

# Analysis of Failure Number Change Based on GAM Model

Yuxin Yang

Beihang University, Beijing  
Email: yangyx893@163.com

Received: Jun. 30<sup>th</sup>, 2019; accepted: Jul. 10<sup>th</sup>, 2019; published: Jul. 17<sup>th</sup>, 2019

---

## Abstract

In this paper, the process data in the process of software production is used to analyze the failure change data in the test process by using GAM with interaction, and to study the influencing factors and the interaction between the factors. In this paper, the fitting effects of GAM and GAM with interaction are compared firstly with the failure change data. Secondly, the prediction results of GAM with interaction and existing machine learning model are compared. The results show that the GAM with interaction is more abundant than that obtained by GAM, and it is better than the existing machine learning method. Therefore, such analysis and analysis is meaningful.

## Keywords

Failure Number, GAM Model, Interaction

---

# 基于GAM模型对失效数变化的分析

杨玉鑫

北京航空航天大学, 北京  
Email: yangyx893@163.com

收稿日期: 2019年6月30日; 录用日期: 2019年7月10日; 发布日期: 2019年7月17日

---

## 摘要

本文针对软件生产进程中测试的过程数据, 利用带有交互作用的GAM, 对测试过程中的失效变化数据进行分析, 研究影响因素以及因素间的交互作用。本文首先对失效变化数据, 比较了GAM和带有交互作用的GAM的拟合效果; 其次, 比较了带有交互作用的GAM与现有的机器学习模型的预测结果。结果表明, 带有交互作用的GAM相较于GAM所得的结果更加丰富, 且优于已有的机器学习方法, 因此这样的探讨分析是有意义的, 并对软件开发以及采取测试决策有一定的实际指导意义。

## 关键词

失效数, GAM模型, 交互作用

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

对于安全关键软件而言, 在整个软件生产的生命周期中会有大量的数据被记录下来, 称之为过程数据[1]。不仅包括诸如软件代码行数、软件复杂度、软件版本、需求设计等软件开发过程数据, 还有很重要的测试过程数据, 比如测试用例的个数, 发现失效的个数、测试覆盖情况。对于过程数据的分析有利于开发人员改变测试决策, 进而提高软件可靠性。对于这部分数据的处理, 现阶段有机器学习、深度学习的方法来进行失效数预测。

另外, 基于软件可靠性模型的传统可靠性分析方法使用失效数据进行建模, 这种可靠性分析方法忽略了软件本身的结构以及测试过程的特征造成对抽样观察到的失效数据的影响, 没有充分利用安全关键软件开发过程以及自测试过程的特征数据。为了改进这种现象, 提出了机器学习的方法对软件测试过程中的数据进行回归, 比如: 决策树、随即森林、岭回归等方法但是这种数据的数量是较少的, 机器学习方法的拟合会出现偏差。并且利用机器学习方法的输入并没有考虑到影响因子之间的交互作用, 为了考虑影响因子之间的交互作用, 故本文提出了广义可加模型[2], 简称 GAM, 对多元数据进行拟合。

GAM 是由数据驱动而非统计分布模型驱动的非参数回归模型, 可对部分解释变量进行线性拟合, 对其他因子进行光滑函数拟合。模型不需要预先设定参数模型, 模型通过解释变量的平滑函数建立, 能够自动选择合适的多项式。GAM 属于非参数回归模型中的一种, 非参数回归不需要模型满足线性的假设前提, 可以灵活的探测数据间的复杂关系, 但是当模型中的自变量数目较多的时候, 模型的估计方差会加大, 另外, 基于核与光滑样条估计[3]的非参数回归中自变量与因变量间关系的解释也有难度, 1985 年 Stone [4] [5]提出加性模型, 模型中每一个加性项使用单个光滑函数来估计, 在每一加性项中可以解释因变量如何随自变量变化而变化, 很好的解决了上述问题。1990 年 Hastie 和 Tibshirani [4] [5]扩展了加性模型的应用范围, 提出了广义加性模型。

本文针对软件生产与自测试过程中产生的过程数据, 利用 GAM 分析影响因素与它的交互作用, 进一步拟合软件的失效数, 并对实际工程提出有效可行的建议。其中, 文章的结构如下:

第一部分是对问题背景的介绍。

第二部分是对问题进行详细叙述与模型的详细介绍。

第三部分是运用模型对数据进行分析的结果。

第四部分是对数据分析结果的总结与实际指导意义。

## 2. 模型介绍

软件开发过程中的测试数据分析对调整测试策略和可靠性分析具有重要的意义。对于较小的数据量, 机器学习模型并没有很好的预测效果, 并且无法给出因素之间的交互作用, 而在软件的实际研发中因素之间的交互作用往往是存在的, 例如, 代码行数和测试用例个数之间的交互作用。本文尝试用 GAM 来

分析各个因素的影响以及因素之间的交互作用，以提高数据分析的效果。

GAM，属于半参模型[6]中的一种，其表示形式如下所示：

$$g(u) = a + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p)$$

其中， $u = E(Y/X_1, X_2, X_3, \dots, X_p)$ ， $g(u)$  是连接函数， $f_1, f_2, \dots, f_p$  是连接解释变量的平滑函数， $X_i$  为解释变量， $f_i(X_i)$  是关于  $X_i$  的非指定类别的非参数函数。

在该模型中，响应变量的分布可以是正态分布，二项分布、poisson 分布等。对  $f_i(X_i)$  的估计方法有平滑样条法，局部加权回归散点平滑法、薄板平滑样条法，平滑参数的选择有交叉验证法、广义交叉验证法等。在 GAM 模型具体分析过程中，采用 R 64.3.5.0 版 R 软件及 mgcv、DAAG 等软件包，来进行实验仿真。

像线性回归一样，可以通过增加形式为  $X_i \times X_j$  的交互项，使得 GAM 能够体现交互效应的作用。另外，可以增加形式为  $f_{ij}(X_i, X_j)$  的低维交互项，这样的交互项可以应用一些二维光滑方法，如局部回归或者二维样条来拟合。

GAM 应用的潜在假设为函数是可加的，允许每个协变量作为一个不加限制的平滑函数，而不是仅仅作为一种呆板的参数函数被拟合，对部分或全部的解释变量采用平滑函数的方法建立模型。GAM 与 GLM [7] 共同的特点是用连接函数来估计响应变量和各解释变量间的关系。与 GLM 不同的是 GAM 中的各函数可以是非参数的形式，因而使得其应用范围更为广泛，GAM 是 GLM 的一种推广。因此 GAM 适用于多种分布类型，多种复杂非线性关系的分析。

### 3. 分析过程

#### 3.1. 数据描述

本文中所用的数据是来自于真实工程中的项目数据，是在软件生产过程中开发人员进行自测试产生的过程数据。变量解释如表 1 所示，数据格式如表 2 所示。

Table 1. Symbolic explanation

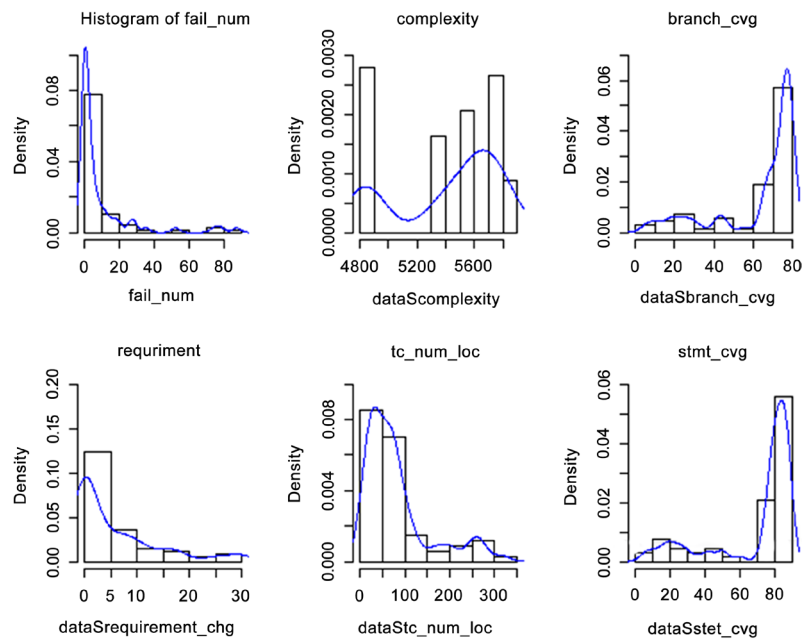
表 1. 符号解释

变量序号	变量符号	变量所代表的意义	变量类型
1	Stage	测试阶段	数值型
2	Weeks	测试周	数值型
3	Requirement_chg	需求变更量	数值型
4	Loc	代码行数	数值型
5	Complexity	复杂度	数值型
6	Tc_num	测试用例个数	数值型
7	Tc_num_loc	测试资源分配特征之一，测试用例针对软件代码行数的分配	数值型
8	Tc_num_complexity	测试资源分配特征之一，测试用例针对软件圈复杂度的分配	数值型
9	Stmt_cvg	语句覆盖率	数值型
10	Stmt_cvg_tc_num	每个测试用例对语句覆盖的贡献	数值型
11	Branch_cvg_tc_num	每个测试用例对分支覆盖的贡献	数值型
12	Branch_cvg	分支覆盖率	数值型

**Table 2.** Data sheet  
**表 2.** 数据简表

stage	weeks	stage_weeks	requirement_chg	loc	complexity	tc_num_loc
1	1	1	0	42140	4840	9.49
1	2	2	0	42140	4840	18.98
1	3	3	0	42140	4840	26.1
1	4	4	0	42140	4840	30.85
1	5	5	0	42140	4840	29.9
1	6	6	4	42140	4840	44.14
1	7	7	4	42140	4840	11.39
1	8	8	0	42140	4840	74.51
tc_num_complexity	tc_num	stmt_cvlg	branch_cvlg	stmt_cvlg_tc_num	branch_cvlg_tc_num	fail_num
8.26	40	5	5	12.5	12.5	1
16.53	80	8	8	10	10	0
22.73	110	12	11	10.91	10	3
26.86	130	15	15	11.54	11.54	20
26.03	126	18	18	14.29	14.29	19
38.43	186	20	20.5	10.75	11.02	0
9.92	48	20	20.5	41.67	42.71	16
64.88	314	24	25	7.64	7.96	3

对数据进行初步的分析(如图 1 所示), 经过 Kolmogorov-Smirnov 检验[8]可知, 假设软件失效数为 poisson 分布, 经检验, P 值为 0.0678, 故假设软件失效数(fail\_num)是呈 poisson 分布类型, 同理可以假设, 这些解释变量为正态分布类型, 因此, 采用 poisson 函数作为连接函数, 将解释变量通过线性组合的方式来联结服从 poisson 分布的响应变量。



**Figure 1.** Variable distribution map  
**图 1.** 变量分布图

多重共线性[9]是指线性回归模型中的解释变量之间存在高度相关关系,它的存在使得模型无法求解。解释变量之间存在的共线性,可利用两个解释变量间的 Pearson 相关系数来判别两者的相关性程度,如果解释变量间的相关系数较大,则两个解释变量之间通常存在严重的共线性关系。处理变量间共线性的方法有主成份分析法、岭回归法、剔除变量法等。

由表 3 可知,代码行数、复杂度、语句覆盖率的相关系数均大于 0.7,因此,它们之间存在着很强的共线性。又有每个测试用例对语句覆盖的贡献和每个测试用例对分支覆盖的贡献相关性很高为 0.998。

**Table 3.** Pearson coefficient

**表 3.** Pearson 系数

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0.97	-0.29	0.64	0.76	0.43	0.42	0.45	0.67	0.64	-0.18	-0.19
2	0.97	1	-0.2	0.72	0.82	0.45	0.44	0.47	0.79	0.77	-0.14	-0.16
3	-0.2	-0.2	1	0.03	-0.02	0.04	0.05	0.04	0.17	0.20	0.14	0.14
4	0.64	0.72	0.03	1	0.98	0.25	0.26	0.30	0.77	0.73	-0.04	-0.05
5	0.76	0.82	-0.02	0.98	1	0.31	0.32	0.35	0.80	0.76	-0.07	-0.08
6	0.43	0.45	0.04	0.25	0.31	1	0.99	0.99	0.39	0.40	-0.40	-0.4
7	0.42	0.44	0.05	0.26	0.32	0.99	1	0.99	0.39	0.41	-0.40	-0.4
8	0.45	0.47	0.04	0.30	0.35	0.99	0.99	1	0.42	0.43	-0.40	-0.4
9	0.67	0.79	0.17	0.77	0.80	0.39	0.39		1	0.99	0.06	0.05
10	0.64	0.77	0.20	0.73	0.76	0.41	0.41	0.43	0.99	1	0.07	0.06
11	-0.18	-0.14	0.14	-0.04	-0.07	-0.4	-0.4	-0.4	0.06	0.07	1	0.99
12	-0.19	-0.16	0.14	-0.05	-0.08	-0.4	-0.4	-0.4	0.05	0.06	0.99	1

### 3.2. GAM 模型的单因素分析

根据变量预分析的结果,将 stage (测试阶段)、weeks (测试周)、requirement\_chg (需求变更量)、loc (代码行数)、complexity (复杂度)、tc\_num\_loc (测试资源分配特征)、tc\_num (测试用例的个数)、stmt\_cvg 语句覆盖率)、branch\_cvg (分支覆盖率)、stmt\_cvg\_tc\_num (每个测试用例对语句覆盖的贡献)、branch\_cvg\_tc\_num (每个测试用例对分支覆盖的贡献)等十二个影响因素作为解释变量,每次选择 1 个影响因素作为解释变量,失效数 (fail\_num)作为响应变量,采用样条平滑函数分别构建模型,并分析每个解释变量对响应变量的影响显著性及模型的拟合优度,如表 4 所示。结果表明:所有的影响因素均在 P 值 < 0.01 水平下对失效数变化有显著影响,表明各影响因素单独作为失效数变化解释变量具有统计学意义; tc\_num (测试用例的个数)、branch\_cvg\_tc\_num (每个测试用例对分支覆盖的贡献)、stmt\_cvg\_tc\_num (每个测试用例对语句覆盖的贡献)对失效数变化影响的模型解释率相对较高,调整判定系数较大,表明这三个因素与失效数变化构建模型方程拟合度较优;其他影响因素对失效数变化的解释率较低,调整判定系数也较低,表明他们单独对失效数进行模型拟合时的解释能力较差,虽然部分解释变量对模型的拟合效果较差,但是他们均通过了显著性检验,均具有统计学意义。

当自由度值为 1 时,函数为线性方程,表明影响因素与响应变量失效数间具有某种线性关系;当自由度大于 1 时,表示函数是非线性曲线方程,影响因素与失效数之间具有某种非线性关系,且值越大,非线性关系越显著。结果表明:loc (代码行数)、complexity (复杂度)的自由度为 1,与失效数变化具有显著的线性关系,branch\_cvg (分支覆盖率)自由度为 2 左右,与失效数变化之间具有一定的非线性关系,而其他变化具有显著的非线性关系。

**Table 4.** Univariate analysis  
**表 4.** 单因素分析

平滑效应项	估计自由度	参考自由度	F	P-value	方差解释率	调整判定系数: R <sup>2</sup>
stage	3.84	4	88.74	<2e-16	5.98	-0.0199
weeks	3.647	4	88.83	<2e-16	6.34	-0.0199
Requiemnt_chg	6	5	2	0.0106	3.8	0.0271
loc	1	1	67.01	<2e-16	4.88	0.0048
complexity	1	1	54.03	4e-15	4.12	0.0014
Tc_num	8.559	9	284.5	<2e-16	21.4	0.0104
Stmt_cvg	4	3	41.43	4e-9	2.91	-0.021
S_c_tc_num	5	3	210.1	<2e-16	13.2	0.021
Branch_c_t_n	3.879	4	236.8	<2e-16	18.6	0.0611
Branch_cvg	1.98	2	17.98	<2e-16	1.37	-0.021
Tc_num_loc	3.278	4	1.645	0.00029	21.7	0.174
Tc_num_comple	3.219	4	1.525	4.71e-07	20.1	0.161

### 3.3. GAM 模型的多因素分析

通过 pearson 相关系数[10]以及单因素影响分析可得到, loc (代码行数)与 complexity (复杂度)、tc\_num\_loc (测试资源分配特征)与 tc\_num\_complexity 与 tc\_num (测试用例的个数)、stmt\_cvg (语句覆盖率)与 branch\_cvg (分支覆盖率)、stmt\_cvg\_tc\_num (每个测试用例对语句覆盖的贡献)与 branch\_cvg\_tc\_num (每个测试用例对分支覆盖的贡献)的相关性高达 99.9%以上,且两者之间通过单因素分析的结果也高度相似,故判定其为可以互相代替的变量。

将单因素分析与去除共线性之后且经过显著性检验的七个影响变量 stage (测试阶段)、weeks (测试周)、requiemnt\_chg (需求变更量)、loc (代码行数)、tc\_num (测试用例的个数)、stmt\_cvg (语句覆盖率)、branch\_cvg(分支覆盖率)作为解释变量,将失效数作为响应变量,进行多影响因素与失效数变化间的 GAM 模型构建拟合,模型同样采用恒等联系函数作为连接函数,平滑回归项采用样条平滑函数估计,模型拟合结果如表 5 所示。

**Table 5.** GAM model hypothesis test results of failure number changes and multiple influencing factors  
**表 5.** 失效数变化与多影响因素的 GAM 模型假设检验结果

指标	Stage	Week	Req_chg	loc	Tc_num	Stmt_cvg	Branch_cvg
估计自由度	2.309	1.129	1	1	6.602	3.414	1
参考自由度	2.888	1.222	1	1	7.685	4.204	1
F	4.500	6.932	2.595	0.0172	3.562	1.865	-1.692
P-value	0.0093	0.012	0.0124	0.01640	0.001	0.0117	0.845

多因素分析表明, branch\_cvg (分支覆盖率)的 p 值>0.05, 故不能假设他对失效数有显著影响, loc (代码行数)、stmt\_cvg (语句覆盖率)、requiemnt\_chg (需求变更量)在 P 值<0.05 水平下显著影响失效数变化,其余驱动因素在 P 值<0.01 水平下显著影响失效数变化。模型判定系数为 0.743, 方差解释率为 75.5%, 表明模型拟合程度较高, 通过显著性检验的 6 个因素对失效数的解释率较高。因此可以确定他们对失效数变化有显著性影响。其中, 模型判定系数[11]与方差解释率[12]定义如下:

$$\text{模型判定系数: } R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SSR + SSE} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

$$\text{第 } j \text{ 个变量的方差解释率: } \frac{\sigma_i^2}{\sum_j \sigma_j^2}$$

### 3.4. GAM 模型的检验

#### 3.4.1. 影响因素对失效数变化作用效应诊断

通过对多影响因素(解释变量)与失效数变化间建立 GAM 模型, 获得解释变量的平滑回归函数, 并得到影响因素对失效数影响效应图, 如图 2 所示。图中虚线表示拟合可加函数的逐点标准差, 即可信区间的上下限, 实线代表失效数的平滑拟合曲线, 横坐标表示各解释变量的实测值, 纵坐标表示解释变量对失效数的平滑拟合值。纵坐标括号数值代表估计自由度值。结果表明, 失效数变化与 loc (代码行数)、requiriement\_chg (需求变更量)间的相关性表现为线性相关, 随着 loc (代码行数)或 requiriement\_chg (需求变更量)数值递增时, 失效数变化也呈现递增趋势, 当 stage (测试阶段)小于 4 时, 失效数变化呈现单增趋势, 其他变量呈现了非线性关系, 并且波动较大, 例如, 对于变量 tc\_num (测试用例的个数)而言, 当超过 200 后, 失效数的变化呈现快速减少的趋势, 这是由于当测试用例增加到一定数量以后, 软件的可靠性就会随之增加, 失效数变化就会随之减少。

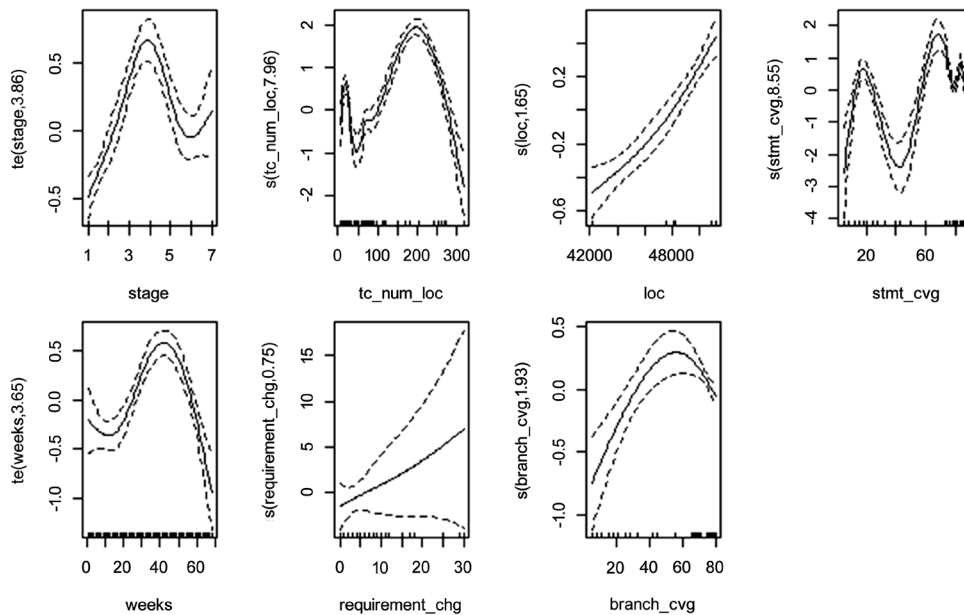


Figure 2. Effect of influencing factors on the change of failure number

图 2. 影响因素对失效数变化的影响效应

#### 3.4.2. 模型交叉检验

十折交叉验证[13], 英文名叫做 10-fold cross-validation, 用来测试算法准确性。是常用的测试方法。将数据集分成十份, 轮流将其中 9 份作为训练数据, 1 份作为测试数据, 进行试验。每次试验都会得出相应的正确率(或差错率)。10 次的结果的正确率(或差错率)的平均值作为对算法精度的估计。

在通过模型假设检验及多曲线共线性分析后得到的解释变量与失效数构建的到的 GAM 模型的拟合

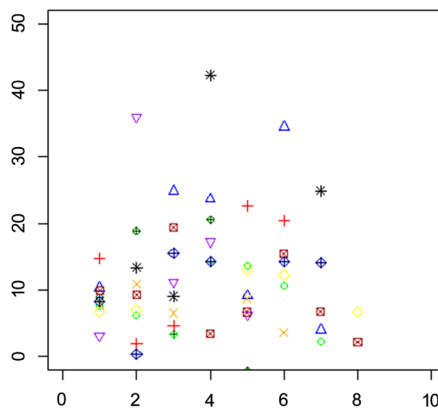
优度较好，影响因素对失效数变化的解释率较高。同时，为了避免过拟合现象，采用十折交叉验证[15]对其建立的 GAM 模型进行验证。在给定 GAM 建模数据中，拿出 9 份数据进行建模，留 1 份数据用刚建立的模型进行预报误差。十折交叉验证的预测结果如图 3 所示，验证结果如表 6 所示，其中利用每份数据集的平方和、方差解释率、以及 RMSE [14]作为检验标准，可以看到预测值与实际值的差值较小，结果说明，GAM 模型的拟合效果较好，准确率较高。其中，RMSE 的定义如下：

$$RMSE = \sqrt{\frac{\sum_{i=0}^n (y_i - \hat{y}_i)^2}{n}}$$

**Table 6.** Ten-fold cross-validation of the failure number GAM model

**表 6.** 失效数 GAM 模型的十折交叉验证

数据集	1	2	3	4	5	6	7	8	9	10
平方和	0.51	0.41	0.49	0.53	0.48	0.42	0.42	0.47	0.51	0.48
方差解释率	70	64.9	66.6	72	69.6	64	70	69.9	67.8	71.5
RMSE	2.78	4.05	3.67	2.34	3.77	5.2	4.2	3.98	4.8	3.77



**Figure 3.** The number of failures is 10% of the predicted value of cross-validation  
**图 3.** 失效数 10 折交叉验证的预测值

### 3.5. 带有交互项的 GAM 模型分析

失效数变化受多个因素的影响，各解释变量之间交互作用对失效数变化也会产生强烈的影响。在第二节中介绍的 GAM 模型中引入两个因素间的交互作用项，可得带有交互作用项的 GAM 模型如下

$$g(u) = a + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p) + f_{1,2}(X_{1,2}) + \dots + f_{p-1,p}(X_{p-1,p})$$

其中， $f_{i,j}(X_{i,j})$ 为第  $i$  个因素和第  $j$  个因素的交互作用项。

把带有交互作用项的 GAM 模型用于第三节给出的数据，输出结果如表 7 所示。用于检验交互作用  $tc\_num$  (测试用例的个数)  $\times loc$  (代码行数)、 $tc\_num$  (测试用例的个数)  $\times branch\_cvg$  (分支覆盖率)、 $tc\_num$  (测试用例的个数)  $\times stmt\_cvg$  (语句覆盖率)的 P 值均小于 0.05,所以可以认为这些交互作用影响是显著的。

从检验结果还可以发现，交互作用项主要发生在  $tc\_num$  (测试用例的个数)与  $loc$  (代码行数)、 $branch\_cvg$  (分支覆盖率)、 $stmt\_cvg$  (语句覆盖率)之间，这也与失效数变化主要受测试用例个数与测试的复杂度(包括代码行数、分支覆盖率、语句覆盖率)的交互作用过程影响的特征吻合。

对具有显著性影响且具有统计学意义的交互作用进行可视化绘图，如图 4 所示。这种图显示了失效



数变化在不同自变量维度下的变化特征。在图 4 中存在统计学意义的为图 4(a)、图 4(d)和图 4(f)，由图 4(a)可知，当 loc (代码行数)小于 48000 时，失效数是不受 tc\_num (测试用例的个数)变化的影响的，当 loc(代码行数)大于 48000 时，失效数才发生了变化，由图四(d)可知，当固定 branch\_cvg (分支覆盖率)时，随着 tc\_num (测试用例的个数)的增加，失效数在一定范围内波动变化，但是当 tc\_num (测试用例的个数)超过 1000 时，失效数呈现快速减小的趋势。图 4(e)呈现的趋势与图 4(d)相似。

本研究结果表明，以上提到的三项交互作用项与单因素模型相比较，在方差解释率以及调整方差系数方面都有所提高。且对于 branch\_cvg(分支覆盖率)在 GAM 模型中的影响不显著，但是引入 tc\_num (测试用例的个数) × branch\_cvg (分支覆盖率)交互项后，对失效数变化有显著性影响。通过在 GAM 模型中引入交互作用项的分析，可以有效的定量分析影响失效数变化的重要特征，同时也可以得出失效数变化不但受各个因素的影响，而且受因素间的交互作用的影响。因此，今后还可以考虑三个因素的交互作用。

最后，建立带有交互作用项的 GAM 模型，模型结果如表 8 所示。得到模型的方差解释为 97.9%，相较于 GAM 模型的方法解释率，提高了 29.7%。判定系数  $R^2 = 0.989$ ，得到模型效果良好，模型拟合程度较优。

**Table 7.** GAM model hypothesis test results of interaction factors and failure numbers

**表 7.** 影响因素交互作用与失效数的 GAM 模型假设检验结果

交叉项	Tc_num-loc	loc—stmt_cvg	Loc—branch_cvg
估计自由度	20.46	2	2
参考自由度	24.26	2	2
F	2.583	1.269	1.299
P-value	0.00225	0.288	0.28
方差解释率	62.3%	3.765	3.84%
$R^2$	0.457	0.00797	0.00885
交叉项	Tc_num—branch_cvg	Tc_num—stmt_cvg	Stmt_cvg—branch_cvg
估计自由度	14.73	14.44	2
参考自由度	18.59	18.27	2
F	2,3	2.193	0.925
P-value	0.00945	0.0137	0.402
方差解释率	50.1%	48.9%	2.77%
$R^2$	0.361	0.346	-0.00223

**Table 8.** Final GAM model parameter table

**表 8.** 最终 GAM 模型参数表

指标	Tc_num-loc	Stage	Weeks	Stmt_cvg-tc_num	Branch-tc_num	complexity
估计自由度	22.1	4	4	7,4	7.4	1
参考自由度	23	4	4	9	9	1
F	70	0.407	11.688	30.022	28	1.45
P	6.70e-7	0.040	0.003	5.9e-09	9.41e-07	0.03

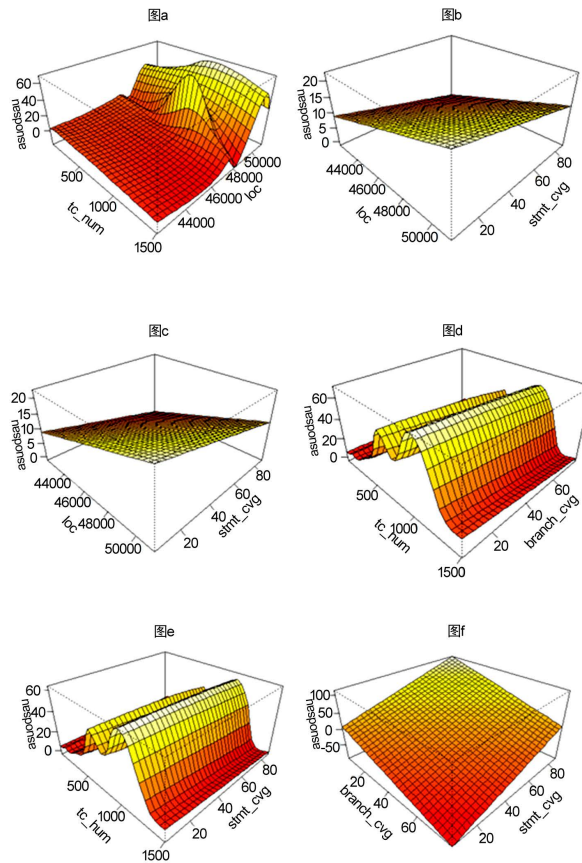


Figure 4. Interaction diagram  
图 4. 交互作用图

### 3.6. 机器学习模型与带有交互项的 GAM 模型比较

为了比较带有交互项的 GAM 模型与现有的机器学习模型[15]，如岭回归、lasso 回归、ElasticNet 回归、支持向量机等预测效果，对 3.1 节中的 68 组样本数据进行随机划分，选取 70%样本作为训练集，30%样本作为测试集，对测试集中的样本，使用 RMSE (root mean square error)指标：

$$RMSE = \sqrt{\frac{\sum_{i=0}^n (y_i - \hat{y}_i)^2}{n}}$$

来评价模型的预测效果。计算结果如表 9 所示。

Table 9. Comparison of machine learning methods and GAM models with interactive terms  
表 9. 机器学习方法与带有交互项的 GAM 模型比较表

方法	RMSE	相对 RMSE
岭回归	12.76	-24.7%
Lasso 回归	14.38	-33.2%
ElasticNet 回归	14.58	-34.1%
支持向量机	10.37	-7.4%
带有交互项的 GAM 模型	9.6	-

表 9 中的第二列表明, 带有交互项的 GAM 模型的 RMSE 值最低, 相比 ElasticNet 回归 RMSE 降低了 34.1%, 相比较好的支持向量机方法仍降低了 7.4%。由此可以得到, 带有交互项的 GAM 模型对数据的预测效果要优于其他模型。

#### 4. 结论

对 3.1 节中的数据使用带有交互项的 GAM 模型进行分析, 发现在模型中添加交互项可以提高模型的拟合和预测效果。例如方差解释率与模型的调整判定系数相对于没有交互项的 GAM 模型都有所提高。用 GAM 模型对数据进行因素分析时发现, branch\_cvg (分支覆盖率) 并没有显著性影响, 但是在模型中添加了交互项后, tc\_num (测试用例的个数) branch\_cvg (分支覆盖率) 的交互项对失效数变化有显著性影响, 这也是符合工程实际应用的。最后将带有交互项的 GAM 模型与其他机器学习模型进行比较, 得到此模型的预测结果的 RMSE 要优于其他机器学习模型, 因此说明带有交互作用的 GAM 模型可以分析软件开发中某些因素的交互作用。

#### 参考文献

- [1] 杨敏娟. 北京市大气污染与居民心脑血管死亡的时间序列分析[J]. 环境科学, 2015, 36(10): 122.
- [2] Humphrey, W.S. (2001) The Future of Software Engineering: I.
- [3] Musa, J.D. (1992) The Operational Profile in Software Reliability Engineering: An Overview. *Proceedings of 3rd International Symposium on Software Reliability Engineering*, Research Triangle Park, 7-10 October 1992, 140-154.
- [4] 李丽霞, 郗艳晖, 周舒冬, 等. 广义加性模型及其应用[J]. 中国卫生统计, 2007, 24(3): 243-244.
- [5] Hastie, T. and Tibshirani, R. (1990) *Generalized Additive Models*. Chapman and Hall, London.
- [6] 贾林, 王彤. GAM 共曲线性及其在空气污染问题研究中的应用[J]. 第四军医大学学报, 2005, 26(3): 280-283.
- [7] Humphrey, W.S. (2001) The Future of Software Engineering: I. Watts New Column, News at SEI, Vol. 4. No. 1, March.
- [8] IEEE Std 1633-2008 (2008) IEEE Recommended Practice on Software Reliability. IEEE Reliability Society, New York.
- [9] 郑骏. 软件可靠性建模的发展和存在的问题[J]. 计算机科学, 1995, 22(6): 74-78.
- [10] 刘超, 金茂忠. 软件测试过程的基本模型 POCERM[J]. 北京航空航天大学学报, 1997, 23(1): 56-60.
- [11] 陈光宇, 黄锡滋. 软件可靠性学科发展现状及展望[J]. 电子科技大学学报: 社科版, 2004, 4(3): 99-102.
- [12] 刘云, 赵玮. 软件可靠性研究与进展[J]. 微机发展, 2003, 13(2): 12-15.
- [13] 杨玉丽. 软件可靠性的研究现状与展望[J]. 电脑知识与技术, 2010, 6(1): 128-129.
- [14] Xie, W., Wu, J., Sun, H. and Zhang, L. (2017) A Distribution-Level Combinational Model to Improve Reliability Prediction Accuracy. *International Journal of Performability Engineering*, **13**, 832-843. <https://doi.org/10.23940/ijpe.17.06.p5.832843>
- [15] Wu, J., Ali, S., Yue, T., Tian, J. and Liu, C. (2016) Assessing the Quality of Industrial Avionics Software: An Extensive Empirical Evaluation. *Empirical Software Engineering*, **22**, 1634-1683. <https://doi.org/10.1007/s10664-016-9440-x>

**知网检索的两种方式：**

1. 打开知网首页：<http://cnki.net/>，点击页面中“外文资源总库 CNKI SCHOLAR”，跳转至：<http://scholar.cnki.net/new>，搜索框内直接输入文章标题，即可查询；  
或点击“高级检索”，下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版：<http://www.cnki.net/old/>，左侧选择“国际文献总库”进入，搜索框直接输入文章标题，即可查询。

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：[csa@hanspub.org](mailto:csa@hanspub.org)