◎工程与应用◎

基于贝叶斯Logistic回归的软件缺陷预测研究

赖永凯[1],陈向宇[2],刘 海[2]
1.韶关学院 教育学院,广东 韶关 512005
2.华南理工大学 计算机科学与工程学院,广州 510006

摘 要:在软件开发初期及时识别出软件存在的缺陷,可以帮助项目管理团队及时优化开发测试资源分配,以便对可能含有缺陷的软件进行严格的质量保证活动,这对于软件的高质量交付有着重要的作用,因此,软件缺陷预测成为软件工程领域内一个研究热点。虽然人们已经使用多种机器学习算法建立了缺陷预测模型,但还没有对这些模型的贝叶斯方法进行研究。提出了无信息先验和信息先验的贝叶斯Logistic回归方法来建立缺陷预测模型,并对贝叶斯Logistic回归的优势以及先验信息在贝叶斯Logistic回归中的作用进行了研究。最后,在PROMISE数据集上与其他已有缺陷预测方法(LR、NB、RF、SVM)进行了比较研究,结果表明:贝叶斯Logistic回归方法可以取得很好的预测性能。

关键词:缺陷预测;贝叶斯Logistic回归;信息先验
文献标志码:A **中图分类号:**TP311 doi:10.3778/j.issn.1002-8331.1812-0254

赖永凯,陈向宇,刘海.基于贝叶斯Logistic回归的软件缺陷预测研究.计算机工程与应用,2019,55(11):204-208.
LAI Yongkai, CHEN Xiangyu, LIU Hai. Research on software defect prediction based on Bayesian Logistic regression. Computer Engineering and Applications, 2019, 55(11):204-208.

Research on Software Defect Prediction Based on Bayesian Logistic Regression

LAI Yongkai[1], CHEN Xiangyu[2], LIU Hai[2]
1.Institute of Education, Shaoguan University, Shaoguan, Guangdong 512005, China
2.School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China

Abstract: Software defect prediction can help project management team to optimize development and test resources in time, so as to carry out strict quality assurance activities for software modules that may contain defects, which plays an important role in the high-quality delivery of software. Therefore, software defect prediction has become a research hot topic in the field of software engineering. Though defect prediction models have been built using several machine learning algorithms, Bayesian approach of these models is not explored. Bayesian Logistic regression method with non-informative and informative priors is proposed to build defect prediction models. The advantages of Bayesian Logistic regression and the role of priors in the performance of Bayesian logistic regression are studied. Finally, compared with other existing defect prediction methods(LR, NB, RF, SVM) on PROMISE dataset, the results show that Bayesian Logistic regression method can achieve good prediction performance.

Key words: software defect prediction; Bayesian Logistic regression; informative priors

1 引言

提供高质量的软件是软件公司在激烈的市场竞争中生存的最重要目标之一。软件缺陷是影响软件质量的首要因素,在软件研制过程中,尽管花费了大量时间

基金项目:国家自然科学基金(No.71502215);广东省科技创新战略专项项目(No.2016A030305001)。
作者简介:赖永凯(1979—),男,讲师,研究领域为可信软件,教育信息化,E-mail:dengfeisouthwest@163.com;陈向宇(1982—),男,博士,副教授,研究领域为计算机技术;刘海(1992—),男,硕士,研究领域为软件测试。
收稿日期:2018-12-20 **修回日期:**2019-01-18 **文章编号:**1002-8331(2019)11-0204-05
CNKI**网络出版:**2019-01-29, http://kns.cnki.net/kcms/detail/11.2127.tp.20190128.1728.003.html

来进行代码审查、软件测试等质量保证活动,但基本上研制后期还是会发现新的缺陷,而越到研制后期,软件缺陷的修改代价就越大。软件缺陷预测的主要目的是根据历史数据来预期软件中可能存在的潜在缺陷,以便项目管理人员更合理地分配开发资源和质量保证资源,比如对容易产生缺陷的文件进行代码审查、专家测试等。近年来,软件缺陷预测是软件工程领域的研究热点,陈翔等[1]针对该问题进行系统的分析、总结和比较:研究人员基于Logistic回归(LR)、朴素贝叶斯分类(NB)、随机森林(RF)、支持向量机(SVM)等机器学习算法开发了各类缺陷预测模型。

Logistic回归是一种概率判别模型,将Logistic回归应用于软件缺陷预测问题时还存在一些不足之处。

(1)极大似然估计方法(Maximum Likelihood Estimate,MLE)是渐近无偏的。已经证明,对于小样本和中等样本,回归系数是有偏的[2],Long[3]就样本大小进行了初步探索:在样本小于100的情况下使用MLE是有风险的,而大于600的样本就基本没有风险了。而本文在研究中使用6个数据集的平均文件数只有400个。

(2)当非事件(无缺陷)和事件(有缺陷)以相同的概率发生时,应用Logistic回归来做缺陷预测具有效果很好,但在实际工程中,通常情况下无缺陷的文件占比更大。

如前所述,Logistic回归假定具有统一的先验,但在实践中,先验可以遵循任何分布。在贝叶斯Logistic回归中,先验 ω 是根据以前的数据确定的。假定先验符合正态分布 $\omega \approx N(\mu, C)$,其中 μ 为均值,C 为协方差,那么后验分布是:

$$P(\omega|D) \propto P(D|\omega) * P(\omega) \tag{1}$$

$$P(\omega|D) \propto \left(\prod_{1}^{n} p_i^{y_i} * (1-p_i)^{1-y_i} \right) * \frac{\exp\left(\frac{-(\omega-\mu)^{\mathrm{T}} C^{-1} (\omega-\mu)}{2}\right)}{\sqrt{\det(2\pi C)}} \tag{2}$$

研究表明,贝叶斯Logistic回归可以提高学习模型的预测精度[4-5],基于此,对贝叶斯Logistic回归进行了研究,并试图回答以下问题:

问题1:Logistic回归和贝叶斯Logistic回归是否存在明显差异?

问题2:使用信息先验和无信息先验开发的分类器的后验概率是否有显著差异?

问题3:贝叶斯Logistic回归与其他广泛使用的分类器在在缺陷预测问题上是否存在显著差异?

2 相关工作

目前,人们已经用各种度量元构建了缺陷预测模型,早期研究表明,代码行数(LOC)与缺陷有很强的相关性,文献[5]的研究表明,代码行数会增加缺陷发生的概率。但LOC度量元不能合理地去度量软件系统的复杂性[1],随后,研究人员逐渐考虑了Halstead科学度量[6]和McCabe环路复杂度(cyclomatic complexity)[7]。随着面向对象开发方法的普及,研究人员提出了适用于面向对象程序的度量元,其中最为典型的是Chidamber和Kemerer提出的CK度量元[8],CK度量元综合考虑了面向对象程序中的继承、耦合性和内聚性等特征,给定一个类,其包含的度量元名称及相关描述见表1。

表1 CK度量元

名称	描述
WMC	类的加权方法数
DIT	类在继承树中的深度
NOC	类在继承树中的孩子节点数
CBO	与该类存在耦合关系的其他类的数目
RFC	该类可以调用的外部方法数
LCOM	类内访问一个或多个属性的方法数

文献[9]研究了代码复杂度和缺陷之间的关系,并得出结论:复杂性度量在缺陷预测中非常有用,且经历大量演化的代码库更容易出现缺陷,这些指标被用于设计各种分类算法。关于缺陷预测算法也有很多研究。Menzies等人[10]和宋擒豹等人[11]对通用缺陷预测框架展开了深入的研究,并得出结论:不存在一种方案,可以在任何数据集上均获得最优性能。一些研究人员尝试借助目前机器学习领域的最新研究进展来做缺陷预测,例如主动学习(active learning)和半监督学习(semi-supervised learning)等。黎铭等人[12]借助基于半监督学习和主动学习的采样方法来从大型软件系统中选出少量代表性程序模块进行标记,并随后构建缺陷预测模型。文献[13]也提出了一种基于搜索的半监督集成跨项目软件缺陷预测方法,大部分情况下该方法可以取得很好的预测性能,文献[14]将随机森林应用于故障易发模块的预测,发现应用于大型数据集时随机森林效果更好。陈琳设计了一种基于迁移Boosting的缺陷预测模型[15],对于缺乏历史数据的软件缺陷预测提供了一个可行的解决方案。Rajni Jindal等人研究了基于神经网络的缺陷预测模型[16],对于部分软件也可以取得很好的预测效果。

3 贝叶斯Logistic回归模型

本文从关于软件缺陷最常用的公开数据集PROMISE中选取了6个数据集作为测试数据库进行实证研究[17],如表2所示,这些项目的平均文件数为400个。

如在第1章中所述,先验 $p(\omega)$ 可以是任何分布,且

表2 选择PROMISE数据库的6个数据集 %

数据集	版本	样本数	缺陷率
Ant	1.3	125	16.00
	1.4	178	22.40
	1.5	293	10.90
	1.6	351	26.20
	1.7	745	22.20
Camel	1.0	339	3.80
	1.2	608	35.50
	1.4	872	16.60
	1.6	965	19.48
Ivy	1.1	111	56.70
	1.4	241	6.60
	2.0	352	11.30
Jedit	3.2	272	33.09
	4.0	306	24.53
	4.1	312	25.31
	4.2	367	13.08
	4.3	492	2.23
Log-4j	1.0	135	25.18
	1.1	109	33.94
	1.2	205	92.19
Lucene	2.0	195	46.67
	2.2	247	58.30
	2.4	340	59.70

有两种类型:信息先验和无信息先验。信息先验可以从历史数据或专业知识领域中获得,而无信息先验不依赖于先前的数据,但可以通过训练数据来近似。

在本文的研究中,使用的无信息先验为Jeffery先验,并提出了一些遵循多元正态分布的信息先验。

3.1 先验选择

3.1.1 Bayes-1先验

Bayes-1先验是信息先验。同一软件的当前版本是前一版本的后验。例如,表2中项目Ivy有3个版本,假设各向同性正态分布 $\omega \approx N(0,\alpha^{-1}I)$ 作为先验,那么可以获得Ivy1.1的后验为:

$$p(\omega|D_{\text{Ivy1.1}})=p(D_{\text{Ivy1.1}}|\omega)*N(0,\alpha^{-1}I) \tag{3}$$

上述后验分布的正态近似将用作Ivy1.4的先验:

$$p(\omega|D_{\text{Ivy1.4}})=p(D_{\text{Ivy1.4}}|\omega)*p(\omega|D_{Ivy1.1}) \tag{4}$$

3.1.2 Bayes-2先验

Bayes-2先验也是信息先验,还是用表2中的Ivy来阐述:

(1)选择Ivy1.1,令 $D_{\text{ivy1.1}}=66\%$,利用此数据进行Logistic回归,得到的参数为:

$$\omega_1=\arg\max_{\omega}\left(p(D_{\text{Ivy1.1}}|\omega)\right) \tag{5}$$

(2)择择Ivy1.4,令 $D_{\text{ivy1.4}}=66\%$,利用此数据进行Logistic回归,得到的参数为:

$$\omega_2=\arg\max_{\omega}\left(p(D_{\text{Ivy1.4}}|\omega)\right) \tag{6}$$

重复上述两个步骤30次,从Ivy1.1和Ivy1.2各得到30个参数。

(3)找出这60个点的最佳拟合多元正态。

(4)使用此多元正态作为Ivy2.0的先验。

3.1.3 Jeffery先验

Jeffery先验是无信息先验 $p(\omega)\propto\sqrt{\det(I(\omega))}$,其中 $I(\omega)$ 是Fisher信息矩阵。在Logistic回归中的Fisher信息矩阵是 $I_{i,j}(\omega)=-E\left[\frac{\delta^2\ln(Likelihood)}{\delta\omega_i\delta\omega_j}\right]$。那么其后分布为:

$$P(\omega|D)\propto P(\omega|D)\sqrt{\det(I(\omega))} \tag{7}$$

3.2 后验取样

在Logistic回归中,得到点估计 ω_{MLE},在贝叶斯方法中,从后验 $P(\omega|D)$ 中抽取一组样本,假设从 $P(\omega|D)$ 中抽取 k 个样本为 $\omega_1,\omega_2,\cdots,\omega_k$,那么第 x 个样本有缺陷的概率为:

$$P(y=1|x,D)=\frac{1}{k}\sum_{1}^{k}1/(1+e^{-\omega_i^{\mathrm{T}}x}) \tag{8}$$

下面讨论从后验 $P(\omega|D)$ 中提取样本的两种方法。

拉普拉斯逼近:后验近似为均值 $\omega_{\max}$ 和协方差矩阵 K 的多元正态分布,其中 $\omega_{\max}=\arg\max_{\omega}\left(p(\omega|D)\right)$、$K=H-1$。从该近似中抽样,只有当 p_{df} 单峰时拉普拉斯近似效果才好,所以本文中没有使用它。

MCMC:另一种从后验抽样的方法是著名的马尔科夫链蒙特卡洛方法(Markov Chain Monte Carlo,MCMC),它需要构造一个以期望分布作为平衡分布的马尔科夫链。在本文的研究中,使用MCMC法中的gibbs抽样。

4 实验

4.1 评价指标

评价指标对于判定预测模型的性能至关重要,在本文的研究中,对软件缺陷预测的结果为有缺陷或无缺陷两种类别,因此其实质是一个二分类问题,即有缺陷为1,无缺陷为0。根据预测结果可以得到一个混淆矩阵,如表3所示。

表3 混淆矩阵

实际结果	预测结果	
	有缺陷	无缺陷
有缺陷	TP(True Positive)	FN(False Negative)
无缺陷	FP(False Positive)	TN(True Negative)

表3中FP代表没有缺陷被预测为有缺陷、TN代表没有缺陷被成功预测为没有缺陷、FN代表有缺陷被错误预测为没有缺陷、TP代表有缺陷被正确预测。

定义查准率(Precision,P)和缺陷检出率(Probability of Detection,PD)为:

$$P=\frac{TP}{TP+FP} \tag{9}$$

$$PD=\frac{TP}{TP+FN} \tag{10}$$

根据以上定义,可以得到一个好的预测模型应该同时具有较高的 P 和 PD,因此,本文用两者的加权平均值F_measure($F1$)来综合评价一个缺陷预测模型的性能:

$$F1=\frac{2\times PD\times P}{PD+P} \tag{11}$$

$F1$ 分布在0~1,其值越接近1表示缺陷预测模型的性能越好。

4.2 实验方法

为了回答问题1和问题2,用Bayes-1先验、Bayes-2先验和Jeffery先验分别进行贝叶斯Logistic回归,为了回答问题3,把贝叶斯Logistic回归与随机森林,朴素贝叶斯和支持向量机进行了比较。详细实验内容如下:

(1)Bayes-1、Bayes-2:两种贝叶斯Logistic回归用贝叶斯逻辑包[18]的logit函数实现,该函数需要数据、先验均值和先验方差作为输入。

(2)Jeffery先验(JP):Jeffery先验的Logistic回归用logistf[19]软件包实现,该软件包通过计算Fisher矩阵返回一组后验样本。

(3)随机森林:随机森林用randomforest[20]函数,实验后将ntree变量设置为100。

(4)朴素贝叶斯:naiveBayes[20]函数用于实现朴素贝叶斯算法。

(5)Logistic回归:用glmnet[21]实现Logistic回归算法,采用交叉验证的方法计算正则化参数。

(6)svm:svm[20]函数包用于实现svm。

4.3 训练与测试

每个模型都在表2中提到的每个软件项目的66%上进行训练,并在同一项目的剩余34%上进行了测试。对于贝叶斯方法,先验来自项目的先前版本,还是以Ivy项目为例来说明训练及测试过程:

(1)在Ivy2.0的66%上训练RF、LR、NB、SVM。

(2)在Ivy2.0的66%上训练JP、Bayes 1和Bayes 2,先验通过于Ivy1.1、Ivy1.4近似。

(3)在Ivy2.0剩余的34%上测试所有模型并计算所有模型的 $F1$。

通过 $F1$ 来比较上述分类方法的性能,每个模型的最终 $F1$ 为多次测试的平均值。

4.4 分类器的统计检验

分类器的性能是根据计算得到的 $F1$ 平均值进行比较的,为了进一步验证各个分类器差异的显著性,在本文研究中使用的统计检验如下。

4.4.1 Wilcoxon符号秩检验

采用Wilcoxon符号秩检验来判断Logistic回归和贝叶斯Logistic回归是否存在明显差异(问题1),显著性水平设置为5%,即置信度为95%。

这里的零假设是:

H0:这两种模型具有相同的性能。

H1:这两种模型的性能具有显著差异。

如果 $p_{\text{value}}<0.05$,则拒绝H0,认为两种模型具有显著差异。

4.4.2 *K*样本检验-Friedman检验

这里的零假设是:

H0:所有分类器的性能相同。

H1:至少2分类器性能不同。

Friedman检验为每个分类器分配一个平均秩,如果零假设不成立,则使用Nemenyi检验对所有分类器进行比较。两个分类器的平均秩必须相差一个临界差(CD)才能被认为是显著不同的。

$$CD=q_a;\infty;L\sqrt{\frac{L(L+1)}{12K}} \tag{12}$$

其中,L 是分类器的数目,K 是项目的数目。

5 结果和讨论

为了回答问题1和问题3,用具有Bayes-1先验、Bayes-2先验和Jeffery先验的贝叶斯Logistic回归,以及Logistic回归(LR)、朴素贝叶斯分类(NB)、随机森林(RF)、支持向量机(SVM)7种预测模型对表2中的数据集进行实验,表4列出了每个项目运行1 000的平均 $F1$,并利用Wilcoxon符号秩检验来回答问题1:Logistic回归与贝叶斯Logistic回归是否存在显著性差异。

表4 比较不同模型的*F*1值

数据集	Bayes1	Bayes2	JP	LR	NB	RF	SVM
ANT	0.78	0.76	0.81	0.80	0.67	0.82	0.64
CAMEL	0.67	0.65	0.74	0.73	0.61	0.71	0.56
JEDIT	0.77	0.78	0.60	0.59	0.76	0.79	0.77
IVY	0.66	0.66	0.68	0.67	0.66	0.66	0.53
LOG4J	0.53	0.71	0.05	0.04	0.48	0.89	0.90
LUCENE	0.65	0.65	0.22	0.23	0.32	0.73	0.59

表5列出了在显著性水平为5%下,具有Bayes-1先验、Bayes-2先验和Jeffery先验的贝叶斯Logistic回归方

法与Logistic回归方法的Wilcoxon符号秩检验结果,分别表示为P(B1-LR)、P(B2-LR)、P(JP-LR)。

表5 Wilcoxon符号秩检验结果

Wilcoxon符号秩检验	P(B1-LR)	P(B2-LR)	P(JP-LR)
结果	5.10E−04	6.090E−04	0.474

从Wilcoxon符号秩检验结果看,具有Bayes-1先验、Bayes-2先验的的贝叶斯Logistic回归方法显著优于Logistic回归方法(问题1)。本文的研究对象的数据点平均值(400个)很小,因而可能导致估计偏差,这可能是导致Logistic回归性能下降的一个原因,此外,有缺陷文件和无缺陷文件的不平衡可能是导致Logistic回归性能下降的另一个原因。同时,从结果还可以看出Logistic回归和使用Jeffery先验的贝叶斯Logistic回归之间没有显著差异,Jeffery先验是无信息先验,这可能是使用Jeffery先验的贝叶斯Logistic回归的效果不显著的原因之一。同时,还观察到信息先验贝叶斯Logistic回归和无信息先验贝叶斯Logistic回归之间的性能存在显著的差异(问题2)。

在观察到贝叶斯Logistic回归(具有信息先验)的性能明显优于Logistic回归后,下一步是将其性能与其他广泛使用的分类器,如随机森林、朴素贝叶斯、SVM机等(问题3)进行比较。利用Freedman检验这些分类器之间是否存在显著差异,结果如表6所示,随机森林排在第1,Bayes-2先验和Bayes-1先验的贝叶斯Logistic回归分别排名第2和第3。实验结果表明,这些分类器的性能存在显著差异,这一结果与其他比较研究[22]的结果是一致的。

表6 Freedman检验结果(*CD*=2.62)

模型	秩均值
RF	2.127 3
Bayes2	3.053 2
Bayes1	3.301 4
SVM	4.429 6
NB	4.816 7
LR	5.327 6
JP	5.435 9

6 结束语

本文建立了三种不同先验的贝叶斯Logistic回归用于软件缺陷预测。利用6个软件项目测试了贝叶斯方法在Logistic回归中的性能。实验结果表明:具有信息先验的贝叶斯Logistic回归的性能明显优于Logistic回归,贝叶斯Logistic回归的性能在很大程度上依赖于先验的选择,具有信息先验的贝叶斯Logistic回归优于无信息先验的贝叶斯Logistic回归。对多种分类方法的比较研究表明,与其他常用分类方法相比,随机森林排序第一,具有Bayes-2先验的贝叶斯Logistic回归排序第二,具有Bayes-1先验的贝叶斯Logistic回归排序第三。

参考文献:

[1] 陈翔,顾庆,刘望舒,等.静态软件缺陷预测方法研究[J].软件学报,2016,27(1):1-25.

[2] Houwelingen J C V,Cessie S L.Predictive value of statistical models[J].Statistics in Medicine,1990,9(11):23.

[3] Long J S.Regression models for categorical and limited dependent variables[M]//Regression models for categorical and limited dependent variables.[S.l.]:Sage Publications,2014.

[4] Schulz E M,Betebenner D,Ahn M.Hierarchical logistic regression in course placement[J].Journal of Educational Measurement,2004,41(3):271-286.

[5] Genkin A,Lewis D D,Madigan D.Large-scale Bayesian logistic regression for text categorization[J].Technometrics,2007,49(3):291-304.

[6] Halstead M H.Elements of software science(operating and programming systems series)[M].New York:Elsevier Science Inc,1978.

[7] Mccabe T J.A complexity measure[M].[S.l.]:IEEE Press,1976.

[8] Kemerer C F,Chidamber S R.A metrics suite for object oriented design[J].IEEE Transactions on Software Engineering,1994,20(6):476-493.

[9] Zimmermann T,Nagappan N,Zeller A.Predicting bugs from history[M]//Software evolution.Berlin,Heidelberg:Springer,2008.

[10] Menzies T,Dekhtyar A,Distefano J,et al.Problems with precision:a response to comments on 'data mining static code attributes to learn defect predictors'[J].IEEE Transactions on Software Engineering,2007,33(9):637-640.

[11] 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.

[12] Li M,Zhang H,Wu R,et al.Sample-based software defect prediction with active and semi-supervised learning[J].Automated Software Engineering,2012,19(2):201-230.

[13] 何吉元,孟昭鹏,陈翔,等.一种半监督集成跨项目软件缺陷预测方法[J].软件学报,2017,28(6):1455-1473.

[14] Guo L,Ma Y,Cukic B,et al.Robust prediction of fault-proneness by random forests[C]//15th International Symposium on Software Reliability Engineering,2005.

(下转第220页)

[6] Liu W,Anguelov D,Erhan D,et al.SSD:single shot multibox detector[C]//European Conference on Computer Vision,2016:21-37.
[7] Redmon J,Farhadi A.YOLO9000:better,faster,stronger[C]//IEEE Conference on Computer Vision and Pattern Recognition,2017:6517-6525.
[8] Redmon J,Farhadi A.YOLOv3:an incremental improvement[J].IEEE Conference on Computer Vision and Pattern Recognition,2018:89-95.
[9] 姚群力,胡显,雷宏.深度卷积神经网络在目标检测中的研究进展[J].计算机工程与应用,2018,54(17):1-9.
[10] 谢林江,季桂树,彭清,罗恩韬.改进的卷积神经网络在行人检测中的应用[J].计算机科学与探索,2018,12(5):708-718.
[11] 彭清,季桂树,谢林江,等.卷积神经网络在车辆识别中的应用[J].计算机科学与探索,2018,12(2):282-291.
[12] 冯国臣,陈艳艳,陈宁.基于机器视觉的安全帽自动识别技术研究[J].机械设计与制造工程,2015,44(10):39-42.
[13] 胡恬.利用几何分析法和BP神经网络进行人脸识别的研究[J].计算机工程与设计,2002,23(9):18-21.
[14] 刘晓慧,叶西宁.肤色检测和Hu矩在安全帽识别中的应用[J].华东理工大学学报,2014,40(3).
[15] He K,Zhang X,Ren S,et al.Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition,2015:770-778.
[16] Krizhevsky A,Sutskever I,Hinton G E.ImageNet classification with deep convolutional neural networks[C]//International Conference on Neural Information Processing Systems,2012:1097-1105.
[17] Lin T Y,Dollár P,Girshick R,et al.Feature pyramid networks for object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2017:2117-2125.
[18] 张素洁,赵怀慈.最优聚类个数和初始聚类中心点选取算法研究[J].计算机应用研究,2017,34(6):1617-1620.
[19] Liu W,Wen Y,Yu Z,et al.Large-margin softmax loss for convolutional neural networks[C]//International Conference on International Conference on Machine Learning,2016:507-516.

(上接第198页)

[20] Malpica N,de Solórzano C O,Vaquero J J,et al.Applying watershed algorithms to the segmentation of clustered nuclei[J].Cytometry Part A,1997,28(4):289-297.
[21] Kimmel R,Kiryati N,Bruckstein A M.Sub-pixel distance maps and weighted distance transforms[J].Journal of Mathematical Imaging & Vision,1996,6(2/3):223-233.
[22] Zhang D,Lu G.A comparative study on shape retrieval using Fourier descriptors with different shape signatures[C]//International Conference on Intelligent Multimedia and Distance Education,2003:1-9.
[23] Goshtasby A,Mokhtarian F,Mackworth A.Scale-based description and recognition of planar curves and two-dimensional shapes[J].IEEE Trans on Pattern Anal Mach Intell,2009,PAMI-8(1):34-43.
[24] Mokhtarian F,Mackworth A K.A theory of multiscale,curvature-based shape representation for planar curves[J].IEEE Transactions on Pattern Analysis & Machine Intelligence,1992,14(8):789-805.
[25] 刘寅,滕晓龙,刘重庆.复杂背景下基于傅里叶描述子的手势识别[J].计算机仿真,2005,22(12):158-161.
[26] 陈书贞,张俊军.基于曲率尺度空间的指纹特征提取算法[J].电子技术,2008,45(1):101-105.
[27] Han Y,Hara A,Kuzuya A,et al.Automatic recognition of DNA pliers in atomic force microscopy images[J].New Generation Computing,2015,33(3):253-270.
[28] Hu M K.Visual pattern recognition by moment invariants[J].IRE Transactions on Information Theory,1962,8(2):179-187.

(上接第208页)

[15] 陈琳.基于机器学习的软件缺陷预测研究[D].重庆:重庆大学,2016.
[16] Jindal R,Malhotra R,Jain A.Software defect prediction using neural networks[C]//International Conference on Reliability,2015.
[17] Ryu D,Choi O,Baik J.Value-cognitive boosting with a support vector machine for cross-project defect prediction[J].Empirical Software Engineering,2016,21(1):43-71.
[18] Polson N G,Scott J G,Windle J.Bayesian inference for logistic models using polya-gamma latent variables[J].Journal of the American Statistical Association,2012,108(504).
[19] Heinze G,Ploner M.logistf:Firth's bias-reduced logistic regression[EB/OL].[2019].https://rdrr.io/cran/logistf/man/logistf.html#heading-3.
[20] Meyer D,Dimitriadou E,Hornik K,et al.Misc functions of the department of statistics[J].Probability Theory Group,2015.23(4):189-205.
[21] Friedman J,Hastie T,Tibshirani R,et al.Regularization paths for generalized linear models via coordinate descent[J].Journal of Statistical Software,2010,33(1):1-22.
[22] Dejaeger K,Verbraken T,Baesens B,et al.Toward comprehensible software fault prediction models using Bayesian network classifiers[J].IEEE Transactions on Software Engineering,2013,39(2):237-257.