基于改进蝙蝠算法的软件缺陷预测模型

杨晓琴
(太原广播电视大学,山西 太原 030002)

摘 要: 软件缺陷预测模型因为软件规模持续扩大以及安全性要求越来越高,变得越来越重要。支持向量机(SVM)模型突出优点是它具有较强的非线性分类能力,所以在软件缺陷预测应用非常广泛。但是,SVM 模型缺乏有效的方法来确定最佳参数,以至于不能达到理想的准确度。所以,提高 SVM 模型的参数,提高 SVM 模型的软件缺陷预测能力成为了研究热点。蝙蝠算法是一种启发式搜索算法,它模型简单,易于实现,但是却易陷入局部最优,因此采用加入莱维飞行的蝙蝠算法对 SVM 模型的参数选择进行优化。为了测试这个新模型的性能,仿真实验使用了一些软件缺陷预测的公共数据集,然后将结果与传统的启发式算法进行比较。实验结果表明,LBA-SVM 模型的分类能力优于其他方法。

关键词: 支持向量机; 软件缺陷预测; 莱维飞行; 蝙蝠算法
中图分类号: TP391 **文献标识码:** A **文章编号:** 1673-629X(2018)12-0074-05
doi: 10.3969/j.issn.1673-629X.2018.12.016

Software Defect Prediction Model Based on Improved Bat Algorithm

YANG Xiao-qin
(Taiyuan Radio and TV University, Taiyuan 030002, China)

Abstract: The software defect prediction model is becoming more and more important because of its continuous expansion of software scale and the increasing security requirements. The protruding advantage of support vector machine (SVM) model is that it has strong nonlinear classification, so it is widely used in software defect prediction. However, the SVM model lacks effective methods to determine the optimal parameters, so that it cannot achieve the ideal accuracy. Therefore, to improve the parameters and the ability of software defect prediction in SVM model has become a hot research topic. Bat algorithm is a heuristic search algorithm, which is simple and easy to implement, but it is easy to fall into local optimal. Therefore we use the bat algorithm with Levy flight to optimize parameters on the SVM model. In order to test the performance of this new model, a number of common data sets are used to predict software defects. The simulation results are compared with the other five methods, which show that the classification of the LBA-SVM model is better than that of other methods.

Key words: support vector machine; software defect prediction; Levy flight; bat algorithm

0 引 言

软件缺陷预测是一种根据以往开发软件模块特征来预测未来开发软件缺陷性的模型。软件缺陷预测能够预测软件存在的缺陷位置和数量,能够知道测试工作,提升测试工作效率,节约成本。

软件缺陷预测技术涵盖了动态、静态两种。以往的动态缺陷预测技术主要有 Rayleigh 分布模型、指数分布模型和 S 曲线分布模型。Rayleigh 分布模型应用广泛,Trachtenberg 等开展了一系列实验,进而实现对缺陷数据的验证,从结果来看这些缺陷数据符合 Rayleigh 分布模型。Rescorla 等在一系列研究之后,阐述了著名的指数分布模型,其将研究重点放在软件验收的测试阶段,通过深入研究,确定了此阶段软件缺陷累计分布规律[1]。基于 S 曲线的分布模型能够了解到,从软件开发一直到回归测试,在缺陷累计数量方面是符合 S 型的,最终意味着来到了软件成熟阶段。Alhazmi 等在一系列研究之后,阐述了著名的 AML 模型,该模式作为 S 曲线模型中的代表模型,和实验数据的拟合度较高,性能较好[2-3]。Joh 等在对软件生命周期中软件缺陷模块占比变动规律进行描述的过程中,

收稿日期: 2018-01-26 **修回日期:** 2018-05-30 **网络出版时间:** 2018-07-04
基金项目: 国家自然科学基金(青年科学基金)(61403272);山西省重点研发计划(工业部分)项目(201703D121042-1);山西广播电视大学课题研究成果(SXYJ201610)
作者简介: 杨晓琴(1983-),女,讲师,硕士,研究方向为群智能优化算法、Web 服务技术等。
网络出版地址: http://kns.cnki.net/kcms/detail/61.1450.TP.20180703.1513.044.html

主要基于 Weibull 统计分布来实现[4]，该方法能够有效地评估软件风险。

与动态软件缺陷预测技术不同，静态软件缺陷预测技术一般将软件预测问题进行转化，最终变为机器学习中的一种分类问题，具体就是基于缺陷预测模型有效划分待预测模块，分为无缺陷以及有缺陷的模块两类。传统分类技术相对丰富一些，如支持向量机(support vector machine，SVM)[5]、贝叶斯网络(Bayesian network，BN)[6]、决策树(decision tree)[7]、神经网络[8]以及 K 最近邻分类算法[9]等。

支持向量机的两个关键参数的选择非常重要，会影响到支持向量机进行分类的最终效果。由于基于支持向量机软件缺陷预测模型有着比较低的准确率，笔者为了解决这个问题，基于改进蝙蝠算法来完善支持向量机的软件缺陷预测模型。改进蝙蝠算法能够优化支持向量机的关键参数，能获得更好的分类模型，提高模型的分类准确性。为了验证该预测模型的预测性能，利用 NASA 的 MDP 数据集进行测试，并与已有的缺陷预测模型进行比较。

1　相关工作

1.1　软件缺陷预测模型

静态缺陷预测技术可以根据历史数据信息有效预测模块缺陷倾向性，不过专家将负责抽取标记软件模块，同时软件缺陷预测的性能也受度量元设计的影响。虽然动态的软件缺陷预测技术能够掌控软件生命周期中所有缺陷的变动状况，但是特征提取难度比较高，有着比较窄的覆盖面，运行环境对其影响较大，文中主要对静态软件缺陷预测技术进行研究。

模型构建包含三个步骤:

(1) 选取软件模块，全方位分析软件代码，对全部的属性值进行统计，有效标记模块所具有的缺陷性，这样就可以获得相对应的训练数据集。在软件模块中，有效挑选 n 个样本数据集为 $D_n = \{(x_1,y_1),\cdots,(x_n,y_n)\}$，其中软件模块 i 的度量属性是向量 $x_i \in R^k$，所有向量均包含多维的度量元，并表示为 $x_i = (a_1,\cdots,a_k)$。$y_i \in Y$ 是第 i 个模块的类标记。专家将负责抽取标记软件模块数据集。

(2) 通过所选择预测方法或技术对数据集进行训练，最终将获得所需的训练样本。对样本的数据集进行学习，能够在软件模块的属性以及缺陷性间实现识别标准的有效构建，将多个模块属性向量进行输入，这样就可以获得特定输出，有助于全方位了解模块所具有的缺陷性，得出软件模块属性和缺陷相关的映射关系。

(3) 对软件缺陷预测技术、方法性能进行验证。首先需要取得预测所获得的结果，有效地评估技术或方法所具有的性能。相关指标主要有准确率、误报率、AUC 取值和缺陷检出率等。不同的软件或系统对于成本和安全性的要求不同，对于不同指标的要求不一样，但是目的都是尽可能地节约软件的测试成本，检测出尽可能多的软件缺陷。

单目标启发式搜索算法在软件缺陷预测中有着较多的应用，对软件缺陷预测技术进行不断完善，这样软件预测将达到更高的准确率。分析启发式搜索算法可以了解到，其表现为搜索速度快，搜索精度高，能够为 NP 难的问题搜索到满意解，可以对缺陷预测技术的参数进行选择，搜索出符合要求的参数值。

支持向量机在学习完训练样本之后，能够找出最优分类面和支持向量，并且对训练的数据集的有效类型进行划分，这样分类面尽可能远离于每一类的距离。若数据表现出线性不可分，依托核函数映射到高维的特征空间，这样数据将获得最优的分类面。支持向量机在泛化能力方面表现十分出色，同时可以有效解决非线性分类问题。Elish 等横向对比了支持向量机以及其他的分类算法，由实验结果了解到，支持向量机表现出了更好的性能[10]。

启发式搜索算法在一定程度上可以优化支持向量机，尤其是惩罚因子以及径向基带宽等层面，主要是对其取值进行优化。王男帅等[11]使用遗传算法做最佳度量属性的选择，避免有用的信息被过早筛除。姜慧研等[12]针对数据集，基于主成分分析法有效降维处理数据集，这样模型将有更高的运算速度，之后再利用蚁群优化算法对支持向量机的参数进行选择。

1.2　支持向量机

1995 年，Cortes 和 Vapnik 第一次阐述了 SVM。后者对非线性分类问题能够进行有效的处理。支持向量机的原理是在空间中寻找一个分类面作为两个类的分割，这个分类面距两类样本的距离越大越好，这样可以更好地分开两类样本。支持向量机突出的一点是可以经过核函数映射，在高位空间来处理线性不可分的情况。

支持向量机处理的样本可以表示为 $(x_1,y_1),\cdots,(x_l,y_l)$，$x \in R^n$，$y \in \{+1,-1\}$，其中 l 为样本数量，n 为 x 的维数。支持向量机通过训练寻找到支持向量进而找到一个离两类间隔最大的分类面，作为两类样本的分界，表示为:

$$w^{\mathrm{T}}x + b = 0 \tag{1}$$

其中，w^{T} 表示最优分类面的法向量; $\varphi(x)$ 表示将 x 映射到高维空间的核函数。

数据集中用于测试的样本去掉类别标签，最后的分类结果与原类别得出的结果进行对分类模型的性能

衡量。

当遇到线性不可分的样本集时，可以通过核函数的复杂计算，将这些样本映射到维数更高的特征空间，在新特征空间中进行训练找到分类面。这里使用径向基函数(RBF)，公式如下：

$$k(x,x_i)=\exp\{-\frac{|x-x_i|^2}{\sigma^2}\} \tag{2}$$

参数 σ 用于控制 RBF 的半径。分类函数为：

$$f(x)=\mathrm{sgn}\{\sum_{i=1}^{L} y_i \cdot a_i \cdot k(x_i,x)+b\} \tag{3}$$

其中，$a_i \in [0,C]$，C 叫做惩罚参数。

支持向量机的两个参数 C 和 σ 很大程度上影响了分类性能。然而，传统上是按照经验给定参数值。为了提供适当的参数，提出应用 LBA 来优化支持向量机，该混合算法称为结合 LBA 和 SVM 的混合改进算法(简称 LBA-SVM)。

2 LBA-SVM 的软件缺陷预测模型

2.1 蝙蝠算法

蝙蝠算法[13-14]是基于模拟蝙蝠捕获猎物行为的启发式群智能优化算法，该算法是在受该物种独特的回声定位行为的模拟启发下提出的。标准蝙蝠算法的结构如下：

$$f_i=f_{\min}+(f_{\max}-f_{\min})\beta \tag{4}$$

$$V_i^t=V_i^{t-1}+(X_i^t-X^*) \tag{5}$$

$$X_i^t=X_i^{t-1}+V_i^t \tag{6}$$

其中，$\beta \in [0,1]$ 是一个随机向量，X^* 是当前全局最佳位置。根据所求解的问题类型，在固定 λ_i(或 f_i)的同时改变 f_i(或 λ_i)。在实际求解过程中，可以根据问题的领域大小确定 f_i 的取值，比如使用 $f_{\min}=0$ 和 $f_{\max}=100$。开始时每只蝙蝠随机分配频率，频率是从 $[f_{\min},f_{\max}]$ 平均得出的。局部搜索时，每只蝙蝠的更新公式如下：

$$X_{\mathrm{new}}=X_{\mathrm{old}}+\varepsilon A^t \tag{7}$$

其中，$\varepsilon \in [-1,1]$ 是随机数；$A^t=<A_i^t>$ 是所有蝙蝠在这一代里的平均响度。然后，脉冲发射的响度 A_i 和速率 r_i 也要随着迭代过程进行更新。一旦蝙蝠发现了猎物，响度会逐渐降低，脉冲速率会逐渐提高。脉冲发射的响度 A_i 和速率 r_i 的更新公式如下：

$$\begin{cases} A^t=\alpha A_i^{t-1} \\ r_i^t=r_i^0[1-\mathrm{e}^{-\gamma t}] \end{cases} \tag{8}$$

其中，α 和 γ 是恒量，α 类似于模拟退火算法中冷却进程表中的冷却因素。

初始化时，每只蝙蝠所发出的响度和脉冲速率随机给出。一般情况下，定义初始的响度 A_i^0 通常在[1，2]之间，初始的发射速率 r_i^0 一般在 0 左右。蝙蝠的响度和发射率将随搜索过程不断更新，从而逐渐飞向最优解。

基于以上分析，蝙蝠搜索算法的主要步骤可以描述如下：

(1) 初始化蝙蝠各参数：位置 $X_i(0)$，速度 $V_i(0)$，响度 $A_i(0)$，脉冲发射速率 $r_i(0)$ 以及脉冲频率 $f_i(0)$；

(2) 按照式 5 和式 6 更新蝙蝠个体的速度和位置；

(3) 对于每个个体 i，产生一个介于(0，1)之间的随机数 rand_1，如果 $\mathrm{rand}_1 < r_i(t)$，则按照式 7 更新个体的位置；

(4) 随机产生(0，1)之间的随机数 rand_2，如果 $\mathrm{rand}_2 < A_i(t)$ & $f(x_i(t)) < f(p(t))$，接受新解 $x_i(t+1)$ 和速度 $v_i(t+1)$；然后分别按照式 8 更新 $r_i(t)$ 和 $A_i(t)$；

(5) 更新群体最优位置 $p(t)$；

(6) 如果满足结束条件则终止算法，否则转入步骤 2。

2.2 基于 Levy 飞行的蝙蝠算法

自然界中的动物寻找食物地点的行为具有随机性，包括搜索方向以及步长。动物的一般活动路径也是一个随机的移动过程。后续的行为方向和步长取决于动物当前所处位置(或状态)。Levy 飞行可以描述为一个运动的实体，通常会进行小步长的移动，能够偶尔迈出异常大的步子，从而改变一个体系的行为。它的运动方向是随机的，但是其运动步长是按照幂次率分布的。Levy 飞行的一个特点是小步的移动占多数，但是也会少数时刻选择大步的移动，这能够保证个体搜索的不重复性。

在标准蝙蝠算法中，所有蝙蝠个体的速度更新公式中受到个体当前最优位置的影响，导致蝙蝠个体比较容易陷入局部最优，而且很难从局部极值点跳出。文中在局部搜索中引入 Levy 飞行[15-16]，利用 Levy 飞行随机游走的特性，利用随机步长以有偏方式进行随机移动，使得蝙蝠个体在局部搜索中进行随机游走，从局部最优中跳出，获得更好的解。蝙蝠个体局部搜索进行莱维飞行根据代数决定，到了一定代数进行一次，这样既兼顾了原本的局部搜索方案，又不至于陷入局部极值点。根据经验设置每 10 代在局部搜索中进行一次莱维飞行。

LBA-SVM 软件缺陷预测模型利用加入莱维飞行局部搜索策略的蝙蝠算法优化支持向量机，其流程如图 1 所示。

(1) 样本数据集初始化。将数据集带入模型，对样本进行初始化处理。

(2) 建立基于 SVM 的分类模型。将样本数据集

带入支持向量机,通过十折交叉法进行训练和测试。

(3) 利用 LBA 算法对 SVM 进行优化。初始化 LBA 中的个体位置为随机的 SVM 参数,将 LBA 个体位置带入 SVM,通过 SVM 分类模型,测试参数的好坏,当迭代次数未达到最大迭代次数时,进行个体位置更新,不断优化参数选择。

(4) 结果输出。在代数达到最大迭代次数时,测试和评估模型的性能。

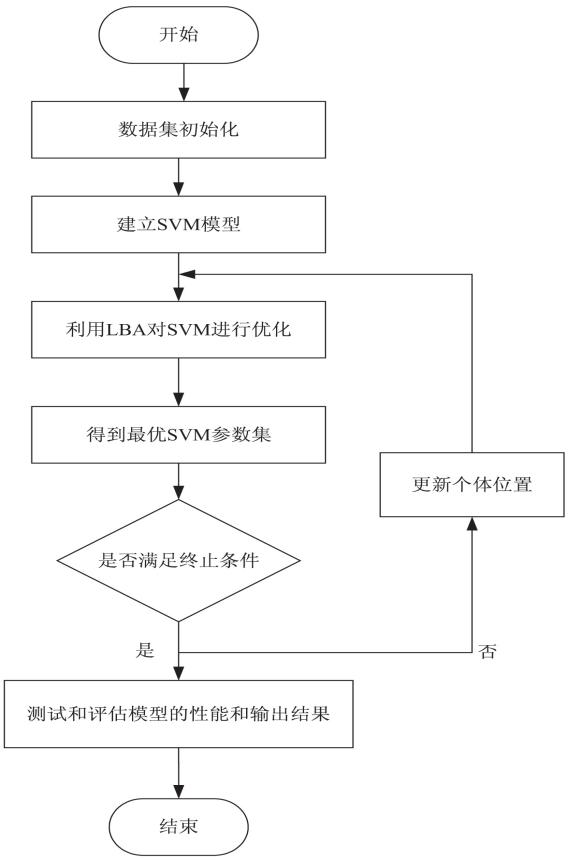

图 1　LBA-SVM 软件缺陷预测模型

3　仿真实验

3.1　数据集及评价标准

该数据为 NASA 提供的 MDP 数据集中的四个数据集: CM1、KC1、PC1、JM1 (http: //mdp. ivv. nasa. gov/) 。数据集信息如表 1 所示。

表 1　数据集

数据集	实例个数	缺陷率/%
CM1	505	9.7
KC1	2 107	15.4
PC1	1 107	6.3
JM1	10 878	19.3

分类问题的软件缺陷预测根据正确预测和错误预测模块数统计并计算相应比例来对模型进行评价,实际操作中,通常使用混淆矩阵对结果进行分析。实际上为缺陷模块,分类结果一致的称为 TP,被误分为非缺陷模块的为 FN; 实际上非缺陷模块被正确分类的总数为 TN,被误分为缺陷模块称为 FP,如表 2 所示。

表 2　混淆矩阵

	预测为有缺陷	预测为无缺陷
真实有缺陷	TP	FN
真实无缺陷	FP	TN

准确度(accuracy) 是最常用的指标,可以理解为预测正确的模块数与总软件模块数的比值,如下:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

3.2　实验参数

为了验证文中提出的软件缺陷预测算法的性能,对其进行了仿真实验,主要基于 MatlabR2013a 平台来开展。使用十折交叉验证方法设计仿真实验。每个数据集平均分 10 份,其中 9 份做训练,剩余 1 份对模型性能进行测试,最后取 10 次预测结果的平均值作为对算法预测能力的评价。

支持向量机两个参数的范围都是(0,1 000) 。蝙蝠算法参数设置如表 3 所示。

表 3　实验参数

种群规模	维度	独立运行次数	脉冲频率	脉冲发射频度	脉冲响度衰减系数	脉冲响度	脉冲频度增强系数
Popsize	D	Run_times	f_i	r_0	α	A_0	γ
50	2	30	(0,5)	0.9	0.9	0.9	0.9

3.3　实验结果

实验结果如表 4 和图 2 所示。

表 4　实验结果

算法	CM1	KC1	PC1	JM1
SVM	86.0	88.0	86.5	85.0
PSO-SVM	91.0	90.0	92.0	92.5
GA-SVM	91.5	89.0	93.5	91.5
BA-SVM	91.0	90.5	90.5	90.5
LBA-SVM	92.0	91.0	93.8	92.7

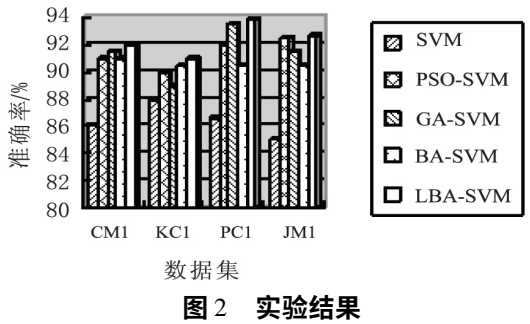

图 2　实验结果

与传统的SVM模型、PSO-SVM模型、GA-SVM模型、BA-SVM模型进行比较，通过表4和图2可以看到，传统SVM的精度无法到达90%以上，而优化过的算法中，准确率都有所提升。说明通过智能算法优化参数，可以有效地提高传统SVM模型的准确率。通过比较发现，在改进的模型中，提出的LBA-SVM的准确度明显优于其他算法，说明莱维飞行策略能够有效地帮助传统BA算法跳出局部最优。利用改进后的算法优化软件缺陷预测模型使得准确度得到了进一步的提升。

4 结束语

支持向量机作为软件缺陷预测模型的核心，具备很好的分类能力。文中从对支持向量机的两个关键参数进行优化出发，以提高软件缺陷预测模型的准确率为目标，改进了标准蝙蝠算法的局部搜索能力，加入了莱维飞行，保证在隔一定代数后能够有一次局部最优点附近进行随机游走，减少了蝙蝠算法陷入局部极值的可能性。通过四个数据集的实验对比表明，LBA-SVM显著提高了准确度。

参考文献:

[1] GOEL A L,OKUMOTO K. A time-dependent error-detection rate model for software reliability and other performance measures [J]. IEEE Transactions on Reliability,1979,28(3):206-211.

[2] ALHAZMI O H,MALAIYA Y K,RAY I. Measuring,analyzing and predicting security vulnerabilities in software systems [J]. Computers & Security,2007,26(3):219-228.

[3] 邵 堃,刘宗田,胡学钢,等. AML:一种面向需求的多Agent建模语言 [J]. 模式识别与人工智能,2007,20(1):131-137.

[4] JOH H C,KIM J,MALAIYA Y K. Vulnerability discovery modeling using weibull distribution [C]//Proceedings of international symposium on software reliability engineering. Seattle,WA,USA:IEEE,2008:299-300.

[5] 郭明玮,赵宇宙,项俊平,等. 基于支持向量机的目标检测算法综述 [J]. 控制与决策,2014,29(2):193-200.

[6] 慕春棣,戴剑彬,叶 俊. 用于数据挖掘的贝叶斯网络 [J]. 软件学报,2000,11(5):660-666.

[7] 杨学兵,张 俊. 决策树算法及其核心技术 [J]. 计算机技术与发展,2007,17(1):43-45.

[8] 焦李成,杨淑媛,刘 芳,等. 神经网络七十年:回顾与展望 [J]. 计算机学报,2016,39(8):1697-1716.

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

[10] 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.

[11] 王男帅,薛静锋,胡昌振,等. 基于遗传优化支持向量机的软件缺陷预测模型 [J]. 中国科技论文,2015,10(2):159-163.

[12] 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.

[13] YANG Xinshe. A new metaheuristic bat-inspired algorithm [C]//Nature inspired cooperative strategies for optimization. [s.l.]: [s.n.],2010:65-74.

[14] MIRJALILI S,MIRJALILI S M,YANG X S. Binary bat algorithm [J]. Neural Computing & Applications,2014,25(3-4):663-681.

[15] 刘长平,叶春明. 具有Lévy飞行特征的蝙蝠算法 [J]. 智能系统学报,2013,8(3):240-246.

[16] 费 腾,张立毅,陈 雷. 混合Levy变异与混沌变异的改进人工鱼群算法 [J]. 计算机工程,2016,42(7):146-152.

(上接第73页)

traffic sign recognition [J]. Neural Networks,2012,32(2):305-323.

[14] GUO Hairu,WANG Xiaojie,ZHONG Yixin,et al. Traffic signs recognition based on visual attention mechanism [J]. Journal of China Universities of Posts & Telecommunications,2011,18(2):12-16.

[15] KHAN J,BHUIYAN S,ADHAMI R. Hierarchical clustering of EMD based interest points for road sign detection [J]. Optics & Laser Technology,2014,57:271-283.

[16] ZAKLOUTA F,STANCIULESCU B. Real-time traffic sign recognition in three stages [J]. Robotics & Autonomous Systems,2014,62(1):16-24.

[17] DESHPANDE A V. Design approach for a novel traffic sign recognition system by using LDA and image segmentation by exploring the color and shape features of an image [J]. International Journal of Engineering Research & Applications,2014,4(11):36-98.