文章编号:1001-9081(2017)S2-0060-05

基于粒子群优化 SVM 的面向对象软件缺陷预测模型

朱朝阳[1],陈相舟[1],王志宏[2],张信明[2*]

(1. 中国电力科学研究院 信息通信研究所,北京 100192; 2. 中国科学技术大学 计算机科学与技术学院,合肥 230027)

(* 通信作者电子邮箱 xinming@ustc.edu.cn)

摘 要:针对电力信息系统软件安全问题,分析了软件缺陷预测方法在面向对象软件开发过程中的重要性,并提出了一种与面向对象软件特征相应的基于粒子群优化的支持向量机软件预测模型。模型主要包括三部分:首先是对原数据进行归一化和特征选择的预处理模块;然后是以预测准确度作为适应度评价的动态惯性权重粒子群优化支持向量机(SVM)参数的模块;最后则是利用第二个模块中的最优参数进行降维数据预测的 SVM 分类模块。实验结果表明,该模型在四个数据集合上的准确率高于对比模型 8.2% ~12.2%,在精准度、查全率和 F 值上平均高出 9.9%,5.6%和 7.7%,说明了该模型的有效性。

关键词:软件缺陷预测;粒子群优化;特征选择;支持向量机;面向对象软件

中图分类号:TP311.53 **文献标志码**:A

Defect prediction model for object oriented software based on particle swarm optimized SVM

ZHU Chaoyang[1], CHEN Xiangzhou[1], WANG Zhihong[2], ZHANG Xinming[2*]

(1. *Information & Communication Department, China Electrical Power Research Institute, Beijing* 100192, *China*;
2. *School of Computer Science and Technology, University of Science and Technology of China, Hefei Anhui* 230027, *China*)

Abstract: In terms of the security problem of power information system, this paper analyzed the importance of the software defect prediction method in object-oriented software development, and proposed a software prediction model based on particle swarm optimized Support Vector Machine (SVM) corresponding to the features of object-oriented software. The model mainly consists of three parts: the first is the preprocessing module which normalizes the original data and selects feature, then the second is adaptive inertia weight particle swarm module which optimizes the parameters of SVM with the prediction accuracy as the fitness. Finally, the last SVM classification module predicts categories of reduced-dimension data using the optimal parameters from the second module. Experimental results show that the accuracy of the proposed model is 8.2% - 12.2% higher than the comparative model, and 9.9%, 5.6% and 7.7% higher on the precision, recall and F value, which proves the validity of the proposed model.

Key words: software defect prediction; particle swarm optimization; feature selection; Support Vector Machine (SVM); object oriented software

0 引言

电力信息系统软件在长期发展过程中,主要是以面向对象设计技术来开发的。从本质上讲,面向对象系统设计是寻求电力软件结构和电力软件功能模型解决方案的过程。因而为了功能的全面性、可靠性,面向对象电力系统软件逐步从功能单一向全面发展,软件本身结构和模型更加复杂,对象规模也更加庞大,导致整个电力信息系统的软件所面临的安全问题也愈加严重,软件缺陷预测能够在软件开发早期预测待测的模块、对象是否有出错倾向从而提出相应的解决方案,包括人力资金等资源的分配,开发进度的控制和安排。因此软件开发过程中尽早地发现软件缺陷、漏洞并尽快解决,才能为国民生产、市场正常运行提供必要的保障,也是使得电网公司降低日后测试代价和周期、提高软件质量的重要途径[1]。

软件缺陷预测技术可分为静态和动态两种缺陷预测技术。动态缺陷预测技术关注软件整个生命周期或者测试阶段缺陷分布及数量随时间的变化规律,并据此预测软件未来缺陷分布;而更加常用的静态预测技术则更关注软件关于缺陷的度量指标,结合各个不同的缺陷相关属性的度量进行缺陷预测,在软件分析和设计阶段及开发的早期阶段均可提供相应的预测功能。现有的静态缺陷预测技术基本上都是基于不同的机器学习算法提出的,例如决策树、随机森林、朴素贝叶斯、BP 神经网络以及人工免疫系统等分类算法,根据某种描述软件特性的软件度量获取模块属性,然后对其进行分类(有缺陷或无缺陷)[2]。这些方法均拥有一定程度的缺陷预测能力,但又或多或少地隐含一些问题,例如决策树过度拟合,忽略特征属性之间相关性的问题;朴素贝叶斯需要已知先验概率、对属性独立性要求较高;神经网络容易陷入局部最优或者拟合程度不够的问题,同贝叶斯模型一样,需要根据专家经验来获取与缺陷相关的因子,计算效率低;支持向量机具有良好的学习和扩展能力,但其最优参数的设置没有统一高效的方法。而且在应对面向对象软件时各类算法都不可避免地需要应对非常多的类和对象特征属性来度量软件,导致“维数灾难”,检测时间过长,预测模型实用性降低。

收稿日期:2017-02-15。 基金项目:国家自然科学基金资助项目(61672485,61379130);国家电网公司科技项目(XX71-15-036)。

作者简介:朱朝阳(1974—),男,江西南昌人,高级工程师,博士,主要研究方向:信息系统安全; 陈相舟(1986—),男,湖南常德人,工程师,硕士,主要研究方向:信息通信仿真与测试; 王志宏(1993—),男,甘肃天水人,硕士研究生,主要研究方向:无线网络、车载自组织网络;张信明(1964—),男,安徽天长人,教授,博士,CCF 高级会员,主要研究方向:无线网络、智能电网。

软件度量方法作为获取软件架构和模块属性的一类标准化方法,实质是按照相应的度量规则,定量地分析软件实体的属性、描述软件及其特征,为软件质量评估、基于度量的软件缺陷预测等工作提供必需的数据来源。传统的结构化软件度量主要包括 McCabe 结构复杂性度量、LOC 语句行度量和 Halstead 软件科学度量等。复杂性软件度量针对的是软件模块内属性,而随着面向对象软件技术的发展,模块间的交互更加显著,因此复杂度度量标准也应当考虑到模块间的属性。以前的结构化度量在面向对象分析和面向对象设计过程中无法提取出关于数据抽象、封装、继承、多态、信息隐藏、内聚和耦合等面向对象软件特有的属性[3-4],依照传统的度量标准得到的软件的特征属性不足以充分表示面向对象软件的内在特性,因而,需要针对面向对象的软件提出特定的度量模型。目前最具代表性的面向对象软件的度量模型为 MOOD 模型和 Chidamber 和 Kemerer 提出的 CK 度量[5],在 CK 度量中包含 6 个具有严格度量理论基础的用于描绘面向对象软件设计规模和复杂度的度量属性,这些度量指标包括: 类的加权方法数 WMC、继承树的深度 DIT、类的孩子数目 NOC、对象之间的耦合 CBO、类的响应集合 RFC 和类方法内聚缺乏度 LCOM。这六个指标包含了面向对象软件的大部分特征,而且在各个数据集中,又对其属性集进行了相应的扩充以应对软件复杂程度的提高。本文中采用较为灵活的 CK 度量进行软件缺陷检测来验证我们所提模型的有效性[5]。

为了优化软件缺陷预测方法的性能,本文提出了一种基于粒子群优化 SVM(Support Vector Machine) 的面向对象软件缺陷预测模型,即先通过 Relief 算法进行特征选择,再利用结合了 SVM 较强的泛化能力和随粒子聚集程度动态更新惯性权重的 PSO(Particle Swarm Optimization) 算法高效的寻优能力的模型进行缺陷预测,获取最优的预测结果。

1　问题描述及相关算法

1.1　问题描述

本文研究电力信息系统软件中缺陷检测问题,软件缺陷的产生在编程实现中是不可避免的,并且会对软件质量产生重要的影响,因此我们需要利用统计学习技术,根据历史数据或现有的故障数据集以及已经发现的缺陷等软件度量数据预测软件系统的缺陷数目及类型,统计没有发现但还可能存在的缺陷数,以决定系统是否可以交付使用。在使用统计学习技术的过程中,首先为了避免“维度灾难”防止处理代价太高,需要对分类集合的特征属性进行降维处理,并对各属性值进行归一化处理; 其次需要给选定分类算法设置恰当的参数,而通过专家经验等设置的参数总是无法相对准确地展示新的预测问题的特性。因此,也需要合适的优化算法求出这些分类算法的相应参数,最终得到最优的缺陷预测结果。

1.2　支持向量机

SVM 是一种通用的前馈神经网络,可以用于模式分类和非线性回归[6],SVM 学习算法来自于统计学理论,其核心是经验风险最小化原则,基本思想是求解核函数和二次规划问题。本文选用 SVM 的原因在于其比较适合从有限样本中获得分类最优解,同时由于核函数的引入使得 SVM 模型能够高效地应对软件缺陷检测这类非线性的问题。核函数将数据映射到高维特征空间,使得样本集合在高维特征空间线性可分,最常用的核函数为高斯核函数,其在保证核矩阵半正定的同时也保留了向无限维空间映射的能力[6],因此适用范围更广,学习能力也较强。

给定训练样本集合 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$,$y_i \in \{-1, +1\}$,该样本集的样本数量为 n,每个样本特征属性维度为 d,即 $x_i = (x_i^1, x_i^2, \cdots, x_i^d)$。支持向量机分类的目的在于在特征空间中找到一个使两个异类支持向量间最大间隔的划分超平面 $w^{\mathrm{T}}x + b = 0$,将样本按类分开,其生成的分类结果最稳定,对未来样本处理能力最强。为了找到间隔最大的划分超平面,需要利用拉格朗日乘子法得到 SVM 凸二次规划问题的基本模型的对偶问题。其基本型的拉格朗日函数为:

$$L(w, b, k) = \sum_{i=1}^{d} k_i(1 - y_i(w^{\mathrm{T}}x_i + b)) + \frac{1}{2}\|w\|^2 \quad (1)$$

据此可得到对偶问题

$$\max_k P = \sum_{i=1}^{d} k_i - \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} k_i k_j y_i y_j x_i^{\mathrm{T}} x_j$$

$$\text{s.t.} \sum_{i=1}^{d} k_i y_i = 0 \quad (2)$$

为了解决非线性划分的问题,又将 x 映射到高位特征空间,原模型转换为 $f(x) = w^{\mathrm{T}}\varphi(x) + b$,其中 $\varphi(x)$ 表示将 x 射到高维特征空间后的特征向量,对偶问题也转换为:

$$\max_k P = \sum_{i=1}^{d} k_i - \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} k_i k_j y_i y_j \varphi(x_i)^{\mathrm{T}}\varphi(x_j)$$

$$\text{s.t.} \sum_{i=1}^{d} k_i y_i = 0;\ k_i \leqslant 0 \quad (3)$$

其中,核函数 $\varphi(x_i)^{\mathrm{T}}\varphi(x_j)$ 是高维特征空间的内积,将其定义为在原来样本特征空间中函数 $\varphi(x_i, y_i)$ 的结果,其隐式地定义了映射的特征空间[7]。

利用以上模型进行样本分类的时候,可能出现“硬间隔”问题,即要求每个样本都符合约束要求,其代价过高,而且难以保证分类结果不是源于过拟合。为了缓解这个问题,放宽了样本的约束条件,尽量在获得最大间隔的同时出现尽可能少的不符合条件的样本。于是,在优化目标中加入了常用替代损失函数 hinge 并引入松弛变量即可得到新的优化目标:

$$\min_{w, b, \zeta} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{d}\zeta_i$$

$$\text{s.t.}\ y_i(w^{\mathrm{T}}x_i + b) + \zeta_i \geqslant 1;\ \zeta_i \geqslant 0 \quad (4)$$

其前项表示的是支持向量间隔,正则化后,可以称其为“结构风险”,描述了模型的性质; 后项则表示误差,称为“经验风险”,描述了训练数据集与当前模型的匹配程度; 参数惩罚因子 C 用来平衡二者的权重,以经验最小化的原则来获取符合人们要求的模型。同样通过拉格朗日乘子法得到优化目标函数拉格朗日函数:

$$L(w, \kappa, \lambda, \zeta, b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{d}\zeta_i + \sum_{i=1}^{d} k_i(1 - y_i(w^{\mathrm{T}}x_i + b) - \zeta_i) - \sum_{i=1}^{d}\lambda_i\zeta_i \quad (5)$$

据此得到目标函数的对偶问题:

$$\max_k P = \sum_{i=1}^{d} k_i - \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} k_i k_j y_i y_j \varphi(x_i, x_j)$$

$$\text{s.t.} \sum_{i=1}^{d} k_i y_i = 0;\ 0 \leqslant k_i \leqslant C \quad (6)$$

求解后得到支持向量展式 $f(x) = \sum_{i=1}^{d} k_i y_i \varphi(x_i, x_j) + b$,因而分类函数如下:

$$classify(x) = \mathrm{sgn}\left(\sum_{i=1}^{d} k_i y_i \varphi(x, x_i) + b\right) \quad (7)$$

1.3 粒子群算法

Eberhart 和 Kennedy 基于对鸟群捕食的行为研究提出一种进化技术——粒子群优化算法(PSO)[8]。PSO 算法先初始化一组随机种群,又叫粒子群,每个粒子按初始化得到的速度穿过解空间,其速度是该粒子和种群其他粒子的历史行为(速度、位置和适应度)的函数,在每一次迭代中都会发生变化。其中每个粒子的位置属性对应一个潜在的解,通过计算当前粒子的适应度来评价当前解的优劣,如果未能符合解的要求则进行粒子速度和位置的更新,并进入下一个迭代过程直到找到符合要求的最优解或者到达最大迭代次数。

2 基于粒子群优化的支持向量机软件缺陷预测

2.1 软件缺陷预测模型

本文提出的软件预测模型如图1所示:首先对训练样本进行归一化和数值化预处理和 Relief 算法进行特征选择,解决单纯的 PSO-SVM 模型处理大样本数据性能低下、适应度低的问题,然后利用 PSO 算法对 SVM 模型的参数惩罚因子 C 和高斯核的带宽 σ 进行优化,直到找到使得分类模型准确率最高的参数,再利用训练好的模型对测试数据及未来数据块进行缺陷预测。

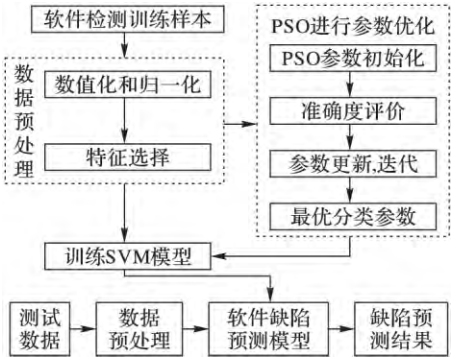

图1 基于粒子群优化的支持向量机软件缺陷预测模型

2.2 特征属性预处理

在实际项目中,每个训练及待测数据集会包含大量的样本特征属性,这为缺陷预测模型带来了极为沉重的计算负担;而且,由于每个属性在分类中的贡献程度不一,无法任意地删除或者添加一些属性,因此我们需要特征选择算法 Relief 算法来解决该问题。Relief 算法是专为两类数据的分类问题设计的,可以对软件缺陷预测问题计算各个特征属性的权重和排名,在完成特征选择后依然可以保证分类模型拥有相当的准确度,并且其运行效率非常高。算法[9]如下:

样本实例空间为 $\boldsymbol{T}=\{\boldsymbol{T}_1,\boldsymbol{T}_2,\cdots,\boldsymbol{T}_n\}$,其中 $\boldsymbol{T}_i=(t_1,t_2,\cdots,t_m)$。

步骤1 为样本的每个特征属性的权值赋值0,并将非数值表示的属性数值化,再按照最大最小归一化的方法将所有的数值作归一化处理。

步骤2 随机选择一个样本 $\boldsymbol{T}_i$,并分别从它的同类型样本中和非同类型样本中选出最近邻样本 $\boldsymbol{T}_{\text{hit}}$ 和 $\boldsymbol{T}_{\text{miss}}$。

步骤3 按照如下等式更新每个属性 t_k 的权值:

$$weight(t_k)=weight(t_k)+\frac{1}{n}\times\frac{D(\boldsymbol{T}_i,\boldsymbol{T}_{\text{miss}},t_k)}{\max(D(t_k))}-\frac{1}{n}\times\frac{D(\boldsymbol{T}_i,\boldsymbol{T}_{\text{hit}})}{\max(D(t_k))} \tag{8}$$

其中,$D(\boldsymbol{T}_i,\boldsymbol{T}_j,t_k)$ 表示 $\boldsymbol{T}_i$ 和 $\boldsymbol{T}_j$ 在属性 t_k 上的欧氏距离,$\max(D(t_k))$ 表示所有样本在属性 t_k 上的最大欧氏距离。

步骤4 从步骤2开始迭代该过程 n 次,最后计算每个特征属性的平均权重,据此对特征属性进行排序,再与特征属性阈值比较或按预设的数量得到最终保留的特征属性集合,用于支持向量机模型训练。

2.3 软件缺陷预测模型参数的优化

在第1章已经讨论了支持向量机模型,为了提高软件缺陷预测模型的泛化能力,选择的核函数为高斯核函数(Radius Basis Function, RBF),即:

$$\varphi(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-\|\boldsymbol{x}_i^2-\boldsymbol{x}_j^2\|/2\sigma^2) \tag{9}$$

因此在本文提出的软件缺陷预测模型中,需要利用粒子群优化算法优化的相关参数有惩罚因子 C 和高斯核的带宽 σ。算法[10-11]如下:

步骤1 初始化速度区间、学习因子 c_1 和 c_2、惯性权重 w 及迭代次数和粒子群 $S=\{(s_{1\_c},s_{1\_\sigma}),(s_{2\_c},s_{2\_\sigma}),\cdots,(s_{num\_c},s_{num\_\sigma})\}$,包括每个粒子的位置$(s_{i_c},s_{i_\sigma})$与速度$(v_{i_c},v_{i_\sigma})$,$num$ 为种群数量。

步骤2 对每个粒子位置计算该粒子的本次迭代的适应度 $fitness(\boldsymbol{S}_i^k)$,本文中以当前惩罚因子 s_{i_c} 和高斯核带宽 s_{i_σ} 得到的 SVM 模型的预测准确率作为适应度函数返回值,即:

$$fitness(\boldsymbol{S}_i^k)=accuracy_{\text{SVM}}(s_{i\_c},s_{i\_\sigma}) \tag{10}$$

步骤3 如果 $fitness(\boldsymbol{S}_i^k)$ 优于个体极值 $\boldsymbol{PBest}_i$,则用其更新该粒子的个体极值,如果 $fitness(\boldsymbol{S}_i^k)$ 同时也优于所有其他粒子的个体极值以及上一轮迭代中的群体极值 $\boldsymbol{GBest}^{k-1}$,则用其更新本次迭代中的群体极值 $\boldsymbol{GBest}^k$。

步骤4 如果到达最大迭代次数或当前群体极值 $\boldsymbol{GBest}^k$ 满足精度要求则可以退出迭代并输出群体极值 $\boldsymbol{GBest}^k$ 作为训练 SVM 模型的最优参数;否则,根据等式

$$\begin{cases}\boldsymbol{V}_i^{k+1}=w\times\boldsymbol{V}_i^k+c_1\times rand1()\times(\boldsymbol{PBest}_i^k-\boldsymbol{S}_i^k)+\\ \qquad c_2\times rand2()\times(\boldsymbol{GBest}^k-\boldsymbol{S}_i^k)\\ \boldsymbol{S}_i^{k+1}=\boldsymbol{S}_i^k+\boldsymbol{V}_i^{k+1}\end{cases} \tag{11}$$

更新每个粒子的速度和位置,并再次从步骤2开始进入下一轮迭代。其中,c_1 和 c_2 主要影响的是粒子个体记忆和种群记忆的权衡,根据经验,设置 $c_1=1.6$,$c_2=1.5$。

惯性权重 w 主要影响的是粒子的历史记忆和当前状态的平衡:如果取值过大则在靠近最优解时,粒子仍为了不陷入局部最优,关注全局搜索的结果而忽视局部搜索影响,导致越过最优解;否则,移动速度太慢,导致无法尽快地向最优解靠近。因此本文提出了动态惯性权重粒子群优化算法,使得种群在快速向最优解附近集中的过程中,逐渐降低移动速度,各个粒子能够更加精确地搜索周围空间的适应度,增强标准 PSO 算法的整体性能。根据以上分析,本文定义一个变量 $close$ 来表示种群的聚集程度,每次迭代 k 的种群聚集程度如下:

$$close_k=\frac{1}{num\times|\boldsymbol{S}_{\max}-\boldsymbol{S}_{\min}|}\times\sum_{i=1}^{num}D(\boldsymbol{S}_i^k,\bar{\boldsymbol{S}}_i^k) \tag{12}$$

$close\in(0,1)$。其中:$D(\boldsymbol{S}_i^k,\bar{\boldsymbol{S}}_i^k)$ 表示每个粒子与种群平均位置(重心位置)的欧氏距离,$|\boldsymbol{S}_{\max}-\boldsymbol{S}_{\min}|$ 表示解空间的最大直径长度,$close$ 描述粒子群在每次迭代后的向最优解空间靠近的情况,其值越大,说明种群越分散,反之则越集中。

当粒子聚集之后,需要逐渐减小惯性权重 w 的值,反之则需要增大惯性权重,通过量化粒子的聚集程度,可以将其映射到惯性权重的解空间,得到在不同集中程度下的惯性权重的值。为了实现以上目的定义式(13)来计算 w 的值,

$$w_k = \frac{1}{1+\exp(-10(close_k - 0.38))} \times (w_{max} - w_{min}) + w_{min} \quad (13)$$

使其能在快速靠近最优空间后优化局部寻优，其中 w_{min} 和 w_{max} 是 w 的下界和下界，按照经验分别设置为 0.8 和 1.2。

3　基于 PSO-SVM 的软件预测模型实验

3.1　实验数据集

为了验证本文所提出的面向对象软件缺陷预测模型的性能，基于 Matlab 实现了文中所提模型，并与 LE-SVM [3] 和 LE-KNN [3] 进行了对比。文中使用 4 个符合 CK 度量的实验数据集来验证缺陷预测模型的有效性，一个是美国国家航空航天局(NASA)提供的 Class-level data for KC1 [12]，包含 145 个样本，共计 89 个特征属性和 60 个无缺陷样本，85 个有缺陷样本；第二个是基于开源 eclipse 的真实数据的 eclipse2.0 数据集 [13]，有 6 728 个不同样本，其中包含 975 个有缺陷样本，5 753个无缺陷样本，第三个是 eclipse3.0 数据集 [13]，其中包含 9 470 个样本，1 522 个有缺陷样本；最后是 ant-1.7 数据集 [14]，有 745 个样本，共 166 个无缺陷样本。对于 eclipse 和 ant 数据集，由于有无缺陷是通过 bug 数量表示的，因而首先需要将其更新为表示有无缺陷的逻辑变量 1 和 0；同时由于流形学习算法在高维降维的过程中会出现数据点丢失的问题，从后三个数据集中随机挑选 700 个样本，并将其随机分为数量相等的两组，分别作为训练集和测试集。

本文中 PSO 算法的种群大小设置为 50，进化次数设置为 100，同时采用十折交叉检验方法训练 SVM 模型，并据此获得 SVM 在测试集上的缺陷检测准确率。SVM 模型是基于 libsvm-3.21 在 Matlab R2015a 上实现的 [15]，LE 算法中输出特征维度为 10，KC1 数据集上 LE-SVM 算法的 SVM 模型的惩罚因子 $C=1$，其高斯核带宽 $\sigma=0.011\,76$，其他数据集 LE-SVM 算法的 SVM 模型的惩罚因子 $C=1$，其高斯核带宽 $\sigma=0.041\,76$，而在本文提出的算法中，Relief 保留的特征维数为 15，SVM 模型的惩罚因子 C 的搜索空间为(0.1，100.0)，高斯核参数的搜索空间为(0.01，10.00)。

表 1　实际缺陷情况与预测结果的交叉矩阵

实际缺陷情况	缺陷预测结果	
	有缺陷模块	无缺陷模块
有缺陷模块	正确预测有缺陷 -N1	错误预测无缺陷 -N2
无缺陷模块	错误预测有缺陷 -N3	正确预测无缺陷 -N4

测试样本总数为 $N=N1+N2+N3+N4$，预测正确的样本数为 $N1+N4$，预测错误的样本总数为 $N2+N3$，本文中需要按照通用的评价指标来分析模型缺陷预测的性能优劣 [16]：

准确度(*accuracy*)表示预测结果正确(有缺陷模块被成功检测为有缺陷，无缺陷模块没有被误判)的样本数与被预测样本总数的比值，计算公式如下：

$$accuracy = (N1+N4)/N \quad (14)$$

精准度(*precision*)表示实际有缺陷且预测有缺陷的样本数与所有预测有缺陷样本数目的比值，可表示为：

$$precision = N1/(N1+N4) \quad (15)$$

查全率(*recall*)表示实际有缺陷且预测有缺陷的样本数目与所有实际有缺陷样本的比值，计算公式如下：

$$recall = N1/(N1+N2) \quad (16)$$

F 值为精准度和查全率的调和平均值，可表示为下式：

$$F = \frac{2}{1/recall + 1/precision} \quad (17)$$

3.2　结果分析

本文将 SVM 模型的训练准确率作为 PSO 算法的适应度，因而准确率更加客观地反映了本文所提软件缺陷预测模型与对比模型的性能差异。图 2 显示了所提模型与对比模型在四个数据集上的准确率，从中可以看出所提模型在四个数据集上均领先于对比算法，主要在于：所提模型可以通过 Relief 算法去除一些对于分类不利的属性，例如数值差异太小的属性，使得预测结果更加准确；同时通过 PSO 获得使 SVM 训练模型性能最优的惩罚因子和高斯核带宽，进一步提高了预测结果的准确率。而对比算法的参数则只能通过经验获取，缺少寻优的过程，使得其结果离最优解有一定的距离，因而预测结果相对于所提模型有一定的差距，所提模型在四个数据集合上的准确率高于对比模型 8.2% ~12.2% 不等。

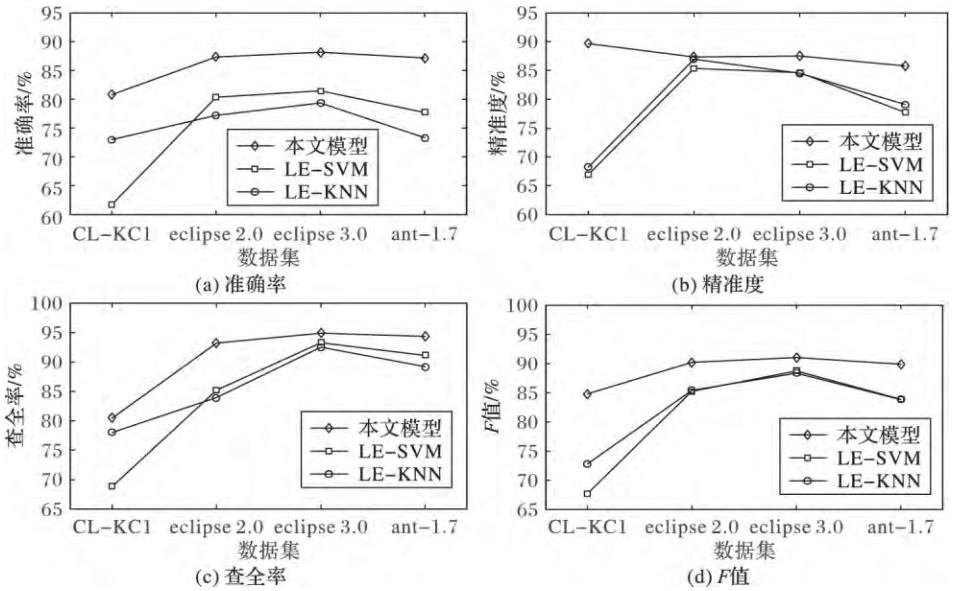

图 2　各个数据集上的性能

在确保预测模型有足够良好的缺陷预测准确性的基础上，即保证模型整体性能的情况下，需要提高缺陷预测模型对于测试集中有缺陷模块的识别能力，即获取更高的预测精准度和查全率，这两项指标分别说明了模型预测有缺陷的结果中有缺陷的比重和预测有缺陷的结果在有缺陷模块中的比重，前者反映需要在无缺陷模块上检测开销，后者反映了需要在预测无缺陷模块上重新预测的额外开销，因此为了确保较低的额外开销，需要尽量提高精准度和查全率。

F 值是精准度和查全率的调和平均值，相当于对于两个指标的平衡和综合考虑，评价该指标的原因在于避免考虑单个指标很高的情况下，另外一个指标却很差导致额外代价并不能降低的问题。

图2(a) ~(d)分别展示了所提模型和对比模型在四个数据集上的精准度、查全率和 *F* 值，从中可以看出 LE-SVM 和 LE-KNN 模型的三项指标在后三个数据集上相差不多的，在 CL-KC1 上的差距也主要是查全率导致的，出现这种现象主要在于流形学习进行特征降维处理之后，并非保留部分原来的特征属性，而是将原来数据集中的主要信息保存在新生成的低维数据集中，对于这类数据，所采用的预测模型对预测结果的影响会降低，同理，图2中两种模型的准确率的差距也相对较小，同样也能说明该问题。而我们所提出的预测模型在进行降维处理时利用了高效且极为适用于二分类问题的 Relief 算法，因为在惩罚因子和高斯核带宽解空间的寻优处理，通过改进的动态惯性权重 PSO 算法最大限度地避免了收敛速度固定和局部最优的问题，使得算法在面对不同的测试集的时候拥有了相应的最优模型参数，因而获得最优的缺陷预测结果。通过计算三种模型在四个数据集上各指标的平均值，可以得到所提模型比性能较优的 LE-SVM 模型在精准度上有9.9%，在查全率上有5.6%，在 *F* 值上有7.7%的领先。

4 结语

本文就电力信息系统中面向对象软件缺陷预测的问题提出了相应的缺陷预测模型，其基本思想在于 Relief 算法进行特征选择避免特征维度过高的问题，再通过根据粒子聚集程度动态更新惯性权重的 PSO 获取最优 SVM 训练模型，从而利用其对 CK 度量的软件模块进行缺陷预测。实验结果表明，该模型在四项通用指标上均有良好的表现，能够得到良好的预测结果。同时该模型只能作为静态分析模型，无法应对软件运行过程中的预测问题，而且该模型由于 PSO 算法的迭代循环过程，在时间性能上的表现还需要进一步的优化。下一步可以在模型预测结果和算法运行时间上进行优化，以应对未来的缺陷预测工作。

参考文献：

[1] 励刚，苏寅生，陈陈. 电力系统软件体系结构和框架设计[J]. 计算机应用，2001，21(9)：78－80.

[2] ABAEI G，SELAMAT A. A survey on software fault detection based on different prediction approaches [J]. Vietnam Journal of Computer Science，2014，1(2)：79－95.

[3] 石陆魁，马春娟，王靖鑫，等. 基于流形学习的面向对象的软件缺陷预测模型[J]. 计算机工程与设计，2014，35(11)：3859－3863.

[4] 张垚，袁志海，江海燕. 一种面向对象软件缺陷的早期预测方法[J]. 计算机技术与发展，2010，20(8)：37－40.

[5] 易彤. 面向对象设计中软件度量学：回顾与热点[J]. 计算机应用研究，2011，28(2)：427－434.

[6] CORTES C，VAPNIK V. Support-vector networks [J]. Machine Learning，1995，20(3)：273－297.

[7] CRISTIANINI N，SCHOLKOPF B. Support vector machines and kernel methods：the new generation of learning machines [J]. AI Magazine，2002，23(3)：31－41.

[8] KENNEDY J. Particle swarm optimization [M]// Encyclopedia of Machine Learning. Berlin：Springer，2011：760－766.

[9] KIRA K，RENDELL L A. A practical approach to feature selection [C]// Proceedings of the Ninth International Workshop on Machine Learning. San Francisco：Morgan Kaufmann Publishers，1992：249－256.

[10] 尚文利，张盛山，万明，等. 基于 PSO-SVM 的 Modbus TCP 通讯的异常检测方法[J]. 电子学报，2014，42(11)：2314－2320.

[11] HUANG C L，DUN J F. A distributed PSO-SVM hybrid system with feature selection and parameter optimization [J]. Applied Soft Computing，2008，8(4)：1381－1391.

[12] BOETTICHER G，MENZIES T，OSTRAND T. PROMISE Repository of empirical software engineering data [DB/OL]. (2007－01－01) [2013－03－17]. http://promisedata.org/repository.

[13] ZIMMERMANN T，PREMRAJ R，ZELLER A. Predicting defects for eclipse [C]// Proceedings of the 3rd International Workshop on Predictor Models in Software Engineering. Washington，DC：IEEE Computer Society，2007：9.

[14] JURECZKO M，MADEYSKI L. Towards identifying software project clusters with regard to defect prediction [C]// Proceedings of the 6th International Conference on Predictive Models in Software Engineering. New York：ACM，2010：Article No. 9.

[15] CHANG C C，LIN C J. LIBSVM：a library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology，2011，2(3)：Article No. 27.

[16] 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.

(上接第51页)

[31] 胡鹤，胡昌振，姚淑萍. 应用部分马尔科夫博弈的网络安全主动响应决策模型[J]. 西安交通大学学报，2011，45(4)：18－24.

[32] 席荣荣，云晓春，张永铮，等. 一种改进的网络安全态势量化评估方法[J]. 计算机学报，2015，38(4)：749－758.

[33] 张勇，谭小彬，崔孝林，等. 基于 Markov 博弈模型的网络安全态势感知方法[J]. 软件学报，2011，22(3)：495－508.

[34] 王元卓，林闯，程学旗，等. 基于随机博弈模型的网络攻防量化分析方法[J]. 计算机学报，2010，33(9)：1748－1762.

[35] 孙薇，孔祥维，何德全，等. 基于演化博弈论的信息安全攻防问题研究[J]. 情报科学，2008，26(9)：1408－1412.

[36] 朱建明，宋彪，黄启发. 基于系统动力学的网络安全攻防演化博弈模型[J]. 通信学报，2014，35(1)：54－61.

[37] 王纯子，黄光球. 基于粗糙贝叶斯博弈的网络攻防策略[J]. 计算机应用，2011，31(3)：784－789.

[38] 陈永强，吴晓平，付钰，等. 基于模糊静态贝叶斯博弈的网络主动防御策略选取[J]. 计算机应用研究，2015，32(3)：887－889.

[39] 张恒巍，余定坤，韩继红，等. 信号博弈网络安全威胁评估方法[J]. 西安电子科技大学学报(自然科学版)，2016，43(3)：137－143.