欧氏距离来度量样本之间的距离。然而,欧氏距离并不能很好地突显样本之间的鉴别信息。因此文中引入距离度量学习方法(Distance Metric Learning),并融入了局部加权和稀疏重构技术,提出了一种新的软件缺陷预测方法,即基于局部稀疏重构的度量学习方法(Local Sparse Reconstruction based Metric Learning,LSRML)。该方法既可以学习鉴别性很好地距离度量矩阵,又融入了稀疏表示中对噪声鲁棒的优点。文中在 NASA 数据库[13]上的实验结果验证了所提方法的有效性。

1　大间隔最近邻算法

这一节简要介绍距离度量学习中的代表方法,即大间隔最近邻算法(Large Margin Nearest Neighbor,LMNN)[14]。该算法的目标是学习一个距离度量矩阵 $\boldsymbol{M}$,使目标样本与训练集中的近邻同类样本尽量靠近,同时与近邻异类样本尽量远离。

给定一组训练样本 $X=\{x_i,l_i\}$, $i=1,2,\cdots,n$,其中训练样本 $x_i\in\mathbb{R}^d$, d 是样本的维数, l_i 是样本类别标签, $l_i\in\{1,2,\cdots,c\}$。根据马氏距离定义两个样本之间距离的公式为:

$$d_M^2(x_i,x_j)=(x_i-x_j)^{\mathrm{T}}\boldsymbol{M}(x_i-x_j) \tag{1}$$

其中, $\boldsymbol{M}$ 为半正定矩阵。

在 LMNN 中,每个输入样本 x_i 的近邻样本集中都关联着两种样本:一种是与 x_i 类别标签相同的样本,称之为目标样本,记为 x_j,目标样本集记为 $T(x_j)$;另一种是与 x_i 类别标签不同的样本,可称为入侵样本,记为 x_k,入侵样本集记为 $I(x_i)$。因此,LMNN 引入了两个惩罚项。第一项惩罚目标样本与输入样本类别相同但两者距离较远的样本,即: $\sum_i\sum_{j\in T(x_i)}d_M^2(x_i,x_j)$。

第二项惩罚与输入样本标签不同但距离较近的入侵样本,使得任一入侵样本 x_k 与 x_i 的距离都要比目标样本 x_j 与 x_i 的距离至少间隔一个单位,定义非等价约束为: $\sum_{i,j\in T(x_i)}\sum_{k\in I(x_i)}[1+d_M^2(x_i,x_j)-d_M^2(x_i,x_k)]_+$。

结合上述两个惩罚项得到如下损失函数(loss function):

$$\varepsilon(M)=\sum_i\sum_{j\in T(x_i)}d_M^2(x_i,x_j)+\mu\sum_{i,j\in T(x_i)}\sum_{l\in I(x_i)}[1+d_M^2(x_i,x_j)-d_M^2(x_i,x_k)]_+ \tag{2}$$

其中, $d_M^2(x_i,x_j)=(x_i-x_j)^{\mathrm{T}}\boldsymbol{M}(x_i-x_j)$,并且 $\boldsymbol{M}\geqslant 0$。

将上式转化为凸规划问题求解,并引入非负松弛变量 $\{\xi_{ijk}\}$,可将其转化为半正定规划问题:

$$\begin{aligned}&\min\sum_{i,j\in T(x_i)}(x_i-x_j)^{\mathrm{T}}\boldsymbol{M}(x_i-x_j)+\mu\sum_{\substack{i,j\in T(x_i)\\k\in I(x_i)}}\xi_{ijk}\\&\text{s.t.}\ (x_i-x_k)^{\mathrm{T}}\boldsymbol{M}(x_i-x_k)-(x_i-x_j)^{\mathrm{T}}\boldsymbol{M}(x_i-x_j)\geqslant 1-\xi_{ijk},\xi_{ijk}\geqslant 0,\boldsymbol{M}\geqslant 0\end{aligned} \tag{3}$$

2　基于局部稀疏重构的度量学习方法(LSRML)

给定样本集 $X=[X_1,X_2,\cdots,X_c]$,第 t 类中任意样本记为 $x_i^t\in\mathbb{R}^d$, $t=1,2,\cdots,c$, c 为类别个数,每类的样本数为 n_t,样本总数为 N。给定一个测试样本 $y\in\mathbb{R}^d$,传统的稀疏表示方法要求训练样本集 X 可以联合表示测试样本,同时表示稀疏 α 满足稀疏性,如下:

$$\arg\min_{\alpha}\|y-X\alpha\|_2^2+\sigma\|\alpha\|_1 \tag{4}$$

其中, $\|\alpha\|_1$ 用来强制稀疏约束; σ 用来平衡重构误差和重构系数的稀疏性。

在式(4)中,稀疏重构误差项 $\|y-X\alpha\|_2^2$ 使用的

是传统的欧氏距离度量方式,而欧氏距离并不能很好地体现样本之间的鉴别信息。

文中专门学习一个距离度量方式,并将其融入到稀疏表示中,即: $\arg\min_{\alpha} d_M^2(y,X)$。

其中, $d_M^2(y,X)=(y-X\alpha)^{\mathrm{T}}\boldsymbol{M}(y-X\alpha)+\sigma\|\alpha\|_1$, $\boldsymbol{M}$ 为半正定矩阵,即需要学习的距离度量矩阵。

为了增强距离度量矩阵 $\boldsymbol{M}$ 的鉴别性,设计了类内稀疏重构项和类间稀疏重构项。对于每个训练样本 x_i, $i=1,2,\cdots,N$,把剩余的训练样本划分为两个样本集 A 和 B。其中, $A=[a_{i1},a_{i2},\cdots,a_{iN_1}]$ 表示和 x_i 标记一致的样本; $B=[b_{i1},b_{i2},\cdots,b_{iN_2}]$ 表示和 x_i 标记不一致的样本。类内稀疏重构项和类间稀疏重构项分别表示为:

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

$$d_{1,M}^2(x_i,B)=(x_i-B\gamma)^{\mathrm{T}}\boldsymbol{M}(x_i-B\gamma)+\sigma\|\gamma\|_1 \tag{6}$$

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

为了突出样本近邻信息在稀疏表示时的重要性,在式(5)和式(6)的基础上,让与 x_i 同类的近邻样本所对应的稀疏系数更大,与 x_i 异类的近邻样本所对应的稀疏系数更小,即:

$$d_{1,M}^2(x_i,A)=(x_i-A\beta)^{\mathrm{T}}\boldsymbol{M}(x_i-A\beta)+\sigma\|D_i\beta\|_1 \tag{7}$$

$$d_{1,M}^2(x_i,B)=(x_i-B\gamma)^{\mathrm{T}}\boldsymbol{M}(x_i-B\gamma)+\sigma\|\tilde{D}_i\gamma\|_1 \tag{8}$$

其中, D_i、$\tilde{D}_i$ 分别是稀疏系数 β 和 γ 的局部权重,定义为:

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

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

其中, $r_1>r_2$。

可以看出,在距离相同的情况下,同类近邻样本对应的权重更小,放在最小化的目标函数中,求得的表示系数就越大,这样近邻样本在稀疏表示时就更加重要。

根据上面定义的局部加权类内稀疏重构项和类间稀疏重构项,并借鉴 LMNN 中最大间隔的思想,则 LSRML 的目标函数定义为:

$$\begin{aligned}\min_{\boldsymbol{M},\beta,\gamma} g(\boldsymbol{M},\beta,\gamma)=&\sum_i d_{1,\boldsymbol{M}}^2(x_i,A)+\mu\sum_i\xi_i+\\&\nu\sum_i(\|D_i\beta_i\|_1+\|\tilde{D}_i\gamma_i\|_1)\\ \text{s.t. } d_{2,\boldsymbol{M}}^2(x_i,B)&-d_{1,\boldsymbol{M}}^2(x_i,A)\geqslant 1-\xi_i,\xi_i\geqslant 0,\\ \boldsymbol{M}&\geqslant 0\end{aligned} \tag{9}$$

式(9)可以使用交替优化(Alternating Optimization)的方式来求解。目标函数中总共有三个未知参数 $\boldsymbol{M}$,β 和 γ,先固定 $\boldsymbol{M}$,求解 β 和 γ;然后固定 β 和 γ,求解 $\boldsymbol{M}$。

首先,初始化距离度量矩阵 $\boldsymbol{M}$ 为欧氏距离度量矩阵,即 $\boldsymbol{M}=\boldsymbol{I}^{d\times d}$。此时目标函数转化为:

$$\beta^*=\arg\min\|D_i\beta\|_1\quad \text{s.t. } \|x_i-A\beta\|_2^2\leqslant\varepsilon \tag{10}$$

$$\gamma^*=\arg\min\|\tilde{D}_i\gamma\|_1\quad \text{s.t. } \|x_i-B\gamma\|_2^2\leqslant\varepsilon \tag{11}$$

其中,容错误差 $\varepsilon>0$。

式(10)、(11)为标准的 l_1 范数的最小化问题,这和稀疏表示中的目标函数类似,可以采用文献[15-16]中的优化算法求解。

得到 β 和 γ 后,目标函数(式(9))可以简化为求解矩阵 $\boldsymbol{M}$ 的函数:

$$\begin{aligned}\min_{\boldsymbol{M}} g(\boldsymbol{M})=&\sum_i d_{1,\boldsymbol{M}}^2(x_i,A)+\mu\sum_i\xi_i\\ \text{s.t. } d_{2,\boldsymbol{M}}^2(x_i,B)&-d_{1,\boldsymbol{M}}^2(x_i,A)\geqslant 1-\xi_i,\xi_i\geqslant 0,\\ \boldsymbol{M}&\geqslant 0\end{aligned} \tag{12}$$

这是一个典型的半正定规划问题,可以通过一个标准的半正定规划工具包进行求解。文中使用了 *cvx* 工具包。

由于 $\boldsymbol{M}$ 是半正定矩阵,可以将 $\boldsymbol{M}$ 写成 $\boldsymbol{M}=\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$,这里 $\boldsymbol{W}$ 是一个线性转换: $\mathbb{R}^d\rightarrow\mathbb{R}^d$。其中, x_i 通过学习到 $\boldsymbol{W}$ 不断更新: $x_i=\boldsymbol{W}^{\mathrm{T}}x_i$,同时通过求解式(10)~(12)不断更新 β, γ 和 $\boldsymbol{M}$。综上所述,LSRML 算法流程可以总结为:

输入:训练样本集 $X=[X_1,X_2,\cdots,X_c]$,收敛误差 τ。

输出:距离度量矩阵 $\boldsymbol{M}$。

步骤 1:初始化矩阵 $\boldsymbol{M}$: $\boldsymbol{M}=\boldsymbol{I}^{d\times d}$。

步骤 2:令 $r=1,2,\cdots$,循环

(1)根据式(10)和式(11)计算 β 和 γ。

(2)根据式(12)求解得到矩阵 $\boldsymbol{M}$。

(3)分解 $\boldsymbol{M}=\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$。

(4)更新训练样本 $x_i=\boldsymbol{W}^{\mathrm{T}}x_i$。

(5)若 $r>2$ 且 $|\boldsymbol{M}^r-\boldsymbol{M}^{r-1}|<\tau$,跳到步骤 3。

步骤 3:输出度量矩阵 $\boldsymbol{M}=\boldsymbol{M}^r$。

3 实 验

本节首先介绍实验所用的数据库,以及缺陷预测的评价指标,然后报告并分析文中 LSRML 和对比方法

的实验结果。

3.1　数据库介绍

实验选用了 NASA MDP 数据库[13]的 5 个工程,每个工程代表着美国宇航局(NASA)的软件系统或者子系统,它们包含不同的静态代码度量和相应的缺陷标记数据。这些数据库通过一个 bug 跟踪系统记录每个模块的缺陷数。NASA MDP 数据库的静态代码度量指标包括软件代码量、可读性、复杂度等等。这些分别由代码行数、操作数以及 McCabe 等度量计算得到。表 1 汇总了 NASA MDP 中 5 个工程的详细信息。

表 1　NASA 数据集

数据集	缺陷样本数	样本总数	特征数	缺陷样本占比/%
CM1	42	344	37	12.21
MW1	27	255	37	10.59
PC1	61	711	37	8.58
PC3	134	1 079	37	12.42
PC4	177	1 288	37	13.74

3.2　性能评价指标

在实验中,使用四种指标来评估方法的缺陷预测效果,即召回率(Recall, P_d)、False Positive Rate (P_f)、F-measure 和 Area Under roc Curve(AUC)。

假设 A 代表有缺陷样本被预测为有缺陷的数量,B 代表有缺陷样本被预测为无缺陷的数量,C 代表无缺陷样本被预测为有缺陷的数量,D 代表无缺陷样本被预测为无缺陷的数量,如表 2 所示。

表 2　四种预测结果

	Predict as defective	Predict as defect-free
Defective modules	A	B
Defective-free modules	C	D

则以上四种指标定义为:$P_d = A/(A + B)$;$P_f = C/(C + D)$;F - measure = 2 * recall * precision/(recall + precision),其中 precision = $A/(A + C)$;AUC 为 ROC 曲线下面积。

这四种评价指标值都在 0 ~ 1 之间,一个好的缺陷预测模型应该会有较高的 P_d,F-measure 和 AUC 值,以及较小的 P_f 值。而且 F-measure 和 AUC 是综合性评价指标,更加重要。

3.3　实验结果与分析

文中选取了几种代表性的缺陷预测方法作为对比方法,包括 CC4.5[6]、NB[7]、SVM[8] 和 CBNN[10](cost-sensitive boosting neural networks)。此外,由于提出的 LSRML 方法融入了距离度量学习,所以也选取 LMNN[14] 作为对比方法之一。实验结果见表 3。

分析表 3 可知,文中提出的 LSRML 在各个数据库上的缺陷预测效果普遍好于对比方法,尤其是 F-measure 和 AUC 评价指标。对于基于传统机器学习方法的 CC4.5、SVM、NB 和 CBNN,LSRML 优势较明显,说明了使用距离度量学习得到的度量矩阵 $\boldsymbol{M}$ 要优于传统的欧氏距离,距离度量学习技术在软件缺陷预测领域是有效的;对于代表性的距离度量学习方法 LMNN,LSRML 方法的优势说明局部稀疏重构项在缺陷预测时的有效性。

表 3　所有方法在 NASA MDP 数据库上的实验结果

方法	CM1				MW1				PC1				PC3				PC4			
	P_d	P_f	F-measure	AUC	P_d	P_f	F-measure	AUC	P_d	P_f	F-measure	AUC	P_d	P_f	F-measure	AUC	P_d	P_f	F-measure	AUC
CC4.5	0.26	0.11	0.25	0.43	0.29	0.17	0.27	0.43	0.38	0.09	0.32	0.50	0.34	0.09	0.29	0.52	0.49	0.08	0.49	0.54
NB	0.44	0.18	0.32	0.52	0.49	0.19	0.31	0.41	0.36	0.11	0.28	0.46	0.28	0.09	0.29	0.47	0.39	0.13	0.36	0.45
SVM	0.15	0.04	0.20	0.45	0.21	0.15	0.27	0.42	0.66	0.10	0.35	0.59	0.64	0.19	0.28	0.53	0.72	0.41	0.47	0.52
CBNN	0.59	0.29	0.33	0.63	0.61	0.25	0.33	0.52	0.54	0.17	0.32	0.51	0.65	0.25	0.38	0.62	0.66	0.18	0.46	0.51
LMNN	0.50	0.21	0.36	0.59	0.55	0.21	0.31	0.48	0.58	0.28	0.35	0.60	0.65	0.23	0.35	0.59	0.62	0.31	0.40	0.49
LSRML	0.66	0.17	0.51	0.72	0.67	0.15	0.51	0.68	0.70	0.22	0.53	0.76	0.78	0.27	0.64	0.72	0.75	0.29	0.61	0.71

4　结束语

文中首次将距离度量学习方法引入到软件缺陷预测中,并且融入了稀疏重构项和样本的局部近邻信息,提出一种新的软件缺陷预测方法,即 LSRML。该方法学习得到的距离度量具有很好的鉴别性。NASA MDP 上 5 工程的数据库表明,LSRML 与现有的代表性缺陷预测方法相比,提高了缺陷预测的效果。

参考文献:

[1]　刘英博,王建民. 面向缺陷分析的软件库挖掘方法综述[J]. 计算机科学,2007,34(9):1-4.

[2]　刘义颖,江建慧. 基于软件失效链的软件错误行为分类研究[J]. 计算机技术与发展,2015,25(4):1-5.

[3]　李　娟,陈　斌. 一种基于 JM 模型的软件安全性测试方法研究[J]. 计算机技术与发展,2012,22(9):246-249.

(下转第 62 页)

由于 ASIFT 是在提取图像特征点之前对图像中的所有像素点构建仿射空间[13-14],也就是进行仿射变换的模拟,这样虽然使得图像特征点的数量得到了提升,但是图像特征点的匹配率却没有提升或提升的效果不明显。通过对 SIFT 算法特征点提取完成后,对提取到的特征点进行仿射空间的构建,这样就能提升特征点的抗仿射能力,进而提升了特征点的匹配率,同时也降低了在进行仿射空间构建时的计算量,节约了匹配时间,提升了匹配速率。该算法的缺点在于特征点的数量没有 ASIFT 算法那么多,仅仅是继承了 SIFT 算法的特征点数量,所以还值得深入研究。

参考文献:

[1] Lowe D G. Distinctive image features from scale-invariant key-point[J]. International Journal of Computer Vision,2004,60(2):91-110.

[2] Ye K,Sukthankar R. PCA-SIFT:a more distinctive representation for local image descriptors[C]//Proceedings of the conference on computer vision and pattern recognition. [s. l.]:IEEE,2004:90-98.

[3] Abdel-Hakim A E,Farag A A. CSIFT:ASIFT descriptor with color invariant characteristics[C]//IEEE computer society conference on computer vision and pattern recognition. [s. l.]:IEEE,2006:1978-1983.

[4] Bay H,Ess A,Tuytelars T,et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding,2008,110(3):346-359.

[5] Morel J M,Yu Guoshen. ASIFT:a new framework for fully affine invariant image comparison[J]. SIAM Journal on Imaging Sciences,2009,2(2):438-469.

[6] Lindeberg T. Scale-space theory:a basic tool for analysing structures at different scales[J]. Journal of Applied Statistics,1994,21(2):223-261.

[7] Lindeberg T. Scale-space theory in computer vision[M]. [s. l.]:Springer Science & Business Media,1994.

[8] Babaud J,Witkin A P,Baudin M,et al. Uniqueness of the Gaussian kernel for scale-space filtering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,1986,8(1):26-33.

[9] 卢 彬. 改进的 ASIFT 算法在图像配准中的应用[J]. 电视技术,2014,38(11):211-214.

[10] 周 颖. 基于 SIFT 算法的图像特征匹配[J]. 现代计算机,2015(2):63-68.

[11] 朱 进,丁亚洲,肖雄武,等. 基于 SIFT 改进算法的大幅面无人机影像特征匹配方法[J]. 计算机应用研究,2015,32(10):3156-3159.

[12] 张忠林,曹志宇,李元韬. 基于加权欧式距离的 k_means 算法研究[J]. 郑州大学学报:工学版,2010,31(1):89-92.

[13] 何婷婷,芮建武,温 腊. CPU-GPU 协同算加速 ASIFT 算法[J]. 计算机科学,2014,41(5):14-19.

[14] 宋耀鑫,张丹丹,唐伶俐,等. 基于 ASIFT 算法的低重叠度无人机影像拼接方法[J]. 遥感技术与应用,2015,30(4):725-730.

(上接第 57 页)

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

[5] 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,2011,38(6):1276-1304.

[6] Wang J,Shen B J,Chen Y T. Compressed C4.5 models for software defect prediction[C]//2012 12th international conference on quality software. [s. l.]:IEEE,2012:13-16.

[7] Wang T,Li W H. Naïve Bayes software defect prediction model[C]//International conference on computational intelligence and software engineering. [s. l.]:IEEE,2010:1-4.

[8] Elish K,Elish M. Predicting defect-prone software modules using support vector machines[J]. Journal Systems and Software,2008,81(5):649-660.

[9] 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.

[10] Zheng J. Cost-sensitiveboosting neural networks for software defect prediction[J]. Expert Systems with Applications,2010,37(6):4537-4543.

[11] Jing X Y,Ying S,Zhang Z W,et al. Dictionary learning based software defect prediction[C]//Proceedings of the 36th international conference on software engineering. [s. l.]:ACM,2014:414-423.

[12] Jing X Y,Zhang Z W,Ying S,et al. Software defect prediction based on collaborative representation classification[C]//Proceedings of the 36th international conference on software engineering. [s. l.]:ACM,2014:632-633.

[13] 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.

[14] Weinberger K Q,Saul L K. Distance metric learning for large margin nearest neighbor classification[J]. Journal of Machine Learning Research,2009,10(1):207-244.

[15] Donoho D L,Tsaig Y. Fast solution of l_1-norm minimization problems when the solution may be sparse[J]. IEEE Transactions on Information Theory,2008,54(11):4789-4812.

[16] Wright J,Yang A Y,Ganesh A,et al. Robust face recognition via sparse representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2009,31(2):210-227.