基于 ACO-SVM 的软件缺陷预测模型的研究

姜慧研[1),2)] 宗 茂[1)] 刘相莹[1)]

1)(东北大学软件学院 沈阳 110819)

2)(东北大学医学影像计算教育部重点实验室 沈阳 110819)

摘 要 针对传统软件缺陷预测模型的应用范围通常被局限在一定的子空间而影响其适用性和准确性的问题,文中利用支持向量机(SVM)的非线性运算能力和蚁群优化算法(ACO)的寻优能力提出了一种基于 ACO-SVM 的软件缺陷预测模型.文中首先对待预测的数据进行主成分分析降低数据的维数以提高运算速度,然后根据蚁群优化算法来计算最优的SVM 参数,然后再运用 SVM 进行软件缺陷的预测.并基于十折交叉方法进行实验,通过与传统方法的对比,证明文中方法具有较高的预测精度.

关键词 软件测试;软件缺陷预测;支持向量机;蚁群算法;主成分分析

中图法分类号 TP311 **DOI** 号:10.3724/SP.J.1016.2011.01148

Research of Software Defect Prediction Model Based on ACO-SVM

JIANG Hui-Yan[1),2)] ZONG Mao[1)] LIU Xiang-Ying[1)]

1)(*Software College, Northeastern University, Shenyang* 110819)

2)(*Key Laboratory of Medical Image Computing of Ministry of Education (Northeastern University), Shenyang* 110819)

Abstract The application of traditional software defect prediction model is limited for its low accuracy and applicability. This paper puts forward a software defect prediction model based on ACO-SVM, which take advantage of non-linear computing power of the SVM and parameters optimization power of the ACO. Firstly, use PCA to reduce the dimensions of software defect metrics to increase the computation speed. Secondly, use ACO to find the optimal parameters for SVM automatically. Last, predict software defect using SAM with the optimal parameters. During the experiments, the authors used the 10-fold experiment methods. The experiment results indicate that the method has a higher prediction precision than the traditional software defect prediction model.

Keywords software testing; software defect prediction; SVM; ant colony optimization (ACO); PCA

1 引 言

随着计算机系统应用领域的不断扩大,软件缺陷预测[1]问题变得越来越受到人们的关注.例如,在银行和股票等系统中,由于系统一旦失效将会导致巨大的经济损失,软件缺陷是软件开发首要因素.而软件缺陷预测模型能够在软件开发的早期预测出哪些模块有出错的倾向从而找到相应的解决方案,是软件可靠性工程的重要组成部分,对提高软件可靠性具有重要的意义.

目前,软件缺陷预测模型主要包括马尔可夫模型[2]、分类回归树模型[3]、人工神经网络模型[4]、线性判别分析模型[5]、时间序列分析模型、分类树模

收稿日期:2010-11-19;最终修改稿收到日期:2011-05-03.本课题得到国家自然科学基金(60973071,50834009)、辽宁省自然科学基金(20092004)资助.姜慧研,女,1963 年生,博士,教授,主要研究领域为模式识别、图像处理与分析等.E-mail:hyjiang@mail.neu.edu.cn.宗 茂,男,1987 年生,硕士研究生,主要研究方向为图像分割.刘相莹,女,1986 年生,硕士研究生,主要研究方向为模式识别.

型[6]等, 但这些方法尚存在一定问题, 难以达到理想的效果. 例如, 马尔可夫模型需要对软件内部错误及失效过程的特性做出很多假设; 分类回归树模型的泛化能力差; 人工神经网络模型的网络结构选择尚无统一完整的理论指导.

逻辑回归(Logistic Regression, LR)[7]是对线性回归的进一步改善, 线性回归难以解决分类时出现的 0~1 之外的非法概率值问题, 针对这一问题, LR 采用对数变换, 使其结果值不再局限于 0~1 范围, 而是负无穷大和正无穷大之间的任意值. 由于 LR 模型进行二次判别可以显著提高计算速度和识别率, 所以被广泛应用于构建软件缺陷预测模型. 但是该方法适用于大样本的情况, 对小样本的预测效果不理想.

支持向量机[8] (Support Vector Machine, SVM)是 Cortes 和 Vapnik 等人于 1995 年首先提出的, 是在统计学习理论基础上发展起来的一种新的机器学习方法, 在解决小样本、非线性及高维模式识别中表现出许多特有的优势, 但 SVM 的参数选择没有理论上的指导, 是一个亟待解决的问题. 蚁群优化算法(Ant Colony Optimization, ACO)[9]是 Dorigo 等人于 20 世纪 90 年代提出的, 是一种模拟蚂蚁种群进化的启发式搜索算法, 具有随机搜索、全局优化、分布式和正反馈等优点.

为了提高软件缺陷预测模型的准确性, 本文提出了一种基于 ACO-SVM 的软件缺陷预测模型. 即分别利用 SVM 和 ACO 建立和优化软件缺陷预测模型, 并基于主成分分析方法进行降维处理, 最后利用 ACO-SVM 建立软件缺陷预测模型.

2　相关工作

2.1　支持向量机

支持向量机的基本思想是求解核函数和二次规划问题, 通过核函数将数据映射到高维特征空间来解决非线性可分的问题. 径向基核函数(Radial Basis Function, RBF)具有较宽的收敛范围, 是较理想的分类依据函数, 其表达式如式(1)所示.

$$\exp(-\sigma\|\boldsymbol{x}-\boldsymbol{x}'\|^2) \tag{1}$$

其中, $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 是特征向量, σ 是径向基核函数的带宽.

在二分类问题中, 假设给定训练集 $T=\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l)\}\in(R^n\times y)^l$, 其中 $\boldsymbol{x}_i\in R^n$, $y_i\in Y=\{+1,-1\}$, $i=1,2,\cdots,l$, $\boldsymbol{x}_i$ 是特征向量, y_i 是类别标签. SVM 将决策函数的求解转化为带有约束条件的二次规划问题求解, 如式(2)所示, 其中 α_i、α_j 是拉格朗日乘子, $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 是核函数.

$$\begin{aligned}&\min_{\alpha}\ \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_jK(\boldsymbol{x}_i,\boldsymbol{x}_j)\alpha_i\alpha_j-\sum_{j=1}^{l}\alpha_j\\&\text{s.t.}\ \sum_{i=1}^{l}y_i\alpha_j=0,\ 0\leq\alpha_i\leq C,\ i=1,2,\cdots,l\end{aligned} \tag{2}$$

方程式的解为 $\alpha^*=(\alpha_1^*,\cdots,\alpha_l^*)^{\mathrm{T}}$, 决策函数的偏置项 b^* 如式(3)所示.

$$b^*=y_j-\sum_{i=1}^{l}y_ia_i^*K(\boldsymbol{x}_i,y_j) \tag{3}$$

决策函数如式(4)所示.

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{l}y_i\alpha_i^*K(\boldsymbol{x}_i,\boldsymbol{x})+b^*\right) \tag{4}$$

2.2　蚁群优化算法

基本蚁群算法引进了正反馈和并行机制, 较好地解决了旅行商(Traveling Salesman Problem, TSP)问题, 但是收敛速度较慢. 1997 年 Dorigo 等人在基本蚁群算法的基础上提出了蚁群系统, 证明了该算法优于模拟退火、进化计算等仿生算法.

蚁群系统的搜索过程如下:

1. 蚂蚁的配置和信息素的初始化.

2. 对每只蚂蚁按式(5)或式(6)选择下一个城市, 并基于式(7)、(8)更新该路径上的信息素.

$$p_{ij}^k(t+1)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}(\eta_{ij})^{\beta}}{\sum\limits_{k\notin tabu_k}[\tau_{ik}(t)]^{\alpha}(\eta_{ik})^{\beta}}, & j\notin tabu_k\\ 0, & \text{其它}\end{cases} \tag{5}$$

$$j=\begin{cases}\arg\max\limits_{j\in J_k(i)}\{[\tau_{ij}(t)]^{\alpha}(\eta_{ij})^{\beta}\}, & q\leq q_0\\ S, & \text{其它}\end{cases} \tag{6}$$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}^k \tag{7}$$

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{l_{jb}}, & (i,j)\in T^k\\ 0, & \text{其它}\end{cases} \tag{8}$$

其中, $q_0\in[0,1]$ 是初始设定的参数, q 是一个随机数, $\tau_{ij}(t)$ 是 t 时刻在城市节点 i、j 之间路径上残留的信息素, η_{ij} 是城市 i 转移到城市 j 的启发式信息, α 用来衡量残留信息素的相对重要程度, β 用来衡量期望信息的相对重要程度, S 是根据式(5)决定的随机变量, $tabu_k$ 是蚂蚁 k 当前走过的城市(禁忌表), l_{jb} 是蚂蚁 k 从开始城市到当前城市已走过的路径长度, Q 为常数, T^k 为蚂蚁 k 经过的路径.

3. 当所有蚂蚁走完所有城市后, 通过计算它们的最短距离获得最佳路径. 按下式更新最佳路径上的信息素.

$$\tau_{ij}^{\mathrm{new}}=(1-\alpha)\tau_{ij}^{\mathrm{old}}+\alpha\Delta\tau_{ij}^k \tag{9}$$

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{1}{l_k}, & (i,j)\in O^k\\ 0, & \text{其它}\end{cases} \tag{10}$$

其中，α 为全信息素挥发系数，l_k 为最佳路径的长度，O^k 表示蚂蚁 k 所走的是最佳路径.

4. 输出最优解.

2.3 主成分分析算法

主成分分析算法(PCA)是由 Hotelling 提出的.主成分分析将给定的一组相关变量通过线性变换转换为另一组不相关的变量，这些新的变量按照方差递减的顺序排列，构成相应的主成分.

PCA 的步骤如下：

1. 设 $\boldsymbol{X}=(X_1, X_2, \cdots, X_n)^{\mathrm{T}}$ 是一个 n 维随机向量，计算 $\boldsymbol{X}$ 的协方差矩阵 $\Sigma=\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$.

2. 计算 Σ 的特征值 $\lambda_i(\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n)$ 及其特征值对应的特征向量 $\boldsymbol{p}_i(i=1, 2, \cdots, n)$，则 $\boldsymbol{X}$ 的第 i 主成分为 $Y_i=\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{X}$.

3. 定义 $\lambda_k \Big\backslash \sum_{i=1}^{n}\lambda_i$ 为主成分 Y_k 的贡献率，定义 $a=\sum_{i=1}^{m}\lambda_i \Big\backslash \sum_{i=1}^{n}\lambda_i$ 为主成分的累积贡献率.主成分的个数由 a 决定，一般取 $a>95\%$.

3 软件缺陷预测模型的建立

3.1 软件缺陷预测模型

基于 ACO-SVM 和 PCA 方法建立软件缺陷预测模型主要包括以下几个步骤，如图 1 所示.

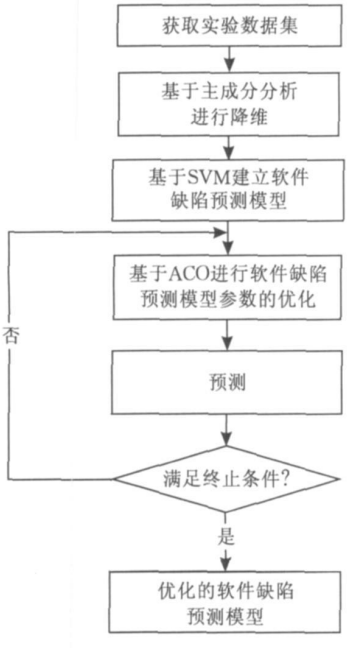


图 1 建立软件缺陷预测模型的流程

建立软件缺陷预测模型步骤如下：

1. 获取软件复杂度度量数据集；

2. 对数据集进行主成分分析，降低数据集维数；

3. 用训练样本基于 SVM 建立软件缺陷预测模型，核函数选择 RBF 核函数；

4. 利用 ACO 进行模型参数 C 和 σ 的优化；

5. 用测试样本进行程序模块的分类预测.如果预测结果满足终止条件，则得到优化的软件缺陷预测模型并结束；否则返回步 4 继续优化模型.

其中，终止条件为模型预测准确率达到了事先给定的阈值或者循环次数超过了事先设定的最大循环次数.

3.2 软件缺陷预测模型参数的优化

本文利用 ACO 算法进行模型参数 C 和 σ 的优化.即根据 ACO 算法搜索出的最优路径来确定模型参数.在参数优化过程中，ACO 算法不是根据路径的长度来更新路径上的信息素，而是根据系统的性能指标(分类准确率)来更新信息素.为了利用 ACO 算法进行参数优化，需要先将模型的参数 C 和 σ 离散化，根据经验，C 和 σ 的有效位个数均取 5 位，各个有效位上的值取[0, 9] 之间的整数.参数 C 的最高位是百位，其取值范围是(0, 999. 99] .参数 σ 最高位是个位，所以其取值范围是(0, 9. 9999] .ACO 算法以 10 个有效位作为行、以 10 个整数值(0～9)作为列，在直角坐标系中，画出 10×10 的平面结构图作为模型参数优化蚂蚁运行图.图中城市节点的横坐标是有效位，纵坐标是该有效位上的取值.每只蚂蚁走完一条路径后，根据小数点的位置可以计算出两个参数的值，根据最优路径就能计算出模型的最优参数.

设启发式信息 $\eta_{ij}=1$，用分类准确率评估支持向量机性能，在全局信息更新过程中，令 Q 是信息素强度，$\Delta\tau_{ij}=Q\cdot Accuracy$，$Accuracy$ 是每次循环中最高的分类准确率，执行过程如下：

1. 使用参数离散化方法离散化 C 和 σ；

2. 初始化 $\tau_{ij}=1$，信息素增量 $\Delta\tau_{ij}=0$；

3. 第一次执行以下搜索过程寻找最优路径：

3 1. 将所有蚂蚁置于坐标系原点；

3 2. 随机地将蚂蚁放置到下一个需要访问的城市节点，且该节点的横坐标 x 不同于以前访问过的城市节点的横坐标；

3 3. 对每只蚂蚁根据式(7)、(8)更新转移路径上的信息素；

3 4. 如果所有蚂蚁完成了路径上的所有城市节点的访问，那么根据式(9)、(10)对最优蚂蚁走过的路径进行信息素更新.否则转到步 3 2.

4. 再次将所有蚂蚁放到坐标系原点；

5. 根据式(5)、(6)使蚂蚁移到下一个城市；

6. 根据式(7)、(8)更新每只蚂蚁转移路径上的信息素.如果蚂蚁完成所有城市的访问，转到步 7；否则转到步 5；

7. 使用蚂蚁移动过程得到的 C 和 σ 训练 SVM. 找到具有最高分类准确率的最优蚂蚁，根据式(9)、(10)更新其所走路径上的信息素. 如果分类准确率达到终止条件或者循环次数超过最大循环次数，则转到步 8，否则转到步 4；

8. 输出 C、σ 和最高分类准确率.

4　基于 ACO-SVM 的软件缺陷预测模型实验

4.1　实验数据

实验数据是美国国家航空航天局(NASA)提供的一个数据包①. 该数据包共有 13 个数据集，如表 1 所示.

表 1　实验数据

数据集	模块数	维数	数据集	模块数	维数
JM1	10878	23	PC1	1107	42
KC1	2107	26	PC2	5589	42
KC3	458	42	PC3	1273	42
KC4	125	42	PC4	3840	42
MC1	9466	41	PC5	17186	42
MC2	161	42	CM1	505	40
MW1	403	42			

本文根据 ERROR_COUNT 属性对各数据集中各模块是否有缺陷进行分类，设每个数据集中 ERROR_COUNT ≥1 的模块为有缺陷模块，否则为无缺陷模块. 为了缩短分类时间，针对模块数比较多的数据集，我们在此数据集中随机选取 200 个模块作为训练样本，选取 100 个模块作为预测样本. 每个数据集选取 10 维数据作为分类度量.

4.2　实验结果与分析

为了验证模型的预测能力，本文采用十折交叉验证(10-fold CV)进行实验. 即将每一个数据集平均分成 10 份，轮流将其中 9 份做训练，剩余的 1 份做预测，取 10 次预测结果的均值作为对算法预测能力的评价.

本文采用准确度、查准率、查全率和 $F1$ 值[10] 来评价模型的预测能力. 这些度量来自表 2 所示的交叉矩阵.

表 2　交叉矩阵

实际值	预测值	
	容易出错模块	不易出错模块
容易出错模块	正确的正例(TP)	错误的负例(FN)
不易出错模块	错误的正例(FP)	正确的负例(TN)

实际正例个数 $P=TP+FN$，实际负例个数 $N=FP+TN$，实例总数 $C=P+N$.

模型评价指标的定义如下：

准确度($accuracy$)表示正确分类的测试实例的个数占测试实例总数的比例，计算公式如下：

$$accuracy=\frac{TP+TN}{C} \tag{11}$$

查准率($precision$)表示正确分类的正例个数占分类为正例的实例个数的比例，计算公式如下：

$$precision=\frac{TP}{TP+FP} \tag{12}$$

查全率($recall$)表示正确分类的正例个数占实际正例个数的比例，计算公式如下：

$$recall=\frac{TP}{P} \tag{13}$$

$F1$ 表示查全率与查准率的调和平均，计算公式如下：

$$F1=\frac{2}{\frac{1}{precision}+\frac{1}{recall}} \tag{14}$$

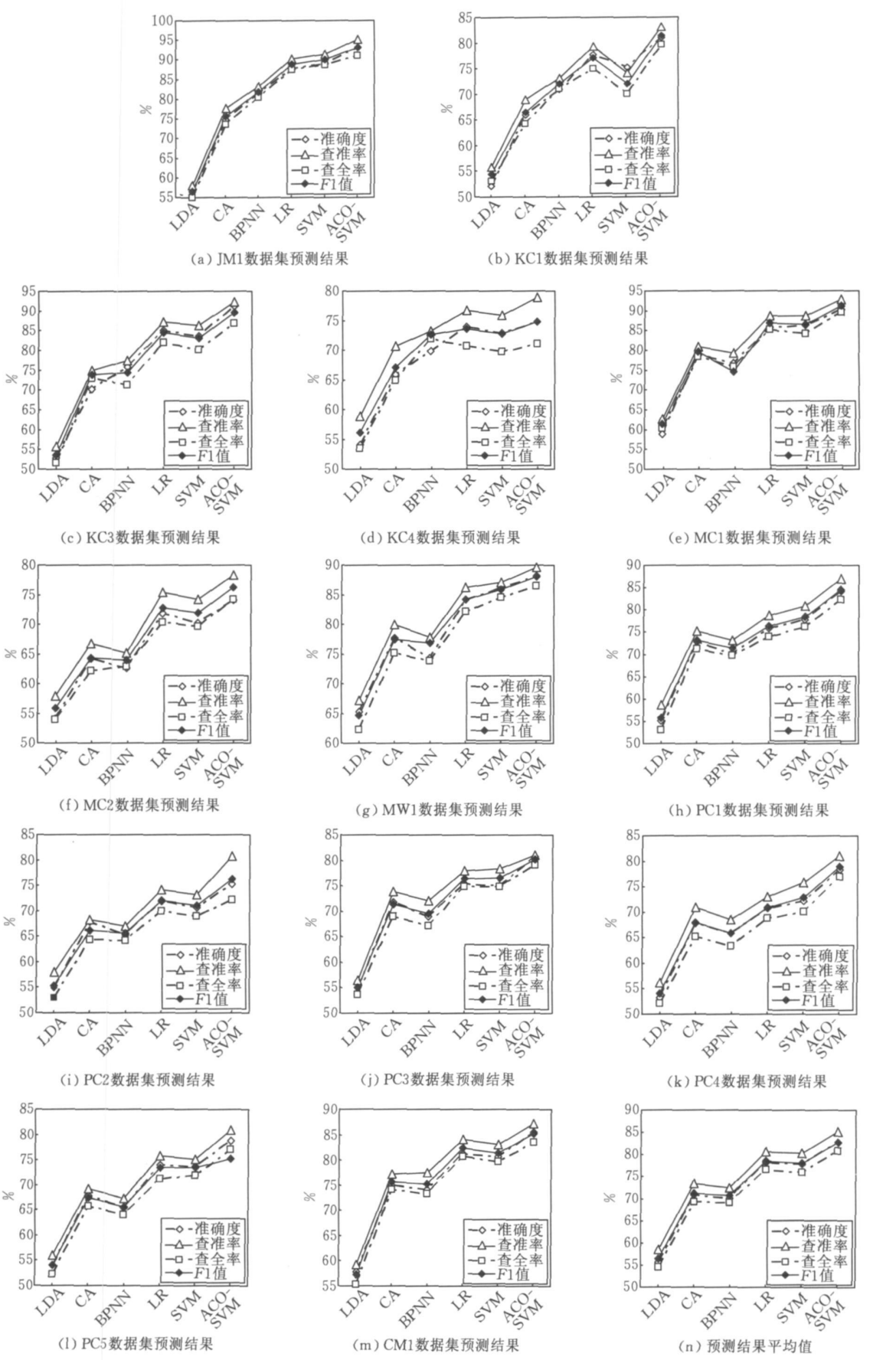
将基于本文方法(ACO-SVM)和 Fisher 线性判别分析(LDA)、聚类分析(CA)、BP 神经网络(BPNN)、逻辑回归(LR)、支持向量机(SVM)等方法建立的软件缺陷预测模型进行比较. 基于十折交叉验证(10-fold CV)分别对 13 个数据集进行实验，计算 13 组评价指标(准确度、查准率、查全率、$F1$ 值)，本文方法在预测过程中的最优参数选择如表 3 所示，预测结果如图 2 所示.

表 3　最优参数选择

数据集	C	σ	数据集	C	σ
JM1	756 47	0 0040	PC1	711 72	0 0040
KC1	195 41	0 0086	PC2	676 79	0 0050
KC3	20 04	0 0054	PC3	4 09	0 0078
KC4	691 16	0 0020	PC4	90 99	0 0070
MC1	594 69	0 0054	PC5	248 03	0 0029
MC2	260 95	0 0028	CM1	440 82	0 0043
MW1	663 06	0 0076			

图 2(a)～(m)分别为本文方法与传统方法在 13 个数据集上的预测结果对比，(n)为 13 个数据集预测结果的平均值对比. 从图 2 可以看出：基于 LDA 建立的模型的预测结果较差，这是由于该方法基于线性分类函数进行分类，而软件复杂性度量数据之间呈非线性关系；基于 CA、BPNN 建立的模型的预测结果明显优于基于 LDA 方法，但是由于 CA 方法需要较多的先验知识，BPNN 网络结构选择尚无一种确定而有效的方法，因此这两种方法的泛化能力不十分理想；基于LR建立的模型预测结果较

① http://mdp.ivv.nasa.gov/



(a) JM1数据集预测结果 (b) KC1数据集预测结果

(c) KC3数据集预测结果 (d) KC4数据集预测结果 (e) MC1数据集预测结果

(f) MC2数据集预测结果 (g) MW1数据集预测结果 (h) PC1数据集预测结果

(i) PC2数据集预测结果 (j) PC3数据集预测结果 (k) PC4数据集预测结果

(l) PC5数据集预测结果 (m) CM1数据集预测结果 (n) 预测结果平均值

图 2 本文方法和传统方法的比较

好，但是该模型受到样本容量的局限，只有样本容量较大时该模型才能取得较好的预测效果；基于 SVM 建立的预测模型的参数选择没有理论指导，只能根据经验选取，故预测能力不稳定；本文方法利用 SVM 在小样本情况下的预测优势，利用 ACO 对模型参数进行优化，并利用 PCA 缩减特征空间以减少计算成本，使得模型的各项评价指标均优于 LDA、CA、NN、LR、SVM 模型. 但是由于在 ACO 参数寻优的过程中需要多次训练样本，计算成本仍然需要进一步减少.

5　结　论

本文的贡献是针对软件缺陷预测问题提出了一种新颖的基于 ACO-SVM 的软件缺陷预测模型，其基本思想是基于 PCA 缩减特征空间、基于 ACO-SVM 建立和优化软件缺陷预测模型. 实验结果表明，该模型比传统方法具有更好的预测效果. 但是该方法在参数寻优过程中需要较长的时间，如何进一步降低模型的运行时间和提高模型的预测准确率，是今后的课题.

参　考　文　献

[1] Challagulla V U B, Bastani F B, I-Ling Yen, Paul R A. Empirical assessment of machine learning based software defect prediction techniques//Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems. Washington, DC, USA, 2005: 263-270

[2] Lyu Michael R. Handbook of Software Reliability Engineering. New York: IEEE Computer Society Press and McGraw-Hill Book Company, 1996

[3] Khoshgoftaar Taghi M, Seliya Naeen. Tree-based software quality estimation models for fault prediction//Proceedings of the 8th International Symposium on Software Metrics. Washington, DC, USA, 2002: 123-128

[4] Stich Timothy Janes, Spoerre Julie K, Velasco Tomas. The application of artificial neural networks to monitoring and control of an induction hardening process. Journal of Industrial Technology, 2000, 16(1): 1-11

[5] Ohlsson Niclas, Alberg Hans. Predicting fault-prone software modules in telephone switches. IEEE Transactions on Software Engineering, 1996, 22(12): 886-894

[6] Khoshgoftaar Taghi M, Seliya Naeem. Software quantity classification modeling using the SPRINT decision tree algorithm//Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence. Washington, DC, USA, 2002: 365-367

[7] Briand L C, Melo W L, Wust J. Assessing the applicability of fault-proneness models across object-oriented software projects. IEEE Transactions on Software Engineering, 2002, 28(7): 706-720

[8] Cortes Corinna, Vapnik Vladimir. Support-vector networks. Machine Learning, 1995, 20(3): 273-297

[9] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66

[10] Wang X H, Shu P, Cao L et al. A ROC curve method for performance evaluation of support vector machine with optimization strategy. Computer Science Technology and Applications, 2009, (2): 117-120

JIANG Hui-Yan, born in 1963, Ph. D., professor. Her main research interests include medical image computer-aided diagnosis, 3D visualization, image processing, model recognition.

ZONG Mao, born in 1987, master. His main research interests focus on image segmentation.

LIU Xiang-Ying, born in 1986, master. His main research interests focus on pattern recognition.

Background

With the extensive application of computer software, the scales of software systems are more and more large and complex, so it is difficult to ensure the reliability of software systems. In some critical applications of software systems such as aviation, aerospace, the reliability of software system is particularly important. Software defect prediction model can predict which modules have a tendency to being wrong in the early stage of software development, so we can find the appropriate solution timely. Software defect prediction plays an important part in promoting software defect.

From 1970s, there are many software defect prediction models were put forward, which include time series analysis model, cluster analysis model, artificial neural network model, logistic regression model, support vector machine model and so on. Support vector machines (SVM) are a set of related supervised learning methods that can analyze data and recognize patterns, used for classification and regression analysis. Because the support vector machine model has the advantages of global optimization, short training time, good generalization performance, the support vector machine model is discussed extensively. However, the parameters of SVM are difficulty. Without much experience, we almost can not choose optimal parameters. Ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. Meanwhile, the ant colony optimization algorithm is widely used for finding the optimal parameters. In this article, this paper puts forward a software defect prediction model based on support vector machine. It uses ant colony optimization algorithm to choose the optimal parameters for support vector machine automatically to solve the problem that the parameters of the support vector machine is hard to choose artificially, and also uses the PCA method to reduce the dimensions of software defect metrics to increase the computation speed.