doi:10.3969/j.issn.1673-4785.2010.04.007

不变量的程序潜在错误预测

杨振兴,刘久富,孙 琳
(南京航空航天大学 自动化学院,江苏 南京 210016)

摘 要:随着软件系统变得越来越复杂和庞大,软件中的安全缺陷也急剧增加,系统中的隐含错误也在逐渐增多.提出一种基于不变量的程序潜在错误预测方法,首先采用支持向量机对程序属性所产生的非函数依赖程序不变量进行学习并产生机器学习模式,然后运用该机器学习模式对需预测的程序进行属性分类,并揭示出代码可能存在的潜在错误,最后通过实验验证该方法是有效的.

关键词:不变量;软件测试;支持向量机;错误预测
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-4785(2010)04-0327-05

Using invariants to predict the potential for errors in programs

YANG Zhen-Xing, LIU Jiu-Fu, SUN Lin
(College of Automation Engineering, Nanjing University of Areonautics and Astronautics; Nanjing China; 210016)

Abstract: As software systems become increasingly complex and large, deficiencies in software security increase sharply and implicit errors increase gradually. A method based on invariants was developed to predict potential errors in programs. First, a support vector machine was used to find program invariants and produce a pattern for machine learning. Then the pattern from machine learning was employed to classify the programs with behavior to be predicted and reveal the latent errors in codes. Finally an experiment was done that verified the effectiveness of the method.

Keywords: invariants; software testing; support vector machine; error prediction

随着软件系统变得越来越复杂和庞大,软件中的安全缺陷也急剧增加.据统计有40%的系统故障是由软件缺陷引起的[1].通常的测试方法是通过执行测试用例集来发现程序源代码中的错误,一旦程序通过对它所测试的测试用例后,测试便不会再通过程序员来发现错误.然而,程序仍然可能存在潜在的错误,此时,如果想通过新的测试来发现潜在的错误可能会显得很困难并需要付出很大的代价.

由查找隐含错误的一般方法知,许多错误都是由很少的一些类别引起,相似的错误都具有相似的特性,然后这些特性能够被概括和识别.例如,3个普通的错误类别是由一个错误引起的(不正确的使用第一个或最后一个数据结构元素),使用未初始化或部分初始化的值[2].

本文提出一种采用支持向量机对程序属性所产生的非函数依赖程序不变量进行学习并产生机器学习模式的方法,然后运用该模式对需预测的程序进行属性分类,并标示出代码可能存在的潜在错误.该方法的输入是一系列给定程序的属性,输出经辨识后能辨识潜在错误的属性.

收稿日期:2009-10-12.
基金项目:国家自然科学基金资助项目(60674100).
通信作者:杨振兴. E-mail:yzxwfu@163.com.

1 支持向量机理论

潜在错误预测与很多因素有关,因此预测模型的输入量将是一个高维、非线性、小样本的模式识别问题.在前人研究的基础上,选用支持向量机作为机器学习理论对潜在错误进行预测.

潜在错误预测分类模型[3]与排序模型[4]的建立即分别寻求以下表达式的成立

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=0}^{l}\alpha_i y_i k(x_i \boldsymbol{x}) + b\right),$$

$$f(\boldsymbol{x}) = \sum_{i=0}^{l}(\alpha_i^* - \alpha_i)K(x_i \boldsymbol{x}) + b. \qquad (1)$$

式中:α 为可揭错误属性,x_i 为 k 个样本中的第 i 个

样本;$K(x_i,x)$为核函数,核函数采用径向基函数,如下式所示[3].

$$k(x_i,x_j) = \exp(-\gamma \| x_i - x_j \|^2). \quad (2)$$

利用支持向量机工具箱即可对式(1)的模式分类,对(2)的模式排序. 本方法使用了 LibSVM 工具包,LibSVM 是台湾大学林智仁副教授等开发设计的一个简单、易于使用和快速有效的 SVM 模式识别与回归的软件包,包含 C-SVC ,nu-SVC 等多种支持向量机工具的实现,这个软件包可以通过互联网直接从相关的网站上获取.

2 基于不变量的程序潜在错误预测框架

合肥工业大学李嘉研究了基于支持向量机的软件可靠性早期预测,在提出模型的基础上,设计了一个软件可靠性早期预测软件系统,该预测系统把支持向量机引入软件可靠性早期预测领域,可以对软件存在的缺陷数进行预测. 但该系统不能对缺陷进行具体的定位,只能给软件开发人员一个预测的缺陷数量[5]. 本文提出的的方法是对单个程序属性,通过辨识能得到存在错误的的程序属性,并通过该程序属性定位程序中的潜在错误.

2.1 程序不变量

所谓不变量就是一些用于描述在程序运行时保持不变的性质的逻辑断言,它经常出现在断言声明、形式化描述中,例如:$y = 4 \cdot x + 3$;$x > \text{abs}(y)$;array a contains no dup licates;$n = n$;child; parent(for all nodes n);size (keys) = size(contents);graphg is acyclic. 常见的不变量形式有程序的前置条件、后置条件、循环不变量和类不变量[6].

2.2 程序潜在错误预测框架

在该方法中用类似 C 语言中的关系表达式描述不变量,这样可以使用多种方法检测不变量,利用工具解析多种格式描述的不变量成为关系表达式的形式. 本文提出的基于不变量程序潜在错误预测方法的体系结构框图如图 1 所示,他主要由 2 部分组成:通过支持向量机训练得到能够辨识潜在错误的模式和被测程序通过模式辨识得到可揭错误属性(fault-revealing properties). 该方法包括如下步骤:

1)将含有已知错误的代码经过词法语法分析检测编配得到含有错误的属性不变量,将去除错误后的代码经过词法语法分析检测编配得到去除错误后的属性不变量;

2)将 1)所述的含有错误的属性不变量和去除错误后的属性不变量分别经过槽的替换得到支持向量机输入的特征向量,其中槽包括变量类型槽、运算符类型槽、属性类型槽和程序变量槽,下同;

3)将 2)所述的特征向量经过 SVM 模式识别与回归进行机器学习得到能够辨识潜在错误的模型;

4)将用户程序依次经过词法语法分析检测编配、槽后得到用户程序的属性特征向量,将用户程序属性与 3)所述的能够辨识潜在错误的模型匹配去除可揭示性错误.

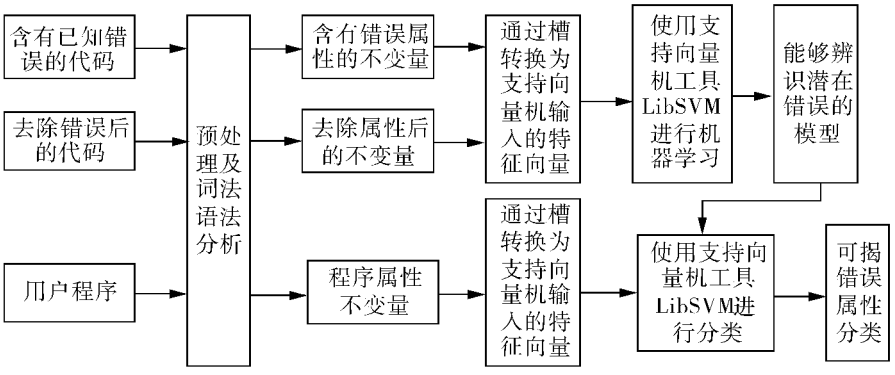

图 1 程序潜在错误预测框图

Fig. 1 The diagram of program potential error prediction

2.3 属性转换为机器学习特征向量

机器学习算法使用特征向量作为输入,因此研究需要将用不变量体现的属性转换为特征向量的形式. 特征向量是一些关于布尔型(bool),整型(int),浮点型(float)等的值,它被认为在多维空间中的一个点. 每维(dimension)被叫做一槽(slot)[7].

在对机器算法的输入之前,每个属性都需要转换成为特征向量,另外,属性如果在当前存在于有缺陷的程序中时,称为可揭错误的属性,如果属性在有缺陷的和无缺陷的版本中都出现了,称为不可揭错误属性,如图 2 所示,左边的椭圆代表含有错误代码的属性,右边的椭圆代表去除已知错误代码的属性,

2 个椭圆公共部分是并未从含有错误代码中去除错误的那些属性即不可揭错误属性.

举个例子,假如这有 4 槽指示一个等式,其中的 3 槽指示涉及变量的类型,还有另外 1 槽指示了变量的个数,其中相对应是和否用 1 和 0 来表示,变量个数用实际的个数值来表示. 在本文的研究方法中,特征向量包含了 44 个槽[8]. 分别列举如下:

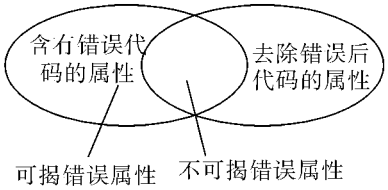

图 2　可揭错误属性

Fig. 2　The fault-revealing properties

1)变量类型槽.

char(字符型),double(双精度),float(单精度),int(整形),long(长整形),short(短整形),void(空型),string(字符串型)共 8 个变量类型.

2)运算符类型槽.

= =,≥,>,≤,<,≠,%(取模),⊆,∈ 共 9 个运算符类型.

3)属性类型槽(cmp 可以为 = =,≥,>,≤,<,≠).

a) Stored:代表一系列数据进行排序

b) Linear:代表变量之间成线性关系

c) Constant:代表不变量之间有常量

d) Min(x):代表不变量为最小值

e) Max(x):代表不变量为最大值

f) $x \neq 0$ or $x \neq$ null:代表变量不为零或空

g) $x == c$:代表不变量恒为常量

h) Array:代表不变量为数组

i) $x[0 \sim n]$ cmp $y[0 \sim n]$:代表变量为一对一的比较,并且长度一致

j) $x[\,] = \sim y[\,]$:代表一个数组为另一个数组的逆序

k) $x[\,]$ cmp y:代表数组的每个量与一个变量比较大小

l) $x[i]$ cmp i:代表数组与自己的序号比大小

m) $x[\,] \neq i$:代表数组全都不等于自己的序号

n) $x[\,]$ cmp $y[\,]$:代表数组的所有值都比另一个大或小

o) Subsequence:代表一个集合是另一个的连续一段子集

p) $x \leqslant c$:c 为常量,x 是标量. 代表变量始终都小于一个常量

q) Iff:代表不变量为当且仅当型

4)程序变量槽.

a) NumVars:代表变量的个数

b) VarinoIndex:代表当前变量的索引

c) IsStaticConstant:代表是否为常静态变量

d) PrestateDerived:代表变量是否是函数开始时派生变量

e) DerivedDepth:代表派生的深度

f) IsClosure:代表变量是否关闭

g) IsParameter:代表变量是否是函数的参数

h) IsReference:代表变量是否是函数传引用

i) IsIndex:代表变量是否是系列的索引

j) IsPinter:代表变量是否为指针

训练阶段需要一个二元组$\langle P, P' \rangle$其中 P 是包含至少一个错误的程序,P' 去除了这些错误之后的程序版本. 程序 P' 并不需要是完全正确无误的,只需去除相对应的错误就可以[8].

3　实例分析

3.1　评价指标

为了更好的分析预测结果,提出模型评价 2 个参数:相关率(Relevance)和简短率(Brevity)[9]. 采用总体相关率和总体简短率,分类相关率和分类简短率以及定量相关率和定量简短率对所述方法进行评价:

$$\text{简短率} = \frac{\text{所有被辨识出的可揭错误属性}}{\text{正确辨识可揭错误属性}},$$

$$\text{相关率} = \frac{\text{正确辨识可揭错误属性}}{\text{所有被辨识出的可揭错误属性}}.$$

属性集的相关率表明被揭错误属性的可能性大小. 总体相关率是所有给定程序的程序属性的整组相关率;分类相关率是被辨识可揭错误属性组的相关率;定量相关率是预先选择定量可能性排考前可揭错误属性的相关率.

一组属性的简短率是与相关率相反的量. 即检测到一个可揭错误属性需要平均属性的个数. 最完美的简短率是 1,也即所有的属性都是可揭错误属性. 同相关率,简短率度量分为总体简短率,分类简短率和定量简短率.

通过机器学习后对预测程序进行模式分类和模式排序,能够得到预测分类的相关率大于总体相关率,即分类的简短率小于总体的简短率,或定量相关率大于总体相关率,即定量简短率小于总体简短率,则表明该潜在错误预测方法是有效的.

3.2　实　例

为了验证理论方法的正确性,作者采用李春葆、

刘斌编著的《C 语言程序设计考点精要与解题指导》一书中的改错题型提取修正前和修正后的程序,通过检测编配代码提取 52 个数据样本,此类数据样本中含有相似数量的可揭错误属性与不可揭错误属性的程序属性不变量. 运用 LibSVM 工具包对所提取的 52 个数据样本进行训练,分别运用 LibSVM 工具中的 C-SVC 分类机和 ε-SVR 回归机对实例属性不变量进行分类和排序.

由于预测模型生成信息量大,为说明该方法的预测步骤,作者对如下一个程序实例列出部分属性不变量对方法进行说明.

```
double[ ] Against(double orig[ ],int num) {
  //orig[ ]为原数组,num 为数组元素的个数
    int i;
    double temp,out[ num ];
    memcpy(out,orig,num)
    for( i = 0; i < =num/2, i + + ){
      temp = out[ i ];
      out[ i ] = out[ num- i ];
      a[ num- i ] = temp;
    }
  return out[ ];
}
```

将实例程序经过检测编配分析得到表 1 中的 8 个程序属性不变量,采用了 8 槽来表示特征向量(实际运用时模式的产生运用了 44 个槽),特征向量信息如表 1,如:out[num] = orig[0]不变量的的特征向量是〈1,0,0,1,0,0,2,1〉(此处的 1 和 0 代表真和假),将之前学习所产生的分类机和回归机分别对表中的 8 个特征向量进行辨识及预测,所得结果如表 1 所示.

表 1 不变量转换为特征向量表

Table 1 Invariants to characteristic vector converter

属性不变量	等式			变量类型			程序变量槽		分类结果	排序结果
	=	≠	⊆	double	int	Array	NumVars	VarinoIndex		
out[num] = orig[0]	1	0	0	1	0	0	2	1	1	91
i = num/2 +1	1	0	0	0	1	0	2	0	1	89
size(out) = size(in)	1	0	0	0	1	0	2	0	1	89
out[num] = null	1	0	0	1	0	0	1	1	1	77
orig⊆out	0	0	1	0	0	1	2	0	-1	47
out⊆orig	0	0	1	0	0	1	2	0	-1	47
out≠null	0	1	0	0	0	1	1	0	-1	32
orig≠null	0	1	0	0	0	1	1	0	-1	32

分析实例程序得到 out[num] = orig[0] 和 i = num/2 +1 存在错误,根据表 1 中的结果,得到预测结果相关率与简短率指标如表 2 所示,并对该方法作如下分析说明:

表 2 预测结果相关率与简短率表

Table 2 The result of relevance and brevity

	总 体	分 类	定量(3)
相关率	2/8 =0.25	2/4 =0.5	2/3 = 0.67
简短率	8/2 = 4	4/2 = 2	3/2 =1.5

1)分类结果中有 4 个属性被辨识为存在错误,实际其中的 2 个存在错误,根据评价指标得到分类的相关率为 0.5 大于总体属性不变量的相关率为 0.25,相应的分类的简短率小于总体简短率.

2)假设规定错误可能性排列在前 3 个属性为可揭错误属性,同上得到定量的相关率为 0.67 大于总体相关率的 0.25,相应的简介性小于总体简介性.

3)根据所预测将得到的可揭错误属性定位到程序中,由程序员审查程序中可能存在的错误并将其修改.

4)分类相关率与定量相关率都大于总体相关率,即说明所提出的方法是有效的.

5)训练样本的获取非常重要,该方法的应用中应取自实际项目中测试得到的历史数据,即测试前的可揭错误属性不变量与去除错误后的的程序不变量作为训练的二元组〈P ,P'〉.

6)训练样本应该是与所预测程序相同编程语言所编写,并从测试该项目组之前所完成的项目中提取数据样本为最佳,这可提高预测的精度,提高测试效率.

4 结束语

提出的程序潜在错误预测方法主要研究了应用支持向量机对代码属性进行学习并产生模式,通过

模式对用户代码进行预测潜在错误的方法,得到可能隐含更深的程序错误,提高程序的可信性.对属性不变量的提取、属性向特征向量的转换、属性分类进行了探讨.重点研究了属性转换为与代码无关的特征向量.

该方法运用支持向量机对软件潜在错误进行定位分析,能快速查找到程序中错误,并可以反复使用,提高软件测试效率;通用性强,在预测阶段用来训练的不是具体的属性,而是学习一些属性槽,通过支持向量机学习之后可以完全应用于同语言的不同程序中.

参考文献:

[1] MARCUS E, STERN H. Blueprints for high availability [M]. Washington: John Wiley, 2003:38-42.

[2] BRUN Y. Software fault identification via dynamic analysis and machine learning[D]. Cambridge Massachusetts: Massachusetts Institute of Technology, 2003.

[3] 汪廷华,田盛丰,黄厚宽. 特征加权支持向量机[J]. 电子与信息学报, 2009,31(3):514-518.
WANG Tinghua, TIAN Shengfeng, HUANG Houkuan. Feature weighted support vector machine[J]. Journal of Electronics & Information Technology ,2009, 31(3):514-518.

[4] 程勇,罗键,吴长庆. 基于支持向量机的环境质量评估方法[J].计算机工程与应用,2009,45(2):209-211.
CHENG Yong, LUO Jian, WU Changqing. Environmental quality evalution based on SVM[J]. Computer Engineering and Applications,2009,45(2):209-211.

[5] 李　嘉. 基于支持向量机的软件可靠性早期预测研究[D].合肥:合肥工业大学, 2005.
LI Jia. The study of software reliability early prediction based on support vector machine[D]. HeFei: Hei fei University of Technology, 2005.

[6] 刘杰,阳小华,罗扬等. 一种交互式的不变量动态发现编配工具[J].计算机应用与软件.2008,10:85-86.
LIU Jie, YANG Xiaohua, LUO Yang, WU Qujing. An interacten instrum enter for dynamically discovering invariant [J]. Computer Applications and Software, 2008, 10:85-86.

[7] ERNST M D, COCKRELL J, GRISWOLD W G. Dynamically discovering likely program invariants to support program evolution[J]. IEEE TSE, 2001,27(2): 1-25.

[8] GRAVES T L, HARROLD M J, KIM J M, PORTER A, ROTHERMEL G. An empirical study of regression test selection techniques[J]. ACM Transactions on Software Engineering and Methodology, 2001,10(2): 184-208.

[9] SALTON G. Automatic information organization and retrieval[M]. New York : McGraw-Hill, 1968:485-498.

作者简介:

杨振兴,男,1985年生,硕士研究生,主要研究方向为软件测试与人工智能.

刘久富,男,1971年生,博士,硕士生导师,主要研究方向为软件测试技术与软件质量工程.

孙　琳,男,1983年生,硕士研究生,研究方向:软件测试.