基于LASSO-LARS的软件复杂性度量属性特征选择研究

周雁舟[1] 乔 辉[1] 吴晓萍[2] 邵 楠[1] 惠文涛[1]
(中国人民解放军信息工程大学密码工程学院 郑州 450004)[1]
(兰州大学数学与统计学院 兰州 730000)[2]

摘 要 针对软件可靠性早期预测中软件复杂性度量属性维数灾难问题,提出了一种基于最小绝对值压缩与选择方法(The Least Absolute Shrinkage and Select Operator,LASSO)和最小角回归(Least Angle Regression,LARS)算法的软件复杂性度量属性特征选择方法。该方法筛选掉一些对早期预测结果影响较小的软件复杂性度量属性,得到与早期预测关系最为密切的关键属性子集。首先分析了LASSO回归方法的特点及其在特征选择中的应用,然后对LARS算法进行了修正,使其可以解决LASSO方法所涉及的问题,得到相关的复杂性度量属性子集。最后结合学习向量量化(Learning Vector Quantization,LVQ)神经网络进行软件可靠性早期预测,并基于十折交叉方法进行实验。通过与传统特征选择方法相比较,证明所提方法可以显著提高软件可靠性早期预测精度。

关键词 软件可靠性早期预测,特征选择,LASSO回归方法,LARS算法,LVQ神经网络
中图法分类号 TP311.5 **文献标识码** A

Research of Software Complexity Metric Attributes Feature Selection Based on LASSO-LARS

ZHOU Yan-zhou[1] QIAO Hui[1] WU Xiao-ping[2] SHAO Nan[1] HUI Wen-tao[1]
(Institute of Cryptography Engineering, PLA Information Engineering University, Zhengzhou 450004, China)[1]
(School of Statistics and Mathematics, Lanzhou University, Lanzhou 730000, China)[2]

Abstract To cope with the software complexity metric attributes dimension disaster which exists in the software reliability early prediction, this paper put forward a software complexity metric attribute feature selection method based on Least Absolute Shrinkage and Selection Operator(LASSO) method and the Least Angle Regression(LARS) algorithm. This method can filter out some software complexity metric attributes which have smaller influence on the early prediction results and can obtain the key attributes subsets associated most closely with the prediction result. This paper firstly analyzed the characteristics of LASSO regression method and its application in feature selection, secondly modified the LARS algorithm so that it can be used to solve the problems which LASSO method involves and get relevant complexity metric attribute subsets, lastly combined with the Learning Vector Quantization(LVQ) neural network to carry on the early software reliability prediction experiment. During the experiment, the authors used the 10-fold experiment methods. The experiment results indicate that the method can improve early prediction accuracy of software reliability.

Keywords Software reliability early prediction, Feature selection, LASSO regression method, LARS algorithm, LVQ neural network

1 引言

目前国内外学界提出了很多种不同的软件可靠性早期预测模型。在现有的各种模型中,基于分类的模型已经被证明可以更好地实现软件可靠性早期预测[1]。然而由于软件模块的复杂性,软件复杂性度量和易出错模块类别的对应关系很难确定,研究人员提出了采用大量的原始复杂性度量属性数据来进行软件可靠性早期预测。然而不加选择地引入所有的复杂性度量属性来进行可靠性早期预测,往往不能取得良好的效果[2]。这是因为,过多地引入复杂性度量属性会导致"维数灾难"和"组合爆炸"[3],使得计算量空前增大,估计和预测精度也会下降;此外,在一些情况下,获得某些软件复杂性度量属性数据代价昂贵,如果这些数据本身对预测结果的影响微乎其微,那么势必造成预测数据的收集和模型应用的费用不必要的增大。

面对上述问题,特征选择[4,5]提出了很好的解决方案。它可以显著降低复杂性度量属性空间的维数,减少分类训练时间,提高预测效率。在具体的应用中,特征选择应该具备如下要求:

a)可解释性,即模型中选择的特征具有科学意义;

到稿日期:2013-01-16 返修日期:2013-04-30 本文受国家863项目计划(2008AA01Z404)资助。
周雁舟(1971—),男,副研究员,硕士生导师,CCF初级会员,主要研究方向为可信计算、操作系统安全及系统工程,E-mail:sparlk@126.com;乔 辉(1988—),男,硕士,主要研究方向为软件可靠性预测;吴晓萍(1987—),女,硕士,主要研究方向为金融统计;邵 楠(1988—),男,硕士,主要研究方向为软件测试;惠文涛(1990—),男,硕士,主要研究方向为身份认证。

b)稳定性；

c)尽量避免在假设检验中出现误差；

d)尽量控制计算的复杂度。

针对上述要求，本文引入一种新的特征选择方法，即基于最小绝对值压缩与选择方法和最小角回归算法(LASSO-LARS)方法，利用它对原始的软件复杂性度量属性集进行降维，之后用得到的属性子集来实现软件可靠性早期预测。主要工作如下：

1)分析了LASSO方法的核心思想及如何使用它来得到精简的复杂性度量属性子集；

2)分析了LARS算法的核心思想并对它进行了一定的调整来解决LASSO问题；

3)将使用上述方法得到的属性精简子集输入到基于LVQ神经网络算法的软件可靠性早期预测模型中，并与其他传统特征选择方法进行对比，验证了该方法的有效性和预测准确率。

2 相关研究

2.1 LVQ神经网络

LVQ神经网络的基本思想是，对于来自训练集的任一模式向量 X，如果 X 与最近的模板属于同一类，则无需学习；否则，将“惩罚”分类错误的模板，“奖励”其对应正确类别的模板。若经过若干次迭代后，所得到的向量量化不再明显变化，则实现模式识别，因此简单易行。其LVQ结构如图1所示。

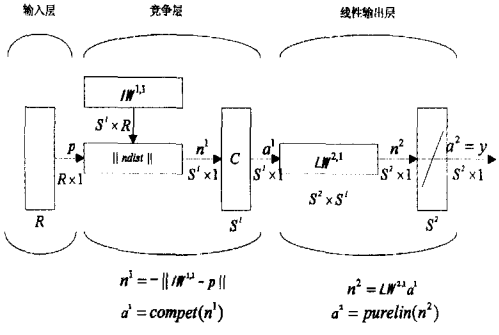

图1 LVQ神经网络结构图

图1中，p 为 R 维的输入模式；S^1 为竞争层神经元的个数；$IW^{1,1}$ 为输入层与竞争层之间的连接权系数矩阵；n^1 为竞争层神经元的输入；a^1 为竞争层神经元的输出；$LW^{2,1}$ 为竞争层与线性输出层之间的连接系数矩阵；n^2 为线性输出层神经元的输入；a^2 为线性输出层神经元的输出。竞争层和线性层的每一个神经元的输出都对应一个分类(子分类或目标分类)结果，所以竞争层通过学习可以得到 S^1 类子分类结果；然后，线性层将 S^1 类子分类结果再分成 S^2 类目标分类结果(S^1 始终大于 S^2)。

2.2 自适应遗传算法

20世纪80年代，Holland教授实现了第一个基于遗传算法的机器学习系统，开创了遗传算法的机器学习新概念。遗传算法利用生物遗传学的观点，在解空间随机产生多个起始点并同时开始搜索，由适应度函数来指导搜索方向，是一种能够在复杂搜索空间快速寻求全局优化解的搜索技术，目前已在优化、机器学习和并行处理等领域得到越来越广泛的应用。

简单遗传算法存在着寻优时稳定性不高、局部搜索能力不强、收敛速度较慢而且容易产生早熟现象等缺陷。本文利用自适应遗传算法[6]完成相关的LVQ神经网络参数寻优工作，使得种群进化更加稳定，避免陷入局部最优。

2.3 软件可靠性早期预测模型

基于AGA-LVQ算法建立的软件可靠性早期预测模型[7]主要包括以下几个步骤，如图2所示。

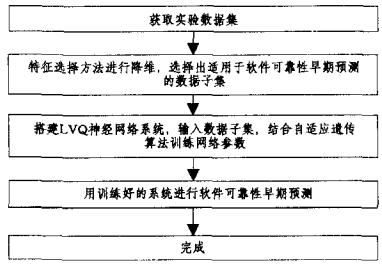

图2 建立软件可靠性早期预测模型的流程

其具体步骤如下：

1. 获取软件复杂性度量属性数据集；

2. 消除数据集中存在的错误数据，然后运用特征选择方法来降维，获得精简的属性子集；

3. 搭建LVQ神经网络系统框架；

4. 用自适应遗传算法结合特征选择获得的属性子集训练LVQ神经网络相关参数；

5. 将测试样本输入训练好的LVQ神经网络系统，来进行软件可靠性早期预测；

6. 实验结束，输出软件可靠性分类结果。

3 LASSO-LARS方法

3.1 LASSO方法

特征选择方法的核心在于我们要降低观察变量的维数，剔除影响因子较小的观察变量，其实质就是加一定的约束条件，将影响因子较小的观察变量的回归系数置为零。加的约束条件不同，就构成了不同的特征选择的方法。本文介绍的LASSO方法就是在目标函数是二次型范数的基础上加上绝对值约束条件构成的。

Tibshirani(1996)提出了一种用于处理高维数据特征选择问题的有偏估计的方法，该方法称为Lasso[8]。LASSO可以通过惩罚函数将影响较小的观察变量的回归系数收缩到0，这样做虽然牺牲了一定的估计偏差，但是能降低预测的方差从而提高预测的精确性。

使用LASSO方法进行软件可靠性早期预测时首先需要对复杂性度量属性数据集进行中心标准化处理，以消除不同指标量纲的影响，使其满足如下要求：

$$\sum_{i=1}^{n} y_i=0,\ \sum_{i=1}^{n} x_{ij}=0,\ \sum_{i=1}^{n} x_{ij}^2=1,\ j=1,\cdots,m \tag{1}$$

式中，m 为维数，n 为变量数。具体到软件可靠性早期预测中，则 m 为软件复杂性度量数据集中各模块的属性数目，如可将HALSTEAD度量[9]、McCabe度量[10]和C&K[11]度量等作为维数，而 n 则为数据集中的模块数。假设一个数据集由458个模块数组成，每个模块均拥有42个复杂性度量属性，则 $m=42$，$n=458$。

令平方差和：

$$S(\hat{\beta})=\|Y-\hat{\mu}\|^2=\sum_{i=1}^{n}(y_i-\hat{\mu}_i)^2,\ \hat{\mu}=X\hat{\beta}=x_{ij}\hat{\beta}_j \tag{2}$$

式中，y_i 是响应变量；$x_i=(x_{i1},x_{i2},\cdots,x_{in})$是观察变量；$\hat{\beta}_j$ 为第 j 个变量的回归系数。

$\hat{\beta}$的绝对值和：

$$T(\hat{\beta})=\sum_{j=1}^{m}|\hat{\beta}_j|\leqslant t \tag{3}$$

LASSO 方法即为在 $T(\hat{\beta})\leqslant t$ 的约束条件下，通过调整各变量的回归系数即$\hat{\beta}_j$的值来求响应变量 y_i 与回归变量$\hat{\mu}$的最小平方差和，即求 $S(\hat{\beta})$的最小值。

LASSO 方法在基于分类的软件可靠性早期预测模型中，y_i 为 0-1 属性变量，用来反映数据集中的模块是否为有缺陷模型，而 x_i 则为其他的复杂性度量属性变量。

通过对回归系数添加约束条件(3)，可以将与响应变量相关度较低的观察变量的回归系数$\hat{\beta}_j$置为 0，从而在特征选择时，将这些回归系数为 0 的变量去除，只保留强相关性的变量。t 的取值可以为 0～∞。当 t 的值比较小时，某些相关度低的变量系数就被压缩为 0，从而将这些变量删除，以达到特征选择的目的；当 t 取得足够大时，将不再具有约束的作用，此时所有的属性都将被选择并且形成一个变量选择序列。

图 3 为 LASSO 对数据集 Cleve 的变量选择序列分析[12]。

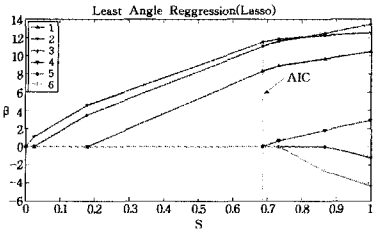

图 3 LASSO 对 Cleve 的特征序列图

3.2 LARS 算法

虽然可以采用向前回归解决 LASSO 方法的求解问题，但是效果一直不令人满意，直到 Efron 等人于 2004 年提出 LARS 算法[13]。LARS 算法是每次在回归残差的基础上选择新的变量，该过程是一个残差不断减小的过程，具有很高的计算效率。回归残差综合了类标签变量与已选变量的信息，将该算法引入 LASSO 问题中，能高效地求得最优解，即所要求的变量选择序列。

LARS 算法的基本思想是：首先选择一个与响应变量相关性最大的观察变量，然后沿这个方向走一定的长度，直到出现第二个观察变量。这两个观察变量与残差的相关性相同，就沿着这两个变量等角度的方向继续走，以此类推，选择出需要的观察变量。

算法步骤描述如下：

Input：数据集 D

Output：与类标签关联性强的特征序列

1. 将所有的回归系数$\hat{\beta}_j$设置为 0，然后找到与响应变量 y_i 最相关的一个观察变量，记为 x_{j1}；

2. 将$\hat{\beta}_j$沿着此观察变量的方向最大限度地增大，直到找到另一个与当前残差 $r=y-\hat{\mu}_i$ 最相关的观察变量被选中为止，将其记为 x_{j2}；

3. 将$\hat{\beta}_j$朝着与(x_{j1},x_{j2})等角度的方向增大，直到出现第 3 个观察变量 x_{j3}符合与当前残差最相关这一条件；

4. 之后一直沿着与(x_{j1},x_{j2},x_{j3})等角度的方向增大，直到出现第 4 个观察变量，以此类推。

将上述算法应用到软件复杂性度量属性集中，可以得到一个特征序列。随着约束值 t 的变化，可以从路径中移除一些不重要的属性，从而达到降低复杂性度量属性集维数的目的。从上述算法步骤中不难看出，LARS 的核心在于等角度方向的选择。下面是对等角度方向选择的介绍。

假设第 k 步时，前 k 个观察变量被选择进来，记它们的集合为 A。由前 $k-1$ 步得到的回归变量为 μ_{k-1}，定义矩阵

$$X_A=(S_1X_1,S_2X_2,\cdots,S_jX_j,\cdots)_{j\in A} \tag{4}$$

式中，$S_j=\text{sign}((Y-\mu_{k-1})'X_j)$。记 $G_A=X_A'X_A$，$A_A=(1_A'G_A^{-1}1_A)^{-1/2}$，则下一步等角度向量定义为：

$$u_A=X_Aw_A,w_A=A_AG_A^{-1}1_A \tag{5}$$

可以验证，它满足

$$X_A'u_A=A_A1_A,(u_A)^2=1 \tag{6}$$

因此，u_A 是一个与所有已选入自变量方向成相同夹角的方向，在该方向上前进会导致残差在各自变量方向与各回归系数变量内积等量递减。

将等角度方向引入前述 LARS 的步骤，得到 LARS 算法的具体实现步骤如下：

1)初始化 $k=0,A_0=\phi,\mu_0=0$；

2)$k=k+1$，设 $\hat{c}_j=X_j'(y-\mu_{k-1})$，$\hat{C}_{\max}=\max_j\{|\hat{c}_j|\}$，$A=\{j:|\hat{c}_j|=\hat{C}_{\max}\}$，对于所有的 $j\in A$，有 $s_j=\text{sign}\{\hat{c}_j\}$；

3)计算 X_A,A_A,u_A 和 $a_j=X_A'u_A$；

4)更新 $\mu_k=\mu_{k-1}+\hat{\gamma}_ku_k$，其中

$$\hat{\gamma}_k=\min_j{}^{+}\left(\frac{\hat{C}_{\max}-\hat{c}_j}{A_A-a_j},\frac{\hat{C}_{\max}+\hat{c}_j}{A_A+a_j}\right) \tag{7}$$

5)重复 2)－3)，直到选择足够的变量个数。

3.3 LASSO 与 LARS 的关系

在求解 LASSO 问题的时候，我们需要保证回归系数的符号与目前相关性的符号相同，也就是：

$$\text{sign}(\hat{\beta}_j)=\text{sign}(\hat{c}_j)=S_j \tag{8}$$

对于 $j\in A$，令

$$\hat{d}_j=s_jw_j,\beta_j(\gamma)=\hat{\beta}_j+\gamma\hat{d}_j \tag{9}$$

则 $\beta_j(\gamma)$在 $\gamma_j=-\frac{\hat{\beta}_j}{\hat{d}_j}$时，符号发生改变。所以，第一次变号发生在$\tilde{\gamma}=\min_{\gamma_j>0}\{\gamma_j\}$时。

为满足式(8)要求，需要对 LARS 算法进行下面的修正：

倘若 $\tilde{\gamma}<\hat{\gamma}$，在 $\gamma=\hat{\gamma}$时，停止正在进行的 LARS 步骤，重新计算等角度方向并且将变量 $\tilde{j}$ 剔除。

根据 LASSO-LARS 方法，可以得到每一个 t 对应的回归系数 β_j，这样我们可以画出回归系数 β_j 随 t 变化的走势图。根据残差图可以定出不同的观察变量对响应变量的影响大小，筛选掉影响较小的观察变量，得到较为精确的估计式。

4 实验分析

4.1 实验环境及实验数据

本文实验所用算法均利用 MATLAB2008a 软件编程实现，程序在配置为 Pentium(R)4 CPU 3.00GHz 2.99GHz，

504MB 内存的电脑上运行。

实验中软件复杂性度量数据来源于 NASA 的 MDP (Metric Data Program)软件度量项目中的 JM1 数据集，可从网址 http://mdp.ivv.nasa.gov 中下载。JM1 数据集来自于一个地面预测系统的工程，用 C++语言编写代码，包含了 10879 个子程序、21 个复杂性度量属性集。

4.2 实验数据预处理

NASA 的软件度量数据集已经被广泛应用于研究中，但很少有学者直接对数据集本身可能存在的问题进行研究。本文参考英国赫特福德大学计算机科学学院的 David Gray 教授的最新研究成果[14]，即他在文献中指出了 NASA 的数据集中存在着大量的“重复数据”、“矛盾数据”以及以“?”来描述的缺失数据，经过研究这些数据已经严重影响了数据的实验结果。

本文按照 David Gray 教授论文中提到的相关方法对 JM1 数据集中的“重复数据”和“矛盾数据”进行了删除；对于缺失数据，由于 JM1 数据集中数据量大且噪声数据较少，本文采用直接忽略样本的方法来处理。经过实验，最终将其子程序的个数删减为 8824 个。

之后又对样本数据进行中心标准化处理，使其满足 3.1 节中提到的 LASSO 方法应用的要求，消除了因量纲不同可能引发的影响。

4.3 LASSO-LARS 特征选择

结合第 3 节的内容，编程实现 LASSO-LARS 方法，实验结果如图 4 所示。从图中可以看出，有 5 条线不趋于 0，它们分别为 1,5,6,9,10 复杂性度量属性，对应的回归系数分别为：$\beta_1=0.003612$，$\beta_5=0.002503$，$\beta_6=-0.00039$，$\beta_9=0.001645$，$\beta_{10}=-3.2E-08$。因此，运用 LASSO-LARS 方法最终选择出的软件复杂性度量属性子集为{1,5,6,9,10}。

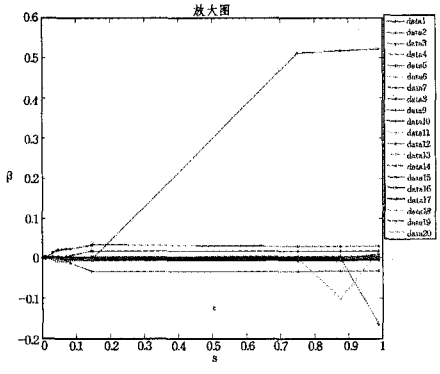

图 4 LASSO-LARS 实验图

4.4 软件可靠性早期预测

输入数据为 4.3 节选取的软件复杂性度量属性子集，结合自适应遗传算法编程来实现 LVQ 神经网络参数寻优。属性子集对应的观察变量数目过多(将近有 9000 多个变量)，若都用来进行参数寻优速度会很慢。本文运用伪随机数抽样方法[15]选取了 1000 个样本作为实验数据集，其中 700 个为训练样本，300 个为预测样本。

运用自适应遗传算法时种群大小设置为 40，进化代数设置为 200。实验结果如图 5 所示，获得最好的遗传算法的交叉概率为 $p_{c1}=0.82$，$p_{c2}=0.77$，变异概率为 $p_{m1}=0.04$，$p_{m2}=0.01$，获得的 LVQ 神经网络相关最优参数中期望误差为 0.24，训练速率为 0.12，隐含层神经元为 30 个。

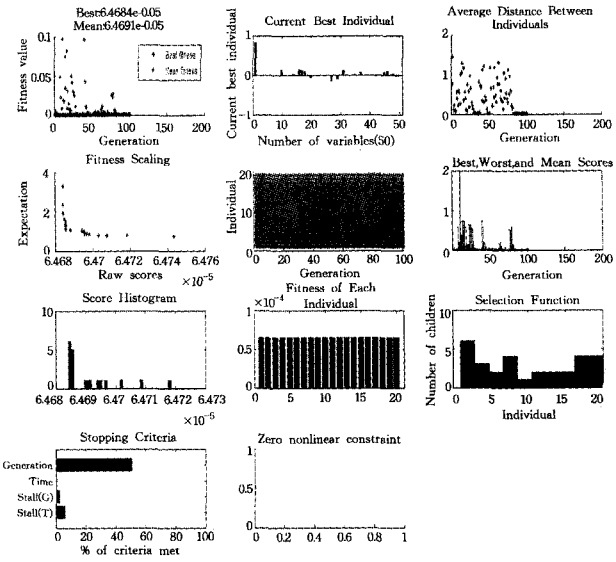

图 5 自适应遗传算法实验图

设置好 LVQ 神经网络参数后即进行软件可靠性早期预测实验。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 JM1 数据子集平均分成 10 份，轮流将其中 9 份做训练，剩余的 1 份做预测，将 10 次预测结果的均值作为对算法预测能力的评价。

本文实验所采用的评价指标为[16]：

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，公式为：

$$accuracy=\frac{TP+TN}{C} \tag{10}$$

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，公式为：

$$precision=\frac{TP}{TP+FP} \tag{11}$$

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，公式为：

$$recall=\frac{TP}{P} \tag{12}$$

F-度量(F-Measure)：是 Precision 和 Recall 的调和平均数。

$$F1=\frac{2}{\frac{1}{presion}+\frac{1}{recall}} \tag{13}$$

实验运行结果分别见图 6、图 7。

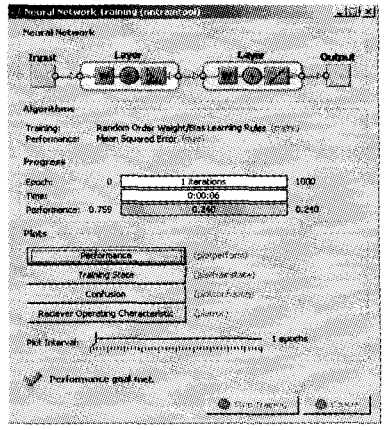

图 6 实验运行结果

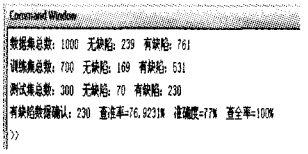

图 7 软件可靠性早期预测结果

4.5 实验结果与分析

将其他传统的特征选择方法如逐步回归(Stepwise Regression,SR)[17]、岭回归(Ridge Regression,RR)[18]和主成分回归(Principle Component Analysis,PCA)[19]等方法建立的精简属性子集输入到 2.3 节所提出的软件可靠性早期预测模型中,所获得的预测结果与 LASSO-LARS 方法进行比较,结果见表 1。

表 1 本文方法与传统方法的对比

预测模型	准确度(%)	查准率(%)	查全率(%)	F1-度量(%)
SR	42.7	45.1	47.6	46.3
RR	54.2	57.3	59.6	58.4
PCA	81.3	81.3	100	89.7
LASSO-LARS	77	76.9	100	86.9

从表 1 中可以看出,基于逐步回归方法建立的软件可靠性早期预测模型效果很差,这是因为逐步回归方法的核心是在引入新变量的同时对原有变量进行统计检验,删除影响不显著的变量来达到降维的目的,这样做很容易引起死循环,从而找不到最优的属性子集;基于岭回归方法的预测性能明显好于逐步回归方法,但岭回归方法只能将变量的回归系数无限压缩却无法趋于零,也就是说它无法降低属性集的维度;基于 PCA 方法的预测性能比较好,且所有指标均优于 LASSO-LARS,但该方法将复杂性度量属性转换或者组合生成了新的属性集,改变了被选属性的原始物理意义,这对理解数据的产生过程和软件可靠性早期预测模型的结构是非常不利的。

本文利用 LASSO-LARS 特征选择方法可以通过残差拟合将影响不显著变量的回归系数直接置为 0,减少了用于实验的复杂性度量属性集,取得了良好的软件可靠性早期预测效果。虽然准确度和查准率两项指标不是特别理想,但是查全率的指标是 100%,这表明它可以将正确分类的正例完全查找出来,这对软件测试过程中可靠性的改善是非常重要的。

本文提出的方法在预测准确率方面同主成分回归方法相比还存在着一定的不足。这主要是两方面的原因引起的,一方面是因为实验所用的 NASA 软件度量项目中的 JM1 软件复杂性度量属性集本身存在着一定的缺陷,而我们用自己的工具获得原始数据又面临着非常大的困难;另一方面是因为实验时 LASSO-LARS 方法中参数 t 的确定方法是人为设置的,这样不容易确定最优的 t 值。

结束语 本文的贡献是将一种新的特征选择方法引入到软件可靠性早期预测中。针对原始软件复杂性度量数据集中存在大量对分析结果影响甚微的复杂性度量属性集,本文用 LASSO-LARS 方法将大多数属性集的回归系数置为 0,仅保留了一小部分对分析结果影响巨大的属性子集。同时本文将该方法与基于 LVQ 神经网络的软件可靠性早期预测模型结合起来,并通过与其他传统的特征选择方法相对比,验证了该方法的有效性和优越性。

实验结果表明,该方法比传统特征选择方法具有更好的预测速度和预测效果,但其在预测准确度和查准率等方面仍有不足,这是下一步的研究课题。

参 考 文 献

[1] Vub C, Fb B, I-Ling Yen, et al. Empirical assessment of machine learning based software defect prediction techniques[C]// Proceedings of the 10thIEEE International Workshop on Object-Oriented Real-Time Dependable Systems. Washington, DC, USA, 2005:263-270

[2] 王琪. 软件质量预测模型中的若干关键问题研究[D]. 上海:上海交通大学,2006

[3] 崔正斌. 软件可靠性预测技术研究[D]. 郑州:信息工程大学,2010

[4] Breiman L, Friedman J, Olshen R, et al. Classification and regression trees[J]. Data Mining and Knowledge Discovery, 1984, 1(1):14-23

[5] Breiman L. Better subset regression using the nonnegative garrote [J]. Technometrics, 1995, 37(4):373-384

[6] 张玲,刘勇,何伟. 自适应遗传算法在车牌定位中的应用[J]. 计算机应用,2008,28(1):185

[7] 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

[8] Tibshirani R. Regression shrinkage and selection via the lasso [J]. J. Royal. Statist. Soc. (B), 1996, 58:267-288

[9] Halstead M. H. Elements of Software Science [M]. New York: Elsevier North Holland, 1977

[10] McCabe T J. A complexity measure [J]. IEEE Transactions on Software Engineering, 1976, SE-2(4):308-320

[11] 李轩,郝克刚,葛玮. 面向对象软件度量的分析和研究[J]. 计算机技术与发展,2006,16(11):38-41

[12] Hastie T, Taylor J, Tibshirani R, et al. Forward stagewise regression and the monotone lasso[J]. Electronic Journal of Statistic, 2007, 1:1-29

[13] Efron B, Hastie T, Johnstone I, et al. Least angle regression[J]. Journal of the Institute of Mathematical Statistics, 2004, 32(2): 407-499

[14] Gray D, Bowes D, Davey N, et al. The Misuse of the NASA Metrics Data Program Sets for Automated Software Defect Prediction [OL]. http://uhra. herts. ac. uk/dspace/bitstream/2299/6363/1/905745. pdf

[15] 高书亮,杨东凯,黄智刚. Galileo 系统伪随机序列生成及其 FPGA 实现[J]. 微计算机信息,2008,24(26):124-125

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

[17] Fredman J H. Multivariate adaptive regression splines [J]. Annals of Statistics, 1991, 19(1):1-67

[18] Hastie T, Taylor J, Tibshirani R, et al. Forward stagewise regression and the monotone Lasso[J]. Electronic Journal of Statistic, 2007, 1:1-29

[19] 张靖,葛玮,郝克刚. 软件度量中主成分分析方法的研究[J]. 计算机技术与发展,2006,12:144-148