基于模糊支持向量机的软件缺陷预测技术

程元启[1]，姚淑珍[1]，谭火彬[2]，李丹丹[1]

（1. 北京航空航天大学 计算机学院，北京 100191；
2. 北京航空航天大学 软件学院，北京 100191）

摘 要：为克服软件缺陷预测中的类不平衡问题，提出机器学习模型 GA-FSVM。去除软件数据集的冗余特征，使用模糊支持向量机作为分类器，针对软件缺陷预测问题提出相应的模糊隶属度函数，使其能适应数据集的类不平衡，应对数据集中的特异点，使用遗传算法进行参数调优，训练分类器。在 NASA 数据集上进行交叉验证的结果表明，和几种常见的算法相比，该方法能够提高有缺陷样本的 F-measure 值。

关键词：软件缺陷预测；模糊支持向量机；类不平衡问题；遗传算法；机器学习
中图法分类号：TP311.5 **文献标识号**：A **文章编号**：1000-7024（2018）09-2753-05
doi：10.16208/j.issn1000-7024.2018.09.010

Software defect prediction technology based on fuzzy support vector machine

CHENG Yuan-qi[1], YAO Shu-zhen[1], TAN Huo-bin[2], LI Dan-dan[1]

(1. School of Computer Science and Engineering, Beihang University, Beijing 100191, China;
2. School of Software, Beihang University, Beijing 100191, China)

Abstract: To solve the class imbalance problem in software defect prediction, a machine learning model GA-FSVM was proposed. The redundant features of software data sets were removed, and fuzzy support vector machine was used as classifier. In addition, the corresponding fuzzy membership functions for software defect prediction were proposed, which not only adapted to the data set of class imbalance, but also dealt with outliner in data set, and genetic algorithm was used for parameter tuning. The results of cross validation on NASA datasets show that the proposed method can improve the F-measure value of defective samples compared with several common algorithms.

Key words: software defect prediction; fuzzy support vector machine; class imbalance; genetic algorithm; machine learning

0 引 言

软件缺陷预测技术可以根据软件的多个特征属性，来预测软件缺陷的数目和分布。目前已经有很多研究人员借助机器学习方法来预测软件项目中的模块是否有缺陷，不仅诸如朴素贝叶斯、决策树等经典的机器学习方法已经被使用[1]，而且基于 Bagging 和 Boosting 的集成学习的方法也被广泛应用[2]。随着人工智能、统计学等技术的不断发展，基于机器学习的软件缺陷预测技术是热门领域。需要注意到的是，对于软件系统而言，含有缺陷的模块往往只是少数，大多数模块并无缺陷，所以在软件缺陷预测中，无缺陷的程序模块（即多数类）的数量往往比有缺陷程序模块（即少数类）的数量多得多，这就是软件缺陷预测中的类不平衡问题，不平衡数据集会让分类器更多地从多数类中获取有效信息，这会使预测结果更加倾向于多数类，在少数类上分类器的性能表现会变差[3]，这会让分类结果不尽人意。为了更好地解决软件缺陷预测中的类不平衡问题，本文在现有工作的基础上，提出软件缺陷预测方法 GA-FSVM。它采用模糊支持向量机作为分类器，并结合软件缺陷预测问题的特点，合理设定模糊隶属度函数，并且采用遗传算法进行模型的参数选择。相比传统的分类器，在少数类上的性能表现更好。

收稿日期：2017-07-13；**修订日期**：2018-02-24

作者简介：程元启（1993-），男，陕西汉中人，硕士研究生，研究方向为机器学习、软件工程；姚淑珍（1965-），女，北京人，博士，教授，CCF 高级会员，研究方向为软件工程、Petri 网；谭火彬（1979-），男，北京人，博士，讲师，研究方向为软件工程；李丹丹（1986-），女，山东菏泽人，博士研究生，研究方向为处理器设计、机器学习、软件工程技术。

E-mail：chengyuanqi@hotmail.com

1 基于模糊支持向量机的软件缺陷预测技术

1.1 模糊支持向量机

模糊支持向量机算法(fuzzy-SVM，FSVM) 是在传统的支持向量机算法(SVM) 的基础上推广的，因此本节首先介绍传统的支持向量机算法，然后介绍模糊支持向量机算法。

SVM 的目的是寻找一个最优分类超平面，并利用此超平面进行分类。SVM 算法要求该超平面在保证分类精度的同时，能够使超平面两侧的空白区域最大化。

以两类数据分类为例，给定训练样本集 $(x_i,y_i),i=1,2,\cdots,l,x\in R^n,y_i\in\{\pm 1\}$，超平面记为 $(w\cdot x)+b=0$，为了使分类面对所有样本正确分类并且具备分类间隔，则必须满足如下约束条件

$$y_i[(w\cdot x_i+b)]\geqslant 1-\xi_i,i=1,\cdots,l$$

其中，$\xi_i>0$ 称为松弛变量。引入松弛变量的目的在于可以将硬间隔分类转化为软间隔分类。此外，经过数学推导可以计算出分类间隔为 $\frac{2}{\|w\|}$，因此构造最大间隔的问题就转化为在以上约束式的条件下计算该条件最值

$$\min\Phi(w)=\frac{1}{2}(w'\cdot w)+C\sum_{i=0}^{l}\xi_i$$

式中：C 是常数，$C\sum_{i=0}^{l}\xi_i$ 是惩罚项，C 值大时对错误分类的惩罚增大，C 值小时对错误分类的惩罚减小。为了计算该问题，引入对偶算法，将以上问题转换为对偶问题

$$\max_{\alpha}W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i x_j$$

$$\text{s.t.}\sum_{i=1}^{l}y_i\alpha_i=0,0\leqslant\alpha_i\leqslant C,i=1,\cdots,l$$

其中，$\alpha_i>0$ 为拉格朗日乘子。通过上式计算出 α^* 后，便可以进一步计算出超平面 $(w\cdot x)+b=0$ 中的 w，b。

但是需要值得说明的是，以上方法只限制于线性可分的情况。对于非线性可分的情况，需要引入核函数进行处理。此时目标函数可以改写为

$$\max_{\alpha}W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i\cdot x_j)$$

$$\text{s.t.}\sum_{i=1}^{l}y_i\alpha_i=0,0\leqslant\alpha_i\leqslant C,i=1,\cdots,l$$

式中：$K(x_i,x_j)$ 为核函数。然而，核函数的最优选择目前还没有一个明确的标准。目前核函数有多个种类：多项式函数、径向基函数、Sigmoid 函数等等。

然而在实际应用中，SVM 的缺点在于并没有对特异点(outliner) 进行处理。特异点是偏离正常位置很远的数据点，由于 SVM 的特性，特异点会对 SVM 的性能造成一定的影响。同时，在二分类问题中，有时我们只想关注其中一个类别的准确度，这是传统的 SVM 算法无法做到的。因此 Chun-Fu Lin 和 Sheng-De Wang 提出了模糊支持向量机算法[4]。

在模糊支持向量机中，给每个数据点增加模糊隶属度(fuzzy membership value)，新的数据点的形式为 $(y_1,x_1,s_1),\cdots,(y_l,x_l,s_l)$，$i=1,2,\cdots,l$，$x_i$ 和 y_i 的定义与之前相同，s_i 是模糊隶属度，且 $\sigma\leqslant s_i\leqslant 1$，$\sigma$ 大于 0 且足够小。

和传统的 SVM 类似，最佳超平面的计算等价于求解以下条件最值问题

$$\min\frac{1}{2}(w'\cdot w)+C\sum_{i=0}^{l}s_i\xi_i$$

$$\text{s.t.}\ y_i[(w\cdot x_i+b)]\geqslant 1-\xi_i,\ \xi_i\geqslant 0,i=1,2,\cdots l$$

可以看到，s_i 直接作用于惩罚系数 ξ_i，因此 $s_i\xi_i$ 是带有权值的惩罚系数。更小的 s_i 可以减少参数 ξ_i 的影响，使得对应的数据点 x_i 的权值降低。类似于传统的 SVM 算法，上式的求解可以转换为求解对偶问题，进而计算出 w，b。

1.2 模糊隶属度函数

根据 1.1 中的论述，模糊隶属度 s_i 的功能就是对不同的数据点赋予不同的权值，对少数类、非特异点应该被赋予更高的模糊隶属度，而对于多数类、噪声数据和特异点，应该被赋予较低的模糊隶属度[5]。由此可见，模糊支持向量机的关键是模糊隶属度的确定。在软件缺陷预测技术中，模糊隶属度需要同时考虑以下两个问题，第一是特异点的影响，第二是数据的不平衡性。在本文中，模糊隶属度由模糊隶属度函数(fuzzy-membership functions) 定义，该函数可以对不同的数据点计算出相应的模糊隶属度。在软件缺陷预测中，含有缺陷的模块是正类，也是少数类，而不含有缺陷的模块是负类，是多数类。如前文所述，特异点的存在和数据的不平衡性都会导致传统的机器学习算法效果比较差。如何合理地调整模糊支持向量机中的模糊隶属度值得研究。

我们定义如下的隶属度函数[6]，假设训练集共有 N 个数据样本，$1\leqslant i\leqslant N$，设 s_i^+ 和 s_i^- 分别是正(少数) 类和负(多数) 类的模糊隶属度函数，其数学形式为

$$s_i^+=f(x_i^+)r^+$$

$$s_i^-=f(x_i^-)r^-$$

式中：$f(x_i)\in[0,1]$，表示 x_i 在本类中的重要程度，一般与该点到本类中心的距离有关，以此来衡量数据点的偏离程度。另外，为了表示类的不平衡性，我们定义参数 r^+ 和 r^-，且满足 $r^+>r^-$。因此，正类的模糊隶属度于属于区间 $[0,r^+]$，而负类的模糊隶属度属于区间 $[0,r^-]$。这样设置隶属度，同时考虑到了特异点问题和类不平衡问题的特点。

函数 $f(x_i)$ 的定义如下

$$f(x_i)=1-\frac{d_i}{max(d_i)}$$

其中，$d_i=\|x_i-\bar{x}\|^{\frac{1}{2}}$，表示 x_i 到本类中心 $\bar{x}$ 的欧氏距离，$max(d_i)$ 表示所有数据点到本类中心最远的距离。

根据参考文献［7］中的论述，当多数类和少数类的权值之比等于其样本数量之比时最优。因此我们设置 $r^+=1$，$r^-=r$，其中 r 是少数类与多数类之比。因此，正类的模糊隶属度可以介于［0，1］，而负类的模糊隶属度介于［0，r］，此处 $r<1$。

这样我们就定义了模糊隶属度函数 s_i 的数学形式，它可以较好地反映出软件缺陷预测中的特异点问题和类不平衡问题。

2 实验验证

为了验证算法的有效性，将本文的 GA-FSVM 与其它常见算法在 NASA 数据集上进行对比验证，并采用 F-measure 值对分类器的性能进行评价。F-measure 值越高说明分类器的性能越好。

2.1 数据集

NASA 数据集是由美国国家航空航天局提供的一个公开软件仓库，这些数据均来自于 NASA 的实际软件项目。NASA 数据集在软件缺陷预测领域已经被广泛使用，方便了广大研究人员进行实验的重现和改进。因此本文也选择此数据集的一部分作为实验数据。

需要注意的是，Martin Shepperd 发现 NASA 数据集存在数据质量问题[8]，比如它存在着严重的数据重复、无效特征和自相矛盾的数据等，因此他们对数据进行了清洗整理并发布于互联网。

本文使用的数据集是经过数据清洗整理之后的数据，可以从对应的网站下载[9]。表 1 展示了实验用到的数据集的基本特征。

表 1 实验所用数据集的基本特征

	CM1	JM1	KC1	KC3	MW1	PC3	PC5
属性个数	38	22	22	40	38	38	39
样本数量	327	7782	1183	194	253	1077	1711
正类数量	42	1672	314	36	27	134	471
负类数量	285	6110	869	158	226	943	1240
正负比例	0.147	0.274	0.361	0.228	0.119	0.142	0.379

2.2 实验步骤

遗传算法的实验步骤如图 1 所示。

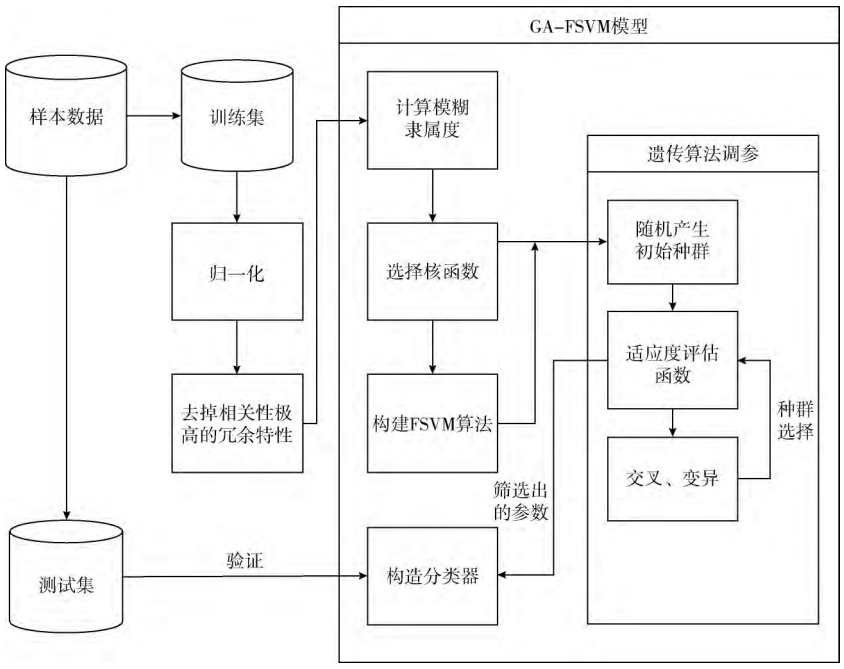

图 1 实验步骤框架

输入：样本数据

步骤 1 将样本数据分割为训练集和测试集。

步骤 2 将数据集归一化处理。

步骤 3 对于训练集中冗余度过高的特征予以删除。

步骤 4 将经过前 3 步处理的训练集输入到 GA-FSVM 算法模型中，核函数选择高斯核函数。

步骤 5 使用 GA-FSVM 进行训练。

步骤 6 构造出分类器。

步骤 7 使用测试集进行验证。

首先，本文将数据集随机分为 10 份，测试集为其中一份，其余的是训练集。在数据的预处理阶段，为了消除不同属性数据范围的不同而造成的影响，我们采用 min-max

标准化方法，将所有训练集和测试集的属性值映射到［0，1］之间。同时，我们注意到 NASA 数据集的部分属性冗余度很高，然而冗余的属性会降低分类器的收敛速度，因此模型仅保留这些冗余属性中的一个，并且剔除掉其余的。冗余的判断标准是：属性间相关系数大于 0.95 且 p 值小于 0.01。随后将经过数据预处理的数据集输入到 GA-FSVM 模型中去，FSVM 的核函数选择高斯核函数，使用遗传算法对 FSVM 进行参数调优。训练出模型后，利用测试集验证模型的性能。另外，本文使用 Weka 工具包验证其余常见算法在本数据集上的表现。

2.3 参数的选择

和 SVM 算法一样，FSVM 算法对参数的选择也非常敏感。传统的参数选取使用类似于穷举的方法，通过反复的实验，人工选取出令人满意的解。本文使用遗传算法进行参数优选[10]。

本文将 C 的范围取［1，8000］，σ 范围取［0.0001，1］，初始种群 1000，迭代次数 500 次。遗传算法的步骤见图 1。为了更好地展示实验的对比效果，本次实验将传统的 SVM 算法引入对照组，并使用同样的遗传算法进行调参，将该方法命名为 GA-SVM。

2.4 性能评价

衡量分类模型好坏的标准有很多，常见的评价指标有：准确率、F-measure、查全率、查准率等。但是对于类不平衡问题，多数类对准确率的影响大于少数类，导致少数类容易被忽视。因此本文采用 F-measure 进行性能评价，它能够同时表示出查全率和查准率的水平

$$F-meansure=\frac{2*Precision*Recall}{Precision+Recall}$$

Precision 和 *Recall* 分别是正类（少数类）的查准率和查全率。它们根据表 2 中的混淆矩阵进行定义。

表 2 混淆矩阵

	正类（预测）	负类（预测）
正类（真实）	True Positives (TP)	False Negatives (FN)
负类（真实）	False Positives (FP)	True Negatives (TN)

则

$$Recall=\frac{TP}{TP+FN}$$

$$Precision=\frac{TP}{TP+FP}$$

2.5 结果展示

本文将 GA-FSVM 和几个常见的机器学习算法进行了对比实验。实验进行十次十折交叉验证，结果取平均数。

从表 3 可以看出，在以上 7 个不平衡数据集中相比较于其它常见算法，GA-FSVM 都取得了更好的 F-measure 值，因此，GA-FSVM 可以在一定程度上减小类不平衡带来的问题。在 PC5 中，GA-FSVM 的训练效果不如 C4.5，但从整体来看，仍然比传统的算法更好。

表 3 实验结果

	CM1	JM1	KC1	KC3	MW1	PC3	PC5
Logistic	0.212	0.18	0.311	0.377	0.267	0.287	0.365
Naive Bayes	0.244	0.268	0.378	0.406	0.39	0.262	0.309
C4.5	0.211	0.291	0.427	0.375	0.195	0.302	0.48
GA-SVM	0.3	0.11	0.396	0.381	0.36	0.14	0.3325
GA-FSVM	0.374	0.322	0.45	0.476	0.432	0.318	0.3845

3 结束语

软件缺陷预测的显著问题之一就是类不平衡，它是由软件缺陷预测的自身特点决定的。传统的分类器在处理类不平衡问题时，预测结果会倾向于多数类而忽视少数类。本文使用模糊支持向量机作为工具来进行软件缺陷预测。针对软件缺陷预测中的类不平衡问题和传统支持向量机对特异点敏感的问题，恰当设置了模糊隶属度函数。在 NASA 数据集上的对比实验结果表明，GA-FSVM 相比其它常见分类算法，取得了更好的效果。

不可否认的是，软件缺陷预测还有很多问题亟待研究，例如：如何获取质量更高的数据集、如何选择更合适的度量元，或者考虑将集成学习技术引入到算法中等，这些都是可以关注的研究方向。

参考文献：

[1] CHEN Xiang, GU Qing, LIU Wangshu, et al. Study on static software defect prediction method [J]. Journal of Software, 2016, 27 (1): 1-25 (in Chinese). [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] Wang S, Yao X. Using class imbalance learning for software defect prediction [J]. IEEE Transactions on Reliability, 2013, 62 (2): 434-443.

[3] 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, 2012, 38 (6): 1276-1304.

[4] Lin C F, Wang S D. Fuzzy support vector machines [J]. IEEE Transactions on Neural Networks, 2002, 13 (2): 464-471.

[5] An W, Liang M. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises [J]. Neurocomputing, 2013, 110 (6): 101-110.

[6] Wang X, Liu X, Matwin S, et al. Applying instance-weighted support vector machines to class imbalanced datasets [C] //IEEE International Conference on Big Data. IEEE Computer Society, 2014: 112-118.

[7] Akbani R, Kwek S, Japkowicz N. Applying support vector machines to imbalanced datasets [C] //European conference on machine learning. Springer Berlin Heidelberg, 2004: 39-50.

[8] Shepperd M, Song Q, Sun Z, et al. Data quality: Some comments on the NASA software defect datasets [J]. IEEE Transactions on Software Engineering, 2013, 39 (9): 1208-1215.

[9] Shepperd M, Nasa iv&v facility, metric data program [EB/OL]. [2004-12-2]. http://openscience.us/repo/defect/mccabehalsted/.

[10] Di Martino S, Ferrucci F, Gravino C, et al. A genetic algorithm to configure support vector machines for predicting fault-prone components [C] //International Conference on Product Focused Software Process Improvement. Springer, 2011: 247-261.

(上接第 2752 页)

参考文献：

[1] Afridi M J, Liu X, Mcgrath J M. An automated system for plant-level disease rating in real world [J]. International Conference on Pattern Recognition, 2014, 8 (7): 148-153.

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [C] //International Conference on Neural Information Processing Systems. Curran Associates Inc, 2012: 1097-1105.

[3] Moon S, Kim S, Wang H. Multimodal transfer deep learning with applications in audio-visual recognition [C] //Multimodal Machine Learning. Quebec: NIPS, 2015.

[4] Shin H C, Roth H R, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning [J]. IEEE Transactions on Medical Imaging, 2016, 35 (5): 1285-1298.

[5] Xie M, Jean N, Burke M, et al. Transfer learning from deep features for remote sensing and poverty mapping [C] //Proc of the 30th AAAI Conference on Artificial Intelligence. Phoenix: Arizona USA, 2016.

[6] Shafik R A, Yang S, Das A, et al. Learning transfer-based adaptive energy minimization in embedded systems [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 35 (6): 877-890.

[7] Persello C, Bruzzone L. Kernel-based domain-invariant feature selection in hyperspectral images for transfer learning [J]. IEEE Transactions on Geoscience & Remote Sensing, 2016, 54 (5): 2615-2626.

[8] ZHUANG Fuzhen, LUO Ping, HE Qing, et al. Survey on transfer learning research [J]. Journal of Software, 2015, 26 (1): 26-39 (in Chinese). [庄福振，罗平，何清，等. 迁移学习研究进展 [J]. 软件学报，2015, 26 (1): 26-39.]

[9] Howard A G. Some improvements on deep convolutional neural network based image classification [C] //International Conference on Learning Representations, 2014.

[10] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [J]. International Conference on Neural Information Processing Systems, 2012, 25 (2): 1097-1105.

[11] Jia Y Q, Shelhamer E, Donahue J. Caffe-convolutional architecture for fast feature embedding [C] //Proc of the 22nd ACM international conference on Multimedia. Orlando: ACM, 2014.