改进PSO-ISVM算法的软件缺陷预测

张 飞
ZHANG Fei

黄淮学院 信息工程学院,河南 驻马店 463000
School of Information Engineering, Huanghuai University, Zhumadian, Henan 463000, China

ZHANG Fei. Software defect prediction based on improved PSO-ISVM algorithm. Computer Engineering and Applications, 2016, 52(11):17-21.

Abstract: In order to improve the prediction accuracy of software defects of support vector machine, this paper proposes a software defect prediction model based on improved support vector machine optimized by particle swarm optimization algorithm. The cost penalty coefficient is introduced to define the fitness function for PSO algorithm, and the fitness function is minimized to eliminate redundant information, to improve the software defects prediction accuracy, to find the optimal parameters of support vector machine. The validity of model is verified with data set. The simulation results show that the proposed model compared with other common defect prediction methods has improved the software defects prediction accuracy and has good nonlinear prediction ability.

Key words: defect prediction; measure and control software; particle swarm optimization; support vector machine

摘 要: 提出基于改进的粒子群优化支持向量机方法(PSO-ISVM)的测控软件缺陷预测方法。通过引入代价惩罚系数,定义粒子群优化算法中的适应度函数,利用最小化适应度函数值作为优化目标,排除大量的冗余干扰信息,提高对测控软件有缺陷模块的预测准确度,寻找支持向量机的最优参数。通过仿真实例分析测控软件有效性,并与常用缺陷预测方法进行比较,表明该模型能加快软件缺陷预测速度和提高对有缺陷模块的预测准确度。

关键词: 缺陷预测; 测控软件; 粒子群优化; 支持向量机
文献标志码: A **中图分类号:** TP391 doi:10.3778/j.issn.1002-8331.1411-0356

测控软件是测控系统的重要组成部分,其质量与可靠性直接影响测控任务的成败。软件测试是保证软件质量的一个关键环节,在软件生命周期早期开展软件缺陷预测可以帮助软件测试人员合理分配测试资源,有效发现软件产品中的缺陷,提高软件质量[1]。

近年来,机器学习、数据挖掘类方法已广泛应用于软件工程领域[2-3],基于该类技术的一系列软件缺陷预测模型[4-7]相继出现,并取得了较好的分类效果。针对传统方法应用范围的局限性,研究人员对各缺陷预测模型提出了相应的改进算法,如文献[8]采用分类准确率作为性能评估指标,利用蚁群算法对支持向量机的参数寻优,建立了基于ACO-SVM的软件缺陷预测模型,文献[9]提出了集成的k-NN软件缺陷预测方法,均在预测精确度上较传统方法有所改善。然而大部分模型虽获得很高的预测精确度(Accuracy),但由于软件缺陷数据集的不平衡性,预测结果往往严重地偏向缺陷数据的多数类,忽略了有缺陷模块上的预测准确度。

对测控软件进行缺陷预测,目的是希望尽可能地发现有缺陷模块。为了提高对有缺陷模块的预测准确度,本文提出了基于改进的PSO-SVM测控软件缺陷预测方法。即通过引入代价惩罚系数,定义粒子群优化算法中的适应度函数,利用最小化适应度函数值作为优化目标,优化支持向量机参数,获得支持向量机模型。通过测控软件实例仿真分析,表明该模型能提高对有缺陷模

基金项目: 河南省科技厅发展计划(No.142102110088);河南省科技攻关项目(No.122102210430)。
作者简介: 张飞(1974—),男,副教授,研究领域:计算机应用,E-mail:azhu_rose@163.com。
收稿日期: 2014-11-28 **修回日期:** 2015-04-17 **文章编号:** 1002-8331(2016)11-0017-05
CNKI**网络优先出版:** 2015-06-05, http://www.cnki.net/kcms/detail/11.2127.TP.20150605.1103.018.html

块的预测准确度。

1　相关工作

1.1　软件缺陷预测

软件缺陷预测是在软件生命周期早期,通过与缺陷的发生具有相关性的软件度量,预测软件模块中缺陷数量或者缺陷分布情况的技术。根据预测结果或者所采用的预测技术,软件缺陷预测可分为两类:

(1)分类技术。该预测技术得到的预测结果是将模块分为有缺陷模块和无缺陷模块。

(2)回归技术。利用回归技术进行预测得到的结果是软件各个模块中所包含的具体缺陷数。

软件缺陷预测的一般过程如图1所示。

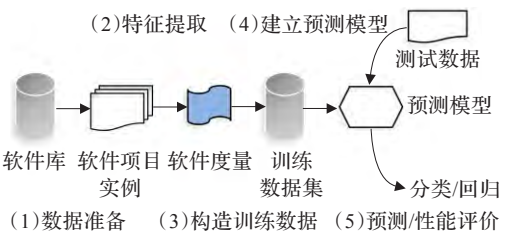

图1　软件缺陷预测的一般过程

1.2　支持向量机

支持向量机(SVM)[10]是一种监督学习算法,它是通过核函数将原始低维不可分的输入数据映射到高维特征空间上,从而达到分类的目的。目前已经广泛应用于机器学习、图像识别、生物序列分析等领域中。支持向量机本质上是如式(1)所示的二次规划求解问题。

$$\min_{w,b}\ \{\frac{1}{2}\|\boldsymbol{w}\|^2\}$$
$$\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}x_i+b)\geqslant 1,\ i=1,2,\cdots,n \tag{1}$$

方程的解为 $\partial=(\partial_1,\partial_2,\cdots,\partial_n)^{\mathrm{T}}$,相应的决策函数为:

$$f(x)=\text{sign}(\sum_{i=1}^{n}\partial_i y_i x_i x+b) \tag{2}$$

1.3　粒子群算法

粒子群算法(PSO)[11]是模拟鸟群捕食行为提出的一种全局搜索算法,较遗传算法和蚁群算法来说,其原理和操作简单,算法较容易实现,运行效率高,已成功运用在很多优化问题上。

粒子群优化算法的主要思想是通过更新粒子的速度和位置,从而不断靠近最优解。假设在一个 D 维的搜索空间中,有 N 个粒子的一个粒子群,则可以用一个 D 维向量 $x_i=(x_{i1},x_{i2},\cdots,x_{id},\cdots,x_{iD})$ 表示每个粒子,每次迭代中粒子移动的速度改变每个粒子搜索的方向和距离,速度用 $v_i=(v_{i1},v_{i2},\cdots,v_{id},\cdots,v_{iD})$ 来表示,通过计算适应值判断粒子位置的优劣。速度和位置更新公式为:

$$v_{id}=w\times v_{id}+c_1 rand(t)(p_{id}-x_{id})+c_2 rand(t)(g-x_{id})$$
$$x_{id}=x_{id}+v_{id} \tag{3}$$

其中,w 为惯性权重,p_{id} 为个体最优,g 为全局最优,c_1 和 c_2 是常数,表示加速系数,它令每个粒子加速向 p_{id} 和 g 移动;$w\times v_{id}$ 表示粒子的先前速度,为粒子的认知能力;$c_2 rand(t)(g-x_{id})$ 表示粒子间的全局信息;$rand(t)$ 是 [0,1) 上均匀分布的随机数。

2　软件缺陷预测模型的建立

目前针对软件缺陷预测的研究较少考虑到数据的类不平衡问题[12]。而对于机器学习方法来说,当训练数据是类不平衡时,建立一个准确有效的预测模型往往非常困难,这是致使预测性能降低的一个主要因素[13-14]。做如下假设:如果在一个训练样本中,98%属于多数类,2%属于少数类,那么利用这个训练样本进行训练得到预测模型,得到的预测结果往往是将所有的样本分类为多数类,也就是说该预测模型的整体分类准确度可达98%。虽然分类准确率非常高,但是在测控软件领域,错误预测一个有缺陷模块比错误地预测一个无缺陷模块要付出更大代价,开发和测试人员希望预测模型能更准确地预测出存在缺陷的模块。因此,本文为了提高对有缺陷模块的预测准确度,提出基于改进的粒子群优化支持向量机的缺陷预测方法。

对于非线性可分问题,通过满足 $\phi(x_i)\bullet\phi(x_j)=k(x_i,x_j)$ 的非线性变换可将非线性分类问题转化到高维空间的线性问题,因此最优化问题变成:

$$\min_{w,b}\ \{\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{n}\xi_i\}$$
$$\text{s.t. } y_i(\boldsymbol{w}\bullet\phi(x_i)+b)\geqslant 1-\xi_i,\ i=1,2,\cdots,n,\ \xi_i\geqslant 0 \tag{4}$$

其中,$\xi_i\ (i=1,2,\cdots,n)$ 表示的是分类误差。C 为惩罚系数。对于 C 的意义可以作如下分析:如果某一点是属于某一类别,但是对它的预测结果是偏离了该类别,那么 C 越大就表示越不想放弃这个点。因此,调节参数 C 对改变支持向量机性能具有关键的影响作用。另外,上式中,核函数 $k(x_i,x_j)$ 中的参数也是需要确定的。这里采用常用的径向基函数:

$$k(x,y)=\exp(-\|x-y\|^2/(2\sigma^2)) \tag{5}$$

作为核函数。本文在利用粒子群优化算法进行参数选择时采用自定义适应度函数的策略以达到一类错误率和二类错误率可调的目的,定义适应度函数为:

$$fitness=Cost_1\times err_1+Cost_2\times err_2 \tag{6}$$

其中,err_1 为一类错误率,$Cost_1$ 为第一类错误所造成的损失系数;err_2 为二类错误率,$Cost_2$ 为第一类错误所造成的损失系数。通过调整 $Cost_1$ 和 $Cost_2$,以寻找最小的适应度函数值为目标,获得最优的支持向量机模型。具体算法流程如图2所示。

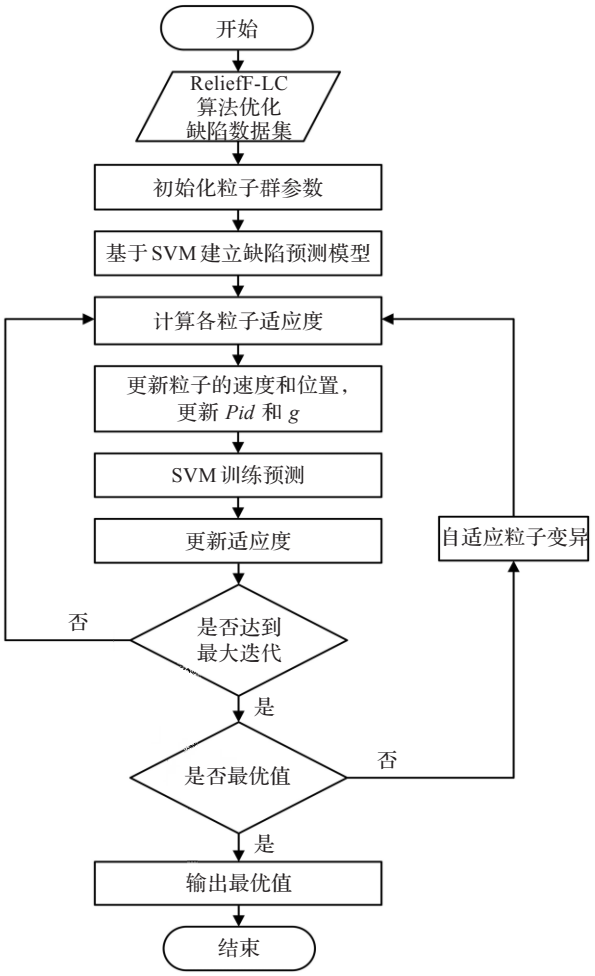

图2　基于改进的PSO-SVM缺陷预测方法流程图

3　数据仿真研究

3.1　测控软件数据源

为了验证PSO-ISVM的软件缺陷预测性能,数据来源是中国河南航空航天局,对其中的工程项目中测控软件的过程文件、测试数据以及源代码进行了收集、整理和归纳,作为本文进行基于改进的PSO-ISVM测控软件缺陷预测研究的基础。共有数据收发类、控制类和数据处理类等3个软件类型的被测件,分别是SJAF、SJGF、SJSF、FWLO、FWMO、FWOO、CLCR、CLTR。表1列出了各软件模块数,软件度量数以及缺陷率等详细数据。

表1　测控软件仿真数据源描述

项目	模块数	软件度量数		缺陷率/%
		代码度量	过程度量	
SJAF	341	28	20	9.8
SJGF	151	28	20	12.2
SJSF	1 096	28	20	16.5
FWLO	102	17	20	3.2
FWMO	696	17	20	11.2
FWOO	486	17	20	12.9
CLCR	1 129	28	20	22.1
CLTR	352	28	20	9.5

3.2　对比模型设计和评价指标

为了观察预测性能,这里与目前常用的支持向量机、决策树以及朴素贝叶斯方法的预测结果进行对比分析。这些模型所取参数情况如表2所示。需要进行说明的是,前文所述的缺陷预测方法中,适应度函数:

$$fitness = Cost_1 \times err_1 + Cost_2 \times err_2 \tag{7}$$

中,令 $Cost_1 = 1$, $Cost_2 = \frac{N_{nd}}{N_d}$,其中 N_{nd} 为无缺陷模块数, N_d 为有缺陷模块数。

表2　模型参数说明

缺陷预测模型	参数说明
J48	$C=0.25, M=2$
SVM	核函数为高斯径向基函数 $C=1, g=0.01$
改进的PSO-SVM	PSO的参数为: 初始化 $c1=1.5, c1=1.7, maxgen=200$ $sizepop=20, k=0.6, wV=1, wP=1, v=3$ $popcmax=100$(SVM参数 c 的最大值) $popcmin=0.1$(SVM参数 c 的最小值) $popgmax=1\,000$(SVM参数 g 的最大值) $popgmin=0.01$(SVM参数 g 的最小值)

为了做到性能评价的更为客观性和可信性,本文采用100轮5折交叉验证的方法进行模型的建立和测试验证:将原始的软件缺陷数据集随机平均分成5组,选择其中四组作为训练数据集训练模型,用训练好的模型对剩下的一组数据进行预测。如此进行100轮。因此,100轮结束后可以得到精确度、I类错误率与II类错误率的500组预测结果。计算500次结果的平均值作为性能评价指标。采用这种方法的原因是:

(1)由于缺陷数据集存在类不平衡的特点,有缺陷的模块只占有极小比例,如果采用10折交叉验证的方法,那么剩下一组作为测试的数据集中可能只存在几个有缺陷模块,甚至不存在有缺陷模块。因此无法得到有效的在有缺陷模块上的预测准确度。

(2)进行100轮的5折交叉验证,是希望每一个软件模块的数据均有机会作为训练数据和测试数据,以得到更加客观的结果。

算法1　仿真实验算法过程

Experiments procedure:

Input:the datasets:D={SJAF、SJGF、SJSF、FWLO、FWMO、FWOO、CLCR、CLTR}

Input:all algorithms:A={SVM, naïve bayes, J48, improved PSO-SVM}

Software metrics preprocess:

Using ReliefF-LC algorithm to select the most important software code metrics

5-fold cross validation:

for i=1:100

```
        for each dataset in D do
          for each algorithm in A do
            perform 5-fold cross validation
          end for
        end for
      end for
Output:
      A、accuracy of 4 classifiers on 8 datasets
      B、the I error rate and the II error rate of 4 classifiers on 8 datasets
```

3.3 性能评价

本文关注对有缺陷模块的预测准确度,因此采用I类错误率和II类错误率作为性能评价指标。混淆矩阵如表3所示。

表3 混淆矩阵

实际分类	预测分类	
	$Class=0$	$Class=1$
$Class=0$	f_{00}	f_{01}
$Class=1$	f_{10}	f_{11}

精确度(*Accuracy*):所有正确被预测的模块数与所有模块数的比例。表示为:

$$Accuracy=\frac{f_{00}+f_{11}}{f_{00}+f_{01}+f_{10}+f_{11}} \tag{8}$$

I类错误率和II类错误率分别定义为:

$$error_{\mathrm{I}}=\frac{f_{01}}{f_{00}+f_{01}} \tag{9}$$

$$error_{\mathrm{II}}=\frac{f_{10}}{f_{10}+f_{11}} \tag{10}$$

从I类错误率和II类错误率的计算公式可知,第一类错误是把不包含缺陷的模块误分类成包含缺陷的模块;第二类错误是把包含缺陷的模块误分类成不包含缺陷的模块。若I类错误率较高,会浪费测试资源在一些质量较高的软件模块上。若II类错误率较高,则会在测试阶段忽视一些高风险模块,因而导致对软件质量的估计过于乐观。

3.4 结果分析

一般的支持向量机、朴素贝叶斯方法、决策树方法和改进的粒子群优化支持向量机方法对8个数据集测试集进行缺陷预测得到表4所示的结果。

(1)利用一般的支持向量机进行分类预测时,I类错误率都很低,即对无缺陷模块的预测结果基本都能接近于全部准确。而II类错误率普遍很高,在SJSF、FWOO、CLCR、CLTR四个软件项目中更是高达100%。

(2)朴素贝叶斯方法和决策树方法的预测结果在II类缺陷率上虽优于支持向量机,但是都以较高的I类错误率作为代价,在全部的八个软件项目中,I类错误率都比支持向量机方法的预测结果有所提高。另外从全局预测准确度来看,在SJAF、SJGF、FWLO、CLTR等几个小样本的软件项目数据上,朴素贝叶斯方法和决策树方法在降低了II类错误率的同时,全局的准确度也有所下降。然而,从实验结果看本文所提出的改进的PSO-SVM不仅在一定程度上降低了II类错误率,也并没有损失全局的预测准确度,反而在大多数的样本上I类错误率也有下降的趋势。

表4 缺陷预测结果比较 %

软件项目	预测模型	全局分类准确率	I类错误率	II类错误率
SJAF	SVM	91.79	1.61	78.79
	NAÏVE BAYES	89.74	2.89	57.58
	J48	86.80	9.65	54.55
	改进的PSO-SVM	92.96	4.50	30.30
SJGF	SVM	86.75	3.76	83.33
	NAÏVE BAYES	66.23	29.32	66.67
	J48	76.16	19.55	55.56
	改进的PSO-SVM	75.50	21.05	50.00
SJSF	SVM	83.48	0	100.00
	NAÏVE BAYES	82.94	6.67	69.61
	J48	86.13	1.20	77.90
	改进的PSO-SVM	88.78	2.84	53.59
FWLO	SVM	87.25	8.33	83.33
	NAÏVE BAYES	88.24	8.33	66.67
	J48	81.37	16.67	50.00
	改进的PSO-SVM	80.39	18.75	33.33
FWMO	SVM	91.95	0.10	98.21
	NAÏVE BAYES	88.51	6.72	66.07
	J48	91.52	2.19	80.36
	改进的PSO-SVM	94.54	1.72	48.21
FWOO	SVM	90.33	0	100.00
	NAÏVE BAYES	84.36	10.02	68.09
	J48	89.51	2.28	87.23
	改进PSO-SVM	91.56	3.87	51.06
CLCR	SVM	81.31	0	100.00
	NAÏVE BAYES	82.99	6.75	61.61
	J48	85.47	5.99	51.66
	改进的PSO-SVM	90.52	4.68	30.33
CLTR	SVM	89.20	0	100.00
	NAÏVE BAYES	82.39	13.69	50.00
	J48	88.92	2.23	84.21
	改进的PSO-SVM	92.05	3.50	44.74

表4中对八个测控软件项目进行缺陷预测的实验结果,将改进的PSO-ISVM方法得到的II类错误率与其他三个方法中性能最好的进行比较,如表5所示。

表5 改进的PSO-ISVM对II类错误率改善效果

项目名称	改善幅度/%	比较对象
SJAF	44.45	J48
SJGF	10.07	J48
SFSF	23.01	Naïve Bayes
FWLO	33.34	J48
FWMO	27.03	Naïve Bayes
FWOO	25.01	Naïve Bayes
CLCR	41.30	J48
CLTR	10.52	J48

通过以上的比较分析,可以得到以下结论:

(1)本文所提出的改进的PSO-ISVM缺陷预测方法在八个测控软件实际项目数据集进行预测时都得到了有效性验证。较其他三种常用缺陷预测方法来说,II类错误率都得到了改善,同时全局的预测准确度都有所提高。与其他三个方法中表现最好的进行比较,II类错误率平均下降了26.84%。

(2)原始样本的大小与缺陷率对预测结果的改善有一定的影响。一般的,样本小、有缺陷模块所占比例小的样本上预测性能上稍差。但是较其他常用缺陷预测方法来说,本文所提出的改进的PSO-ISVM缺陷预测方法在小样本上表现也优异。这也是它的优点所在,能够较好地解决训练数据样本过小的问题。

需要进行说明的是在上述进行实例分析的测控软件缺陷预测模型中,所用核函数均为高斯径向基核函数。由于核函数的参数属于支持向量机模型中的未知参数,因此核函数不同,在本文的模型中需要确定的参数也是不同的。为了观察核函数的影响,分别采用线性核函数和多项式核函数代替高斯径向基核函数。仿真结果显示了采用线性核函数和多项式函数代替高斯径向基核函数结果基本是一致的,因此核函数的选择对本文的测控软件缺陷预测模型的预测性能影响不大。

4　结束语

对测控软件开展软件缺陷预测技术的研究,充分利用历史缺陷数据,在软件生命周期早期阶段即可预测缺陷发生可能性较高的软件模块,一方面使测试资源可以得到优化配置,另一方面可以使开发人员和测试人员在早期就能注意到这些高缺陷率模块,尽早发现和修复缺陷。本文针对现有研究只关注整体分类准确率而忽略了II类错误率过高的问题,提出了基于改进的PSO-ISVM测控软件缺陷预测方法。该方法通过在粒子群优化算法的适应度函数中引入损失惩罚系数,优化支持向量机参数,从而达到提高对有缺陷软件模块的预测准确率。

参考文献:

[1] 李心科,金元杰.基于灰色预测理论的软件缺陷预测模型研究[J].计算机应用与软件,2009,26(3):101-103.

[2] Halkidi M, Spinellis D, Tsatsaronis G, et al.Data mining in software engineering[J].Intelligent Data Analysis, 2011, 15:413-441.

[3] Xie T, Thummalapenta S, Lo D, et al.Data mining for software engineering[J].IEEE Computer, 2009, 42:55-62.

[4] Kalsi M, Singh J.A hybrid approach of module sequence generation using neural network for software architecture[J]. International Journal of Science and Research, 2013, 2(5):133-137.

[5] 尹然,丁晓明,李小亮,等.基于SA-BP神经网络的软件缺陷预测模型的研究[J].西南师范大学学报,2013,38(3):147-152.

[6] Singh Y, Kaur A, Malhotra R.Software fault proneness prediction using support vector machines[C]//Proceedings of the World Congress on Engineering, 2009:1-6.

[7] Varade S M, Ingle M D.Overview of software fault prediction using clustering approaches and tree data structure[J].The International Journal of Engineering and Science, 2012, 1:239-242.

[8] 姜慧研,宗茂,刘相莹.基于ACO-SVM的软件缺陷预测模型的研究[J].计算机学报,2011,34(6):1148-1154.

[9] 何亮,宋擒豹,沈钧毅.基于Boosting 的集成k-NN软件预测方法[J].模式识别与人工智能,2012,25(5):792-802.

[10] Cortes C, Vapnik V.Support-vector networks[J].Machine Learning, 1995, 20(3):273-297.

[11] 郭平.软件可靠性工程中的计算智能方法[M].北京:科学出版社,2012.

[12] Wang Shuo, Yao Xin.Using class imbalance learning for software defect prediction[J].IEEE Trans on Reliability, 2013, 62(2):434-443.

[13] Hall T, Beecham S, Bowes D, et al.A systematic review of fault prediction performance in software engineering[J]. IEEE Trans on Software Engineer, 2012, 38(6):1276-1304.

[14] Arisholm E, Briand L C, Johannessen E B.A systematic and comprehensive investigation of methods to build and evaluate fault prediction models[J].J Syst Software, 2010, 83(1):2-17.