

# 基于模糊推理的软件测试度量方法

涂玲 周彦晖 张为群 周亚洲

(西南大学计算机与信息科学学院 重庆 400715)

**摘要** 如何形成软件测试有效的成本效益策略一直以来都是软件工程的重要研究内容。许多研究者致力于软件测试的有效性和质量度量的研究并已取得大量成果。但是,软件测试中最重要的问题之一——测试度量中内在的不确定和不相关的关系仍然没有解决。由此,提出一种新的基于模糊逻辑的软件测试质量和测试有效性度量方法,探讨了基于软件质量特性和相似性推理的方法,试图解决不同测试工作任务之间的测试质量和有效性一致的问题。同时提供实验的结果证明提出的方法的可行性。

**关键词** 模糊推理,软件测试,ISO/IEC 9126,测试度量

**中图分类号** TP306+.2 **文献标识码** A

## Fuzzy Logic Based Metric in Software Testing

TU Ling ZHOU Yan-hui ZHANG Wei-qun ZHOU Ya-zhou

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)

**Abstract** How to provide cost-effective strategies for software testing has been one of the research focuses in software engineering for a long time. Many researchers in software engineering have addressed the effectiveness and quality metric of Software Testing, and many interesting results have been obtained. However, one issue of paramount importance in software testing—the intrinsic imprecise and uncertain relationships within testing metrics—is left unaddressed. To this end, a new quality and effectiveness measurement based on fuzzy logic was proposed. The software quality features and analogy-based reasoning were discussed, which can deal with quality and effectiveness consistency between different test projects. Experimental results were also provided to verify the proposed measurement.

**Keywords** Fuzzy logic, Software testing, ISO/IEC 9126, Testing metric

## 1 概述

软件工业面临软件测试成本持续增长的问题,因此在测试中改善效率以及提高有效性和质量是降低测试成本的两个主要策略。

20世纪80年代有许多研究关注于改善测试效率,文献[1-3]中描述了几种自动产生测试用例的白盒测试方法。在软件工程中,被当作管理问题来看待的测试的有效性和质量度量经常由经验判断和定性分析来解决。为了定量管理,在文献[4-6]中定义了许多好的软件测试度量,其主要思想是通过向代码中注入模拟故障并在测试程序的过程中量化故障的检出率,项目管理人员依靠注入错误和测试者发现的软件本身错误的比率这一基本度量来判断测试的有效性和质量。这些传统的方法经过进一步的发展已日趋成熟<sup>[7]</sup>,但是,因为源代码对于测试者透明的原因,这样的测试度量难以用于黑盒测试,另一方面,错误注入过程产生的成本也对软件开发成本产生负面影响。

我们认为,希望获得更多关于测试质量的本质质量的主要障碍是一些模糊和不确定的关系。例如测试工程师的经

验、软件领域知识和质量意识等要素和关联,由于其内在特性而难以表示和测量。直到现在,软件工程(SE)领域中为了表示这些模糊的要素和关联仍然经常使用固定的数值区间。经过这样变换后的数值(精确)表示往往与人们对这些要素的理解(模糊、不确定)方式不同,影响了度量的准确性和有效性。因此,建立一种新的度量来处理这种模糊关系同时更好地适应黑盒测试就显得尤为必要。

一些人工智能技术比如模糊逻辑能有效处理模糊和不确定性,在SE领域被广泛使用<sup>[8,9]</sup>,甚至在白盒测试中被用于产生测试用例以改善效率。将模糊逻辑用于软件测试度量是一种新的尝试。这些精确有效的度量将提高测试管理的能力,同时降低测试的成本和工作量。产生成本合理的以及结果可靠准确的有效测试是优良度量的标准。

## 2 基于度量的模糊逻辑

### 2.1 在测试用例中使用的软件质量特性

针对软件产品质量的问题,国际标准化组织第一联合技术委员会和国际电工技术委员会出版了一系列软件产品质量标准,如著名的ISO/IEC 9126<sup>[11,12]</sup>。ISO/IEC 9126定义了

到稿日期:2008-12-05 返修日期:2009-01-21

涂玲 工程师,主要研究方向为软件测试,E-mail: tuling@swu.edu.cn;周彦晖 副教授,主要研究方向为软件测试、软件工程;张为群 教授,主要研究方向为软件测试、软件工程。

包含软件产品质量 6 个特性以及 27 个子特性,如表 1 所列。

在典型的验证测试项目中,一系列关注软件产品不同功能和不同质量特性的测试用例构成测试项目。为了达到适当的测试强度,测试用例应在不同的功能和软件的质量特征上重叠,这种关注和重复的程度可以视为每个测试用例和功能和质量特性的关联关系。测试用例与功能的关联比较明确,但是与质量特性之间的关联就比较模糊。如果某些测试用例与某些质量特性无关联或关联太少,则在这些质量特性上的测试强度不够;若关联过多,则测试成本增加而且效率不佳。虽然这样的关联关系是模糊且不确定的,它仍然能够被语义值和模糊集度量。显然,如果测试项目的测试用例与质量特性的关联程度具有更多的相似性,则它们具有更多在测试质量、成本和效率上的相似性。

表 1 ISO/IEC9126 中的特征属性与子特征属性

质量特性	子特征
功能性	适应性,准确性,互操作性,安全性,功能依从性
可靠性	成熟性,容错性,可恢复性,可靠依从性
可用性	可理解性,可学习性,可操作性,吸引力,可用依从性
效率	时间特性,资源利用率,效率依从性
可维护性	可分析性,可改变性,稳定性,可测试性,维护依从性
可移植性	适应性,可安装性,可替代性,共存性,移植依从性

## 2.2 测试项目的模糊集表示

这一步是依据相应的软件产品质量特性描述软件测试项目特征,我们采用模糊权重分配算法来处理测试用例和软件质量特性之间的关系。

一个权重函数用来为 ISO/IEC 9126 中每个质量特性  $C_j$  分配权重  $W_{i,j}$ ,在基于领域的文档关联计算中经常使用这样的典型权重函数,称为特征权重函数(Feature Weighting Function,简称 FWF)。根据文献[14]中关于 FWF 的详细介绍,我们得出如下定义:

定义 1  $W_{i,j}$  是质量特征属性  $C_j$  的权重,定义为测试用例  $i$  中与  $C_j$  关联的测试用例步骤数与整个测试用例中  $C_j$  的测试步骤数的比率。总的权重  $w_j$  为含有  $n$  个测试用例的测试项目中  $W_{i,j}$  的和。公式如下:

$$w_j = \sum_{i=1}^n w_{i,j} = \sum_{i=1}^n (\text{step}_{i,c_j} / \text{step}_{c_j} \times 100) \quad (1)$$

在上面的公式中,  $\text{step}_{i,c_j}$  表示和  $C_j$  相关的第  $i$  个测试用例的步骤数。

考虑如下说法,测试步骤与质量特性的相关性为 low, nominal, average 和 high,语言变量测试步骤与质量特性的相关性取值为 low, nominal, average, high。为了方便表达,我们将质量特性名称直接视为语言变量,从而提出如图 1 所示的模糊集成员函数表示方法,其中权重对应于式(1)。

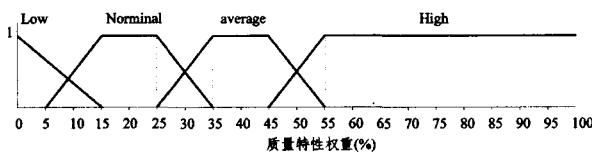


图 1 模糊集成员函数

举例说明,表 2 显示了权重,语义值和基于 FWF 函数成员值的一部分计算结果。

表 2 测试用例的模糊数据和质量特征属性

质量特征属性	权重	语义值	$\mu$
--------	----	-----	-------

准确性	91.3	High	1
准确性	91.3	Average	0
安全性	34.3	Average	0.93
安全性	34.3	Nominal	0.07
.....	.....	.....	.....

## 2.3 测试质量相似性的模糊推理

通过模糊集来表示测试用例和质量特性的相关性由语义值来划分。对于特性  $Q$ (所有质量特性的全集或子集),  $C_j$  是  $Q$  中第  $j$  个特性,表示一个语言变量。  $A_k$  是和第  $k$  个与属性语义值相关的模糊集。  $\mu_{A_k}$  表示  $A_k$  的成员函数。

在模糊集理论中,为了比较  $P_m$  和  $P_n$  两个测试项目中与语言变量  $C_j$  相关联的模糊集的相似性,相关联的模糊集必须具有含有  $C_j(P_m)$  和  $C_j(P_n)$  两个变量的成员函数。这个模糊集涉及一个模糊关系,我们在模糊关系中定义 if-then 规则如下:

定义 2  $R_{C_j}^i(P_m, P_n)$  表示和两个测试项目  $P_m, P_n$  与模糊语言变量  $C_j$  (质量特性)相关的一个模糊关系,  $A_k$  为其特定值,这个关系蕴含如下 If-then 规则:

$$\text{if } C_j(P_m) \text{ is } A_k \text{ then } C_j(P_n) \text{ is } A_k \quad (2)$$

因此,对每个语言变量  $C_j$ ,相关的模糊关系都蕴含了这个 if-then 规则。基于这样的规则,相似性可由模糊关系中每个语言值计算得出。给出如下定义:

定义 3  $S_{C_j}^i(P_m, P_n)$  表示两个测试项目  $P_m$  和  $P_n$  和语言变量  $C_j$  相关的第  $K$  个模糊集的相似性,给出如下公式:

$$S_{C_j}^i(P_m, P_n) = \text{simp}(\mu_{A_k}(P_m), \mu_{A_k}(P_n)) \quad (3)$$

其中,  $\text{simp}$  是计算两个参数相似性的不确定函数。

模糊关系  $R_{C_j}^i(P_m, P_n)$  和每个模糊变量  $C_j$  的语义值相关。这个关系的组合是和变量  $C_j$  的所有语义值相关的模糊关系,用  $R_{C_j}^i(P_m, P_n)$  表示。随后在  $R_{C_j}^i(P_m, P_n)$  中组合应用 if-then 规则形成聚集。基于这个聚集,我们定义两个测试项目和  $C_j$  的所有语义值相关的相似性如下:

定义 4 令  $S_{C_j}^i(P_m, P_n)$  表示两个测试项目  $P_m$  和  $P_n$  和语言变量  $C_j$  的所有语义值相关的相似性,则:

$$S_{C_j}^i(P_m, P_n) = \text{aggr}(S_{C_j}^i(P_m, P_n)) \quad (4)$$

其中,  $\text{aggr}(x)$  是计算参数  $x$  的聚合性的不确定函数。

在式(3)和式(4)中,  $\text{simp}()$  和  $\text{aggr}()$  还未确定,我们可以基于应用环境和上下文使用可能函数的集合。我们倾向于如下简单的函数:

$$\text{simp}(x, y) = \begin{cases} \min(x, y) \\ x \times y \end{cases}$$

$$\text{aggr}(x) = \begin{cases} \max(x, y) \\ \sum x \end{cases}$$

然后,在式(3)和式(4)中的这些函数的可能组合如下:

$$S_{C_j}^i(P_m, P_n) = \begin{cases} \max(\min(\mu_{A_k}^i(P_m), \mu_{A_k}^i(P_n))) & \text{max-min aggregation} \\ \max(\mu_{A_k}^i(P_m) \times \mu_{A_k}^i(P_n)) & \text{max-product aggregation} \\ \sum_k \mu_{A_k}^i(P_m) \times \mu_{A_k}^i(P_n) & \text{sum-product aggregation} \\ \sum_k \min(\mu_{A_k}^i(P_m), \mu_{A_k}^i(P_n)) & \text{sum-min aggregation} \end{cases}$$

当计算测试项目相似性的时候,这 4 个聚合函数并不都需要。在文本表达式中,它们应该满足一些直观的、明显的角色需要。如果它和角色相矛盾,函数应该被忽略。文献[8]中

提出了一些原理来证明聚合函数。本文选择 3 个原理来证明：

$$S^C_j(P_m, P_n) \neq 0 \Leftrightarrow \exists A_{ik} \text{ 使 } \mu_{A_{ik}}(P_m) \neq 0 \text{ 且 } \mu_{A_{ik}}(P_n) \neq 0 \quad (5)$$

$$S(P, P_i) \leq S(P, P) \quad (6)$$

$$S^C_j(P_m, P_n) \leq 1 \quad (7)$$

函数聚合性证明结果如表 3 所列,从结果看出,只有 max-min 聚合函数满足式(5)、(6)、(7)这些原理。在我们的度量中采用了这个聚合函数。

表 3 验证聚合函数的结果

聚合函数	公式(5)	公式(6)	公式(7)
max-min	满足	满足	满足
max-product	满足	不满足	满足
sum-min	满足	满足	不满足
sum-product	满足	不满足	不满足

### 3 应用示例

下面举例说明表 2 的计算过程。假设如下测试用例用于测试一个典型 ERP 软件的用户登录过程,该软件使用 J2EE 开发,后台数据库为 Oracle 9i。

从表 4 所列的实例中可以看出 1—5 步均需要系统功能正常,对操作正确响应。而第 6 步与所测试系统的软件不直接相关,因此共 5 步与准确性(Accuracy)质量要素相关。而第 2—6 步均涉及到系统安全性(Security)质量要素。虽然第 6 步不是所测试软件的直接功能,但如果系统未对后台数据库进行必要的安全设置,将使整个软件的安全性崩溃。

因此,在式(1)中,Step20060040-fun-03, Accuracy = 5, Step20060040-fun-03, Security = 5, 并应该将它们累计到 stepAccuracy 和 stepSecurity 中。

表 4 实例

用例编号	20060040-fun-03
用例名称	用户登录测试
测试目标	判断系统是否能够完成授权用户登录,并提供相应的安全性。
测试用例步骤与数据	<ol style="list-style-type: none"> <li>1. 已存在用户登录系统,使用正确用户名、正确口令:用户名:sys,口令:bestOTOyou</li> <li>2. 已存在用户登录系统,使用错误用户名、错误口令:用户名:sys,口令:1223332</li> <li>3. 不存在用户登录系统,使用不存在用户名。用户名:sys-1,口令:123</li> <li>4. 使用构造的特殊口令串登录,可能使口令判断语句结果为永真。用户名:sys,口令:123 or 1=1。</li> <li>5. 使用超长字符串口令登录。用户名:sys,口令:粘贴长度为 2049 个字符的串。</li> <li>6. 使用数据库管理器缺省口令直接连接后台 Oracle 数据库,用户名:system,口令:manager</li> </ol>
系统的预期结果	<ol style="list-style-type: none"> <li>1. 登录进入系统</li> <li>2. 提示口令错误</li> <li>3. 提示用户不存在</li> <li>4. 提示口令错误</li> <li>5. 系统提示口令错误,不发生系统崩溃、溢出</li> <li>6. 后台数据库已改变所有缺省口令,不能登录后台数据库</li> </ol>

对所有的测试用例均可以由测试人员在编写测试用例时按照各质量特性(Quality Characteristics)进行计数,如假设某个测试的所有测试用例的总步骤数为 531, Accuracy、Security 质量特性的权值计算如表 5 所列。

利用以上数据代入图 1 的模糊成员函数,得到如表 2 的成员关系值。

表 5 质量特征的权重

Quality Characteristic	Step C <sub>j</sub>	W <sub>j</sub>
准确性(Accuracy)	485	91.3%
安全性(Security)	182	34.3%

### 4 实验结果

为了确保实验结果如实反映测试过程,我们在第三方软件测试组织的实际测试项目中选择测试数据。实验依据三种样本数据设计成 3 个类别,以检验度量的不同方面。

ISO/IEC 9126 中的特征属性能适合不同的需求,我们选择 12 个适合于大多数软件的能够被可执行的步骤测试的特征属性。

在第一类别中,两个项目相似性的结果如图 2(a)中所示,它表示每个质量特征属性相似性的值。一些相似性值较低的属性,比如可替代性(11)和时间行为(7),由于软件本身需求的差异,并不代表相似性的巨大差异。当两个成员函数的值都小于 1 并且集中于一个很小的区间时,max-min 聚合函数将产生小于 1 的相似性值。在模糊理论中,它叫做 1 的临界区。因此,正如我们所预测的,在属于同一个测试人员完成的两个测试项目中,每个质量特性的度量非常相似,而两个测试项目在总体上也有很高的相似性。

表 6 选择的质量特性

1 准确性 (accuracy)	2 互操作性 (interoperability)	3 安全性 (security)
4 容错性 (fault tolerance)	5 可恢复性 (recoverability)	6 可操作性 (operability)
7 时间行为 (time behavior)	8 资源利用率 (resource utilization)	9 稳定性 (stability)
10 可安装 (install ability)	11 可替换性 (replace ability)	12 共存性 (coexistence)

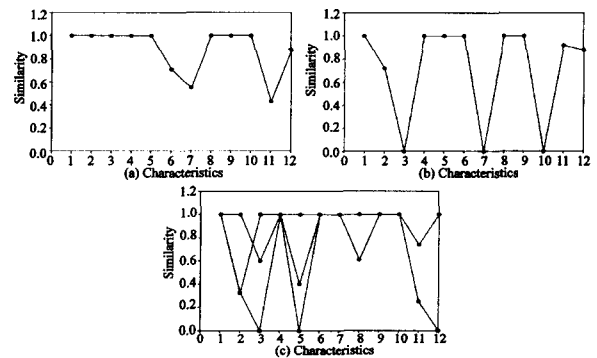


图 2 实验结果分析

图 2(a)表示两个测试项目属于一个测试人员,它们被随机选取,且软件类型相同,(b)表示两个测试项目属于一个测试人员,它们被随机选取,且软件类型不相同,(c)表示三个测试项目属于不同的测试人员,它们被随机选取,且软件类型不相同。

在分类 2 中,两个测试项目的目标是两个不同类型的软件产品。一个是基于 ERP 的网络应用,另一个是智能桌面游戏。首先,安全性特征属性(3)的相似值为 0,有趣的是在桌面游戏测试用例中,安全性特征属性(3)的值为 0。也许,测试者认为没有必要在桌面游戏中测试安全性。其二,当我们检测另外一个质量特性——时间行为时得出如下结果:在游戏测试中,只有一个测试用例——游戏加载时间测试和时间

性能相关,但在 ERP 软件的测试用例中存在打开文件时间、数据库更新时间和访问时间等 12 个与时间行为相关。最后一个偏离的凸点表示的质量特性是可安装(10),显然它的值偏离是由于这两个软件安装的复杂程度及其不同引起。

为了完成第三个实验,我们随机选取三个基于 GIS 的软件产品。我们为一些测试后有较高评价的项目中创建一个档案资料库。通过比较档案资料库的数据,我们计算 3 个项目的质量测试值,如图 2(c)所示。显而易见的是,质量总是在那些无功能的属性上波动,如交互性、安全性和共存性。

**结束语** 本文关注于如何改善测试成本和质量这两个软件测试中最重要的问题,提出一种新的度量方法。采用 ISO/IEC 9126 中的特征属性作为统一的质量特性。引入模糊集来表示测试用例和质量特征之间的模糊和不确定的相互关系,为测试案例中各成分之间不精确的、不确定的相互关系带来更合理的表示。采用新的度量方法,在不提高成本的情况下,软件测试的质量和有效性更容易测量。特别是在早期的测试中,它能改善质量评估和测量的能力。它可以更进一步地用来解决测试管理中和质量相关的问题,比如质量对比和质量适应性分析。

实验的结果进一步显示本文提出的度量方法能有效测量和评估测试项目的质量和有效性,能够作为测试管理的工具。值得注意的是,软件的类型在实验的度量中有显著的干扰作用,因此,这种度量方法更适用于通用的软件类型。

## 参 考 文 献

- [1] DeMillo R A, Offlutt A J. Constraint-Based Automatic Test Data Generation[J]. IEEE Transactions on Software Engineering, 1991, 17(9): 900-910
- [2] Dustin E, Rashka J, Paul J. Automated Software Testing: Introduction, Management, and Performance[M]. Addison-Wesley, 1999
- [3] Weyuker E, Goradia T, Singh A. Automatically Generating Test

Data from a Boolean Specification[J]. IEEE Transactions on Software Engineering, 1994, 20(5): 353-363

- [4] Kan S H. Metrics and Models in Software Quality Engineering, Second Edition[M]. Addison-Wesley, 2003
- [5] Potter R W. The Art of Measurement: Theory and Practice [M]. Prentice Hall PTR, 2000
- [6] Fenton N E, Pfleeger S L. Software Metrics: A Rigorous and Practical Approach, Second Edition[M]. PWS Publishing Company, 1997
- [7] Voas J M, McGraw G. Software Fault Injection: Inoculating Programs against Errors[M]. Wiley, 1998
- [8] Idri A, Abran A. A Fuzzy Logic Based Measures For Software Project Similarity: Validation and Possible Improvements[C]// Proceedings of the 7th International Symposium on Software Metrics. England, UK, IEEE Computer Society, 2001: 85-96
- [9] Bosc P, Damiani E E, Fugini M G. Fuzzy Service Selection in a Distributed Object-Orient Environment[J]. IEEE Transactions on Fuzzy System, 2001, 9(5): 682-698
- [10] Last M, et al. The Data Mining Approach to Automated Software Testing[C]// SIGKDD'03. ACM 1-58113-737-0/03/0008. August 2003
- [11] ISO/IEC JTC1/SC7. Software engineering-Product quality-Quality Model, ISO, 2001
- [12] ISO/IEC JTC1/SC7. Software engineering -Product quality-External quality metrics, 2003
- [13] Salton G, Buckley C. Term Weighting Approaches in Automatic Text Retrieval[J]. Information Processing and Management, 1998, 24(5): 513-523
- [14] Damiani E, Fugini M G. Automatic Thesaurus Construction Supporting Fuzzy Retrieval of Reusable Components[C]// Proceedings of ACM SIG-APP Conference on Applied Computing (SAC'95). Nashville, TE, USA, 1995
- [15] 黄贤英, 张丽芳. 基于粒子群优化的模糊聚类算法[J]. 重庆工学院: 自然科学版, 2008, 22(11): 120-123

(上接第 73 页)

络拓扑进行仿真。

## 参 考 文 献

- [1] The network simulator - ns-2[OL]. www.isi.edu/nsnam/ns/
- [2] Elias P, Feinstein A, Shannon C. A note on the maximum flow through a network[J]. IEEE Transactions on Information Theory, 1956, 2(4): 117-119
- [3] Ahlswede R, Cai N, Li Shuo-Yen R, et al. Network information flow[J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204-1216
- [4] 彭木根, 王月新, 王文博. 无线自组织网络的网络编码技术[J]. 中兴通讯技术, 2007, 13(004): 56-60
- [5] 熊志强, 黄佳庆, 刘威, 等. 无线网络编码综述[J]. 计算机科学, 2007, 34(3): 6-9
- [6] 马冠骏, 许胤龙, 林明宏, 等. 基于网络编码的 P2P 内容分发性能分析[J]. 中国科学技术大学学报, 2006, 36(011): 1237-1240
- [7] Ma Guan-jun, Xu Yin-long, Lin Ming-hong, et al. A content distribution system based on sparse linear network coding[C]//

Third Workshop on Network Coding(Netcod 2007). 2007

- [8] Gkantsidis C, Rodriguez P. Network coding for large scale content distribution[C]// INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, 2005, 4
- [9] Wang M, Li B. How practical is network coding[C]// Proc. of the Fourteenth IEEE International Workshop on Quality of Service(IWQoS 2006). 2006: 274-278
- [10] Li S, Cai R. Linear network coding [J]. IEEE Transactions on Information Theory, 2003, 49(2): 371-381
- [11] Koetter R, Médard M. An algebraic approach to network coding [J]. IEEE/ACM Transactions on Networking(TON), 2003, 11(5): 782-795
- [12] Ho T, Medard M, Koetter R, et al. A random linear network coding approach to multicast[J]. IEEE Transactions on Information Theory, 2006, 52(10): 4413-4430
- [13] Galois field arithmetic library[OL]. http://www.partow.net/projects/galois/index.html