

基于支持向量机回归模型的测试用例生成与重用

钱忠胜,俞情媛,宋 涛,朱懿敏,祝 洁,赵 畅

(江西财经大学信息管理学院,江西南昌 330013)

摘 要: 在软件测试领域,利用遗传算法生成测试用例是一个研究热点.传统方法在利用遗传算法生成测试用例时,需要计算每个个体的适应度.为了降低适应度计算的时间消耗并重用测试用例,提出一种融入支持向量机回归模型的测试用例生成与重用的方法.在使用遗传算法生成测试用例的过程中,利用一定数量的个体及其适应度作为样本训练支持向量机回归模型.在之后的种群进化中,根据回归模型计算个体适应度,同时利用回归模型查找适应度较高的个体并重用到新种群的进化中.在某大型程序实验中,该方法与同类经典方法相比,覆盖率提高了3%,平均进化代数也有所降低,其降低百分比达85.97%.

关键词: 测试用例;测试重用;支持向量机;遗传算法;适应度

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112(2021)07-1386-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20200426

Test Case Generation and Reuse Based on Support Vector Machine Regression Model

QIAN Zhong-sheng, YU Qing-yuan, SONG Tao, ZHU Yi-min, ZHU Jie, ZHAO Chang

(School of Information Management, Jiangxi University of Finance & Economics, Nanchang, Jiangxi 330013, China)

Abstract: In the field of software testing, it is a hot research spot to generate test cases using genetic algorithm. In the traditional process of generating test cases by genetic algorithm, it is necessary to calculate the fitness of each individual. In order to reduce the time consumption of fitness calculation and reuse test cases, a test case generation and reuse method based on support vector machine regression model is proposed. In the process of using genetic algorithm to generate test cases, a certain number of individuals and their fitness are used as samples to train the support vector machine regression model. In the subsequent population evolution, individual fitness is calculated according to the regression model. At the same time, individuals with higher fitness are found by the regression model and applied to the evolution of the new population. In the experiment on a large program, compared with that of the same classical method, the coverage rate of this approach is increased by 3% and the average evolutionary time is also reduced by 85.97%.

Key words: test case; test reuse; support vector machine; genetic algorithm; fitness

1 引言

减少在软件测试上的花费,提高测试用例生成效率是软件领域热门研究对象之一.

近年来,遗传算法在软件测试中的应用取得了丰硕的成果^[1-3].使用遗传算法生成新的用例时,传统方法计算个体适应度需要花费大量运行时间^[1].

测试的重用是指在新的测试中重复使用到已生成的测试资源,从而提高软件测试效率.

本文就如何减少计算适应度的时间消耗和提升测

试用例的利用率,提出一种融入支持向量机回归(Support Vector Machine Regression, SVR)模型的测试用例生成与重用方法.使用遗传算法在种群迭代过程中生成的个体训练SVR模型,此方法两次使用到该模型,其一利用该模型代替插桩法进行个体适应度值的模拟计算,其二在测试用例重用的过程中引用该模型查找适应度较高的个体,将这些优秀个体引入到种群进化中.基于SVR模型的测试用例生成与重用方法如图1所示.

径个体的适应度范围。

遗传算法模块包含 SVR 模型的训练以及模型使用的方法。

4.1.2 遗传算法模块

采用遗传算法进行测试用例的生成,主要步骤如下:

- (1) 种群初始化;
- (2) 种群个体输入插桩程序求得每个个体的适应度值,并输入到 SVR 模型训练;个体适应度由训练好的模型计算其近似值;
- (3) 对于适应度值在预设阈值内的个体利用插桩程序求其准确值;

(4) 满足算法终止条件或达到最大迭代次数,算法终止,否则转步骤(5);

- (5) 对个体进行选择、交叉、变异操作,转步骤(3).

利用训练模型代替插桩法计算适应度值旨在减少种群进化时间.用例的重用会再次运用到训练模型,下一节将详细讲述.

4.2 结合 SVR 模型的测试用例重用

重用是将训练好的模型引用到用例生成中,再一次提高用例的生成效率.

测试用例重用模块

用训练的 SVR 模型完成重用步骤如下:

- (1) 利用训练好的模型选择适应度较高的个体;
- (2) 将选择测试的数据作为遗传个体封装成数据库引入到遗传算法的进化过程;
- (3) 种群在进化下一代时以一定概率与数据库中引入的个体(随机选择)进行交叉;
- (4) 为避免出现局部最优的情况,若个体被引用超过一定数值(此处将其设置为 10),则从数据库中将其移除;
- (5) 满足算法终止条件或达到最大迭代次数,算法终止,否则转步骤(6);
- (6) 在每一代新的种群进化过程中,重复步骤(3)和步骤(4).

5 实验设计与分析

为了检验 SVR 模型模拟个体适应度的准确性,以及用例重用的有效性.本节通过大量实验进行验证与对比分析.

5.1 评价标准

这里主要使用如下 4 个评价指标.

(1) 覆盖率:对程序运行 100 次,记录在最大进化代数 $T=1000$ 内目标路径被覆盖的总次数,覆盖率为这 100 次的平均值.

(2) 平均进化代数:目标路径被覆盖的每个程序运行

100 次,若最大进化代数 $T=1000$ 内覆盖了目标路径,记录此时的进化代数;若最大进化代数内没有覆盖,记录最大进化代数,平均进化代数即为这些记录值的均值.

(3) 执行时间:测试数据或路径的选择与测试数据的生成所消耗的时间总和.

(4) 执行时间差百分比:SVR 法(本文方法的适应度值预测部分)和本文方法分别相对于插桩法的执行时间差占插桩法所用时间的百分比.

5.2 实验设置

本节选取 2 个基准程序和 8 个工业程序进行实验.程序信息如表 1 所示.

表 1 被测程序的基本信息

程序编号	待测程序	代码行	函数个数	简单功能描述
1	Triangle	15	1	三角形判定
2	Bubblesort	15	1	冒泡排序
3	Gobang	265	6	五子棋赢棋算法
4	Check	209	13	个人信息校验算法
5	Datahand	462	14	银行数据处理
6	Space	972	35	字符串语言解释器
7	Go	1233	39	抽象策略性游戏
8	SpiceAlgo	2880	65	桌面交互部分算法
9	Noark	26010	285	游戏服务器端框架
10	Spice	149050	7254	虚拟桌面交互系统

实验输入变量均采用二进制编码,通过遗传算法完成用例重用.实验参数初始种群为 40,交叉概率为 0.7,变异概率为 0.1,最大进化代数为 1000.

5.3 实验结果及讨论

从小规模与较大规模程序中分别随机抽取 1 个程序,训练 SVR 模型,通过预测值与真实值对比,判定模型的预测效果.

这里选用折线图展示适应度对比结果,包括插桩法、SVR 方法、插桩法与 SVR 方法间的差值(即 SVR 适应度预测值与真实值之差),如图 2 所示,以及适应度值之差百分比,如图 3 所示.两图的横坐标均表示测试数据编号,图 2 的纵坐标为个体适应度值,图 3 的纵坐标为适应度值之差百分比.

从图 2 的差值曲线中可初步看出,预测值和真实值之间的差距较小,表明 SVR 模型模拟适应度的准确性较高,便于在后续选择合适的适应度值范围.

由图 3 可知,在较小和较大的两个程序上,本文方法预测的适应度值与插桩法计算的真实值之差的百分比在较小范围内波动.其中小程序的适应度值之差百分比主要分布在 $-10\% \sim 9\%$ 内,偶尔有差距较大的情况(部分在图中用红点标记);较大程序的适应度值之差百分比分布较为集中,主要在 $-5\% \sim 7\%$ 之间.相比在较

小程序上,SVR模型在较大程序上的预测效果更好,随着程序规模增大,本文预测效果会越好.

表 2 是插桩法、SVR法(本文方法的适应度值预测

部分)以及本文方法测试程序生成测试用例的结果.

下面将插桩法和SVR法、插桩法和本文方法以及SVR法和本文方法做对比分析.

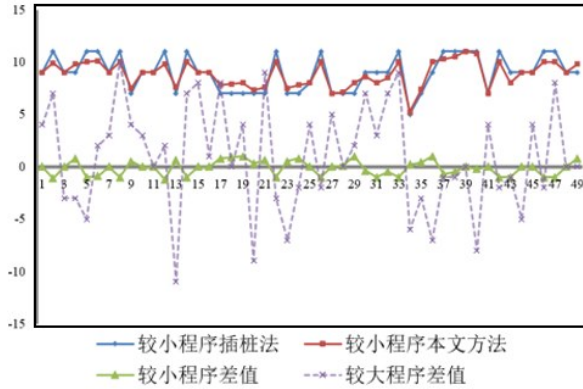


图2 适应度预测值与真实值对比结果

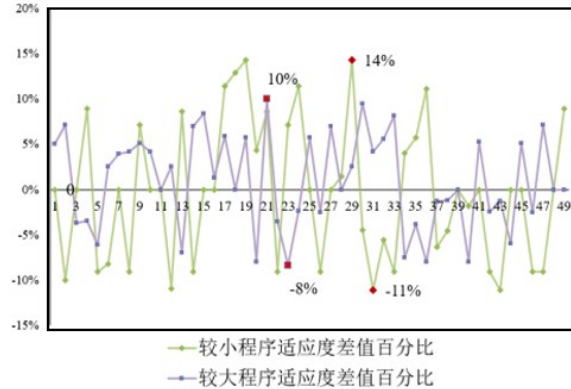


图3 适应度值之差百分比图

表 2 插桩法、SVR法与本文方法对程序测试的结果对比

待测程序	评判标准	方法	插桩法		SVR法(本文方法的适应度值预测部分)		本文方法		执行时间差百分比(%)	
			覆盖率(%)	平均进化代数	覆盖率(%)	平均进化代数	覆盖率(%)	平均进化代数	插桩法与SVR法之差比率	插桩法与本文方法之差比率
Triangle (三角形判定)			74	295.67	76	210.03	100	10.03	-43.44	-66.89
Bubblesort (冒泡排序)			49	695.02	50	693.10	100	9.25	-20.32	-41.28
Gobang (五子棋)			66	349.24	60	398.44	100	18.60	22.18	8.41
Check (信息校验)			83	230.90	77	306.64	100	38.10	13.19	4.32
Datahand (企业数据处理)			85	198.01	82	254.76	100	15.90	14.00	8.31
Space (字符串语言解释器)			64	362.30	62	389.65	100	21.20	27.66	17.82
Go (抽象策略性游戏)			71	284.10	66	299.68	100	36.10	18.92	9.22
SpiceAlgo (桌面交互部分算法)			62	371.50	60	386.92	100	34.20	15.17	5.76
Noark (游戏服务器端框架)			58	578.62	54	597.86	100	30.90	17.20	7.37
Spice (虚拟桌面交互系统)			53	595.10	52	618.34	97	40.20	18.52	9.09

(1) 对比插桩法和 SVR 法可知,在两个评价指标(目标个体的覆盖率以及平均进化代数)方面相差较小,可认为基本保持一致。

时间方面,当待测程序规模增大时,插桩法生成测试用例所用的时间显著增加,而 SVR 法的时间增加相对缓慢。其中,插桩法测试程序花费的最少时间为 35.82s,最多为 1575.31s,而 SVR 法花费的最少时间为 51.38,最多为 1286.62s。因此,SVR 法在测试规模相对较大程序时能节约时间,程序规模越大,更能体现该方法的优点。

(2) 本文方法是在 SVR 法的基础上将 SVR 模型运用到测试用例的重用中。由表 2 可知,本文方法测试程序提高了目标路径的覆盖率,平均进化代数也有所减小。前 9 个被测程序的覆盖率均为 100%,第 10 个最大的程序为 97%;测试冒泡排序程序的平均进化代数仅有 9.25,而最大的平均进化代数是大型程序 10 得到的 40.20,该值也比较小。

时间方面,本文方法在小的程序上优势不明显,

表 3 本文方法与其他四种方法对程序测试的结果对比

待测程序	各种方法	覆盖率(%)		平均进化代数		平均进化代数降低百分比(%)	
		参数同本文法	参数同对比法	参数同本文法	参数同对比法	参数同本文法	参数同对比法
1	文献[2]	/	100	/	70.90	85.85	85.85
	本文方法	100	/	10.03	/		
2	文献[3]	100	100	10.34	10.08	10.54	4.07
	本文方法	100	100	9.25	9.67		
6	文献[4]	100	100	91.30	83.68	76.78	72.96
	本文方法	100	100	21.20	22.63		
10	文献[1]	94	93	286.52	295.67	85.97	76.90
	本文方法	97	96	40.20	68.30		

说明:因文献[2]方法与本文方法的算法性质不同,其实验参数无法统一,故结果均采用各自原实验数据

由表 3 可知,在覆盖率方面,五种方法均表现良好。其中,在最大的复杂 Spice 程序(程序 10)的两组实验中,本文方法覆盖率分别为 97%、96%,比神经网络法(文献[1])均提高了 3%(神经网络法分别为 94%、93%)。

本文方法的优势还体现在平均进化代数方面,在两组实验参数设置分别相同的情况下,4 个不同程序中的平均进化代数均小于其他方法,表明本文方法更优。当程序规模增大时(排除较小的程序 1),平均进化代数降低百分比数值逐渐提高,这说明本文方法在规模越大的程序中其优势会越来越明显。

6 总结与下一步工作

本文提出一种基于 SVR 模型的测试用例生成与重用方法,主要工作与贡献如下:

(1) 在遗传算法生成用例的过程中,引用生成的用例及插桩法求得的适应度作为样本训练 SVR 模型,该

因此,排除程序 1 和程序 2 这两个规模非常小的程序,本文方法测试程序花费的时间均比传统的插桩法要少。

(3) 与 SVR 法相比,本文方法肯定会多用时间,因为 SVR 法只是作为本文方法的一部分,用于模拟预测种群个体的适应度值。若该值在优秀个体的适应度值范围内,则将其对应的个体进行插桩验证。

5.4 对比分析

通过上述分析讨论,结果表明本文方法在用例生成方面较其他两种方法明显占优。下面进一步将本文方法与其他典型方法展开对比分析。

在四个程序,即表 1 的程序 1 (Triangle)、程序 2 (Bubblesort)、程序 6 (Space)、程序 10 (Spice)上,将本文方法分别与姜淑娟^[2]、丁蕊^[3]、夏春燕^[4]和姚香娟^[1]提出的方法进行对比,设置了两组实验:1) 实验环境和参数与本文方法相同;2) 实验环境和参数与相应对比方法相同。实验结果如表 3 所示。

模型能够模拟计算测试用例适应度值。

(2) 将训练模型运用到遗传算法生成用例的过程中,并根据模型预测的个体适应度值设定优秀个体适应度值区间,从而减少个体适应度求值时间。

(3) 运用训练的测试模型在相应程序用例数据库中查找适应度较高的个体,将其重用到遗传算法的用例生成中。这进一步减少用例生成时间,并提高目标路径覆盖率。

在未来工作中,进一步考察程序间用例重用的局限性,特别考虑训练模型拟合效果比较差的情况,以提高用例重用的通用性与准确性。

参考文献

- [1] 姚香娟, 巩敦卫, 李彬. 融入神经网络的路径覆盖测试数据进化生成[J]. 软件学报, 2016, 27(4): 828-838.
YAO Xiang-juan, GONG Dun-wei, LI Bin. Evolutional

test data generation for path coverage by integrating neural network [J]. Journal of Software, 2016, 27(4): 828 – 838. (in Chinese)

- [2] 姜淑娟, 王令赛, 薛猛, 等. 基于模式组合的粒子群优化测试用例生成方法[J]. 软件学报, 2016, 27(4): 785 – 801.

JIANG Shu-juan, WANG Ling-sai, XUE Meng, et al. Test case generation based on combination of schema using particle swarm optimization [J]. Journal of Software, 2016, 27(4): 785 – 801. (in Chinese)

- [3] 丁蕊, 董红斌, 张岩, 等. 基于关键点路径的快速测试用例自动生成方法[J]. 软件学报, 2016, 27(4): 814 – 827. DING Rui, DONG Hong-bin, ZHANG Yan, et al. Fast automatic generation method for software testing data

based on key-point path [J]. Journal of Software, 2016, 27(4): 814 – 827 (in Chinese)

- [4] 夏春艳, 张岩, 万里, 等. 基于否定选择遗传算法的路径覆盖测试数据生成[J]. 电子学报, 2019, 47(12): 2630 – 2638.

XIA Chun-yan, ZHANG Yan, WAN Li, et al. Test data generation of path coverage based on negative selection genetic algorithm [J]. Acta Electronica Sinica, 2019, 47(12): 2630 – 2638. (in Chinese)

- [5] Nhyuk C, Hyunggoy O, Young-Woo L, Sungho K. Test resource reused debug scheme to reduce the post-silicon debug cost [J]. IEEE Transactions on Computers, 2018, 67(12): 1835 – 1839.

作者简介



钱忠胜(通信作者) 男,1977年1月出生,江西鹰潭人.2008年在上海大学获工学博士学位.现为江西财经大学教授,博士生导师.主要研究方向为软件工程等.

E-mail:changesme@163.com



俞情媛 女,1997年5月出生,江苏淮安人.江西财经大学信息管理学院硕士研究生.主要研究方向为软件测试等.