

# 基于 Markov 链路径使用模型的软件统计测试

高海昌<sup>1</sup>, 冯博琴<sup>1</sup>, 曾 明<sup>2</sup>, 贺晓红<sup>1</sup>

(1. 西安交通大学电子与信息工程学院, 西安 710049; 2. 西安交通大学软件学院, 西安 710049)

**摘 要:** 探讨了建立基于路径使用的 Markov 链模型来分析模块内部代码结构的统计测试方法的可能性。提出将统计理论用于软件的结构测试, 从理论上得出软件的结构可靠性结论。提出具有有限状态、离散时间序列的 Markov 链作为程序的使用模型和测试模型, 使用 Kullback 判别式作为测试链收敛到使用链的判断准则, 从理论上证明了测试链到使用链收敛的必然性。理论分析和初步的实验证明该方法是可行和有前途的。

**关键词:** 软件统计测试; 结构测试; Markov 链; 路径使用模型; Kullback 判别式

## Statistical Software Test Based on Markov Chain Path Usage Model

GAO Haichang<sup>1</sup>, FENG Boqin<sup>1</sup>, ZENG Ming<sup>2</sup>, HE Xiaohong<sup>1</sup>

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049;

2. School of Software, Xi'an Jiaotong University, Xi'an 710049)

**【Abstract】** A Markov chain model based on path usage model is developed to analyze the inner code structure in statistical software test. The statistical theory is used on software structure test to obtain the reliability of the software structure theoretically. A Markov chain with finite state, scattered time sequence is put forward as the usage model and test model of the program. The Kullback discriminant is used as the convergence judgment criteria of the test-chain to usage-chain, and the inevitability of the convergence is testified theoretically. The primary experiments and theorized analysis prove the method is approving and promising.

**【Key words】** Software statistical test; Structure test; Markov chain; Path usage model; Kullback discriminant

统计测试是用统计学原理来进行软件测试的一种方法, 软件的统计测试建立起了关于软件系统期望质量的统计结论。Whittaker 于 1994 年首次将 Markov (马尔科夫) 链用于软件统计测试<sup>[1]</sup>, Poore 也提到操作应用模型只是统计测试模型的一种典型方式<sup>[2]</sup>。但是这两种基于操作使用模型的统计测试都从黑盒测试角度出发的, 输入域和目标函数从软件规范得出, 与程序结构无关。这就提出了一个问题, 是否可以将统计理论用于结构测试 (白盒测试), 从而从理论上得出软件的结构可靠性结论?

基于此, 本文提出了建立基于路径使用的 Markov 链模型, 来分析模块内部代码结构的统计测试方法。提出具有有限状态、离散时间序列的 Markov 链作为程序的使用模型和测试模型, 引入 Kullback 判别式, 评估使用链 U 和测试链 T 的相似程度, 分析了从测试链 T 到使用链 U 的收敛, 并以此为基础进行测试停止判断。

### 1 Markov 链统计测试策略

#### 1.1 统计测试理论和过程

当一个总体太庞大时, 难以进行彻底的研究。而对一个大型程序, 测试路径模型需要建立模块的所有可能执行路径, 也是一个庞大的天文数字。此时在统计上的正确采样可以看作是对测试软件总体进行评估的一个基础。从图 1 可以看出传统的统计设计和软件统计测试之间的并行性<sup>[2]</sup>。在统计协议下, 可以对模块结构进行建模。

基于模块路径的统计测试遵循自底向上的测试集成策略。这种统计测试是从最底层模块开始来进行构造和测试的,

在进行时要求所有从属于某个给定层次的模块总是存在的, 而且也不再有用桩模块的必要<sup>[3]</sup>。统计测试涉及到的步骤如图 2 所示。

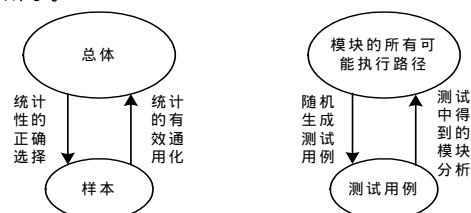


图1 统计方法与软件测试间的并行性

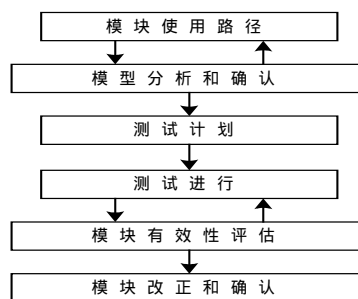


图2 统计测试过程

**基金项目:** 国家“863”计划基金资助项目(2003AA1Z2610)

**作者简介:** 高海昌(1978 - ), 男, 博士生, 主研方向: 软件工程, 智能优化算法; 冯博琴, 教授、博导; 曾 明, 教授; 贺晓红, 硕士生

**收稿日期:** 2005-12-28

**E-mail:** gaohaich@gmail.com

## 1.2 路径使用 Markov 链和测试 Markov 链

在物理学中，很多确定性现象遵从如下演变原则<sup>[3]</sup>：由时刻  $t_0$  系统或过程所处的状态，可以决定系统或过程在时刻  $t > t_0$  所处的状态，而无需借助  $t_0$  以前系统或过程所处状态的历史资料。把上述原则延伸到随机现象，可引入 Markov 性：

**定义 1** 过程（或系统）在时刻  $t_0$  所处的状态为已知的条件下，过程在时刻  $t > t_0$  所处状态的条件分布与过程在时刻  $t_0$  之前所处的状态无关，就说这个过程具有 Markov 性。

可以用分布函数来表述 Markov 性。设随机过程  $|X(t), t \in T|$  的状态空间为  $I$ 。如果对时间  $t$  的任意  $n$  个数值  $t_1 < t_2 < \dots < t_n (n \geq 3, t_i \in T)$ ，在条件  $X(t_i) = x_i (x_i \in I, i = 1, 2, \dots, n-1)$  下， $X(t_n)$  的条件分布函数恰等于在条件  $X(t_{n-1})$  下等于  $X_{n-1}$  的条件分布函数，即

$$\begin{aligned} P\{X(t_n) \leq x_n | X(t_1) = x_1, X(t_2) = x_2, \dots, X(t_{n-1}) = x_{n-1}\} \\ = P\{X(t_n) \leq x_n | X(t_{n-1}) = x_{n-1}\}, x_n \in R \end{aligned} \quad (1)$$

则称过程  $\{X(t), t \in T\}$  具有 Markov 性，或称此过程为 Markov 过程。

**定义 2** 时间和状态都是离散的 Markov 过程称为 Markov 链，简称马氏链，记为  $\{X_n = X(n), n = 0, 1, 2, \dots\}$ 。它可以看作在时间集  $T_1 = \{0, 1, 2, \dots\}$  上对离散状态的过程相继观察的结果。

假定链的状态空间  $I = \{a_1, a_2, \dots\}, a_i \in R$ 。Markov 性通常用条件分布律来表示，即对任意的正整数  $n, r$  和  $0 \leq t_1 < t_2 < \dots < t_r < m, (t_i, m, n+m \in T_1)$  有：

$$\begin{aligned} P\{X_{m+n} = a_j | X_{t_1} = a_{i_1}, X_{t_2} = a_{i_2}, \dots, X_{t_r} = a_{i_r}, X_m = a_i\} \\ = P\{X_{m+n} = a_j | X_m = a_i\} \end{aligned} \quad (2)$$

其中， $a_i \in I$  称条件概率

$$P_{ij}(m, m+n) = P\{X_{m+n} = a_j | X_m = a_i\} \quad (3)$$

为马氏链在时刻  $m$  处于状态  $a_i$  条件下，在时刻  $m+n$  转移到状态  $a_j$  的转移概率。

由于链在时刻  $m$  从任何一个状态  $a_i$  出发，到另一时刻  $m+n$ ，必然转移到  $a_1, a_2, \dots$  诸状态中的某一个，因此：

$$\sum_{j=1}^{\infty} P_{ij}(m, m+n) = 1, i = 1, 2, \dots \quad (4)$$

由转移概率组成的矩阵称为马氏链的转移概率矩阵。由式(4)知，此矩阵的每一行元素之和等于 1。可见，Markov 链是一个状态序列，它的关键特性是状态  $i$  出现的概率依赖且仅依赖其前一个状态。

**定义 3** 对路径使用模型来说，Markov 链的状态空间是模块的控制流图的结点。将该模型称为路径使用 Markov 链。

**定义 4** 对测试模型来说，Markov 链的状态空间与路径使用模型相同，这个模型称为测试 Markov 链。

## 2 基于模块路径使用的状态空间

### 2.1 路径使用 Markov 链

使用模型以类状态转换图的形式表示，这里结点代表软件使用时该模块经过的路径结点，过渡弧表示状态间的可能执行的概率。若图存在任何循环或回路，则遍历模型时形成有限状态的无穷路径。

建立路径使用 Markov 链的基本任务就是识别模块的各分支结点和这些结点之间的联系。路径使用链从被测模块的控制流图导出，同时在建立路径使用链时，根据各分支的估

计概率标上该路径执行的概率。控制流图确定后，使用链的状态也得以确定。为了便于记录，增加开始和结束状态。确定使用链后，可以计算各状态转换的概率分布。

### 2.2 路径测试 Markov 链

当路径使用模型完成后，可以利用该模块的测试用例进行测试。提供测试用例需要同时提供当前测试用例可能经过的路径，以便于实际执行时的路径进行比较计算。这一测试过程在自主开发的自动测试工具 WBoxTool 中进行了实现，在测试过程中通过插装记录每个测试用例的通过路径。

当一组测试用例执行完毕后，比较实际执行路径与给定预期的执行路径，若二者不同则认为该模块发生错误，标识为一个软件失效，此时可进行跟踪测试以改正错误。当  $n$  组测试用例执行完毕，进行路径测试 Markov 链的状态转移概率的计算。概率计算方式为从路径测试 Markov 链的开始状态出发，计算每一条过渡弧的通过次数占总测试用例个数的百分比。初始计算时，状态图上的路径使用 Markov 链的概率分布差别较大。经过  $m$  个  $n$  组的测试用例执行，测试模型的概率分布逐渐趋近于使用模型，当二者的差值小于用户可接受的域值时，即可停止测试。

表 1 中的 Markov 链的一些标准分析结果可以用于软件测试过程中的数据分析。将这一测试模型看作是马氏链的转移概率矩阵  $U$ ，其状态间过渡的概率分布即为测试模型中的结点之间的路径经过的概率； $\cdot$  表示除状态  $i$  和  $j$  以外的其它状态集合。

表 1 Markov 链的一些标准分析结果

公式计算结果 (从状态 $i$ 出发)	概率分析公式	公式含义
一组测试用例（一次运行）经过状态 $j$ 的概率	$y_{ij} = U_{ij} + \sum_{k \in r} U_{ik} \cdot y_{kj}$	输入一组测试用例，状态 $j$ 可能出现的概率
测试必然经过状态 $j$ 时所需测试用例个数	$h_j = 1 / y_{ij}$	状态 $j$ 至少经过一次所需执行的测试用例个数
一组测试用例（一次运行）经过弧 $j,k$ 的概率	$z_{jk} = y_{ij} U_{jk}$	输入一组测试用例，弧 $j,k$ 可能出现的概率
测试必然经过弧 $j,k$ 时所需测试用例个数	$h_{jk} = 1 / z_{jk}$	弧 $j,k$ 至少经过一次时所需执行的测试用例个数

路径使用 Markov 链  $U$  在测试中具有固定的概率。但是，测试 Markov 链  $T$  中的概率则是不停进行更新的，同时追踪  $T$  的进展也是监视测试过程的一部分。令  $s_1, s_2, \dots, s_m$  代表逐次输入的测试用例，并输入程序  $P$  执行。相应的测试链序列为  $T_1, T_2, \dots, T_m$ ，表示了  $T$  在测试中的进展。无论何时需要  $T_i$  的状态过渡概率的计算，都可以将这些弧出现的频率次数转换为相对频率概率。

### 3 测试停止标准——测试链到使用链的收敛

软件统计测试的停止标准可以简单地描述为选择一个目标可靠性，然后进行测试，直到可靠性估计达到或超过目标可靠性时停止测试。

测试链到使用链的逼近给出了直接基于使用链和测试链的统计属性的一个停止标准。使用链是模块的理想测试的模型，每一条弧的概率都建立在实际使用的最佳估计，并且没

有失效情况出现。而测试链是特定测试历史的模型，包括失效情况发生。因此，使用链表示了没有失效情况下的统计测试情况，而测试链代表了有失效情况下的统计测试。两个模型之间的差异成为测试过程的有用衡量。

### 3.1 Kullback 判别准则

**定义 5** Kullback 判别式是两个随机过程的对数似然率的期望值<sup>[4]</sup>，即：

$$K(U, T) = \lim_{n \rightarrow \infty} \frac{1}{n} \left[ \log \left( \frac{\Pr[X_0, X_1, \dots, X_n | U]}{\Pr[X_0, X_1, \dots, X_n | T]} \right) \right] \quad (5)$$

这里， $\Pr[X_0, X_1, \dots, X_n | \lambda]$  表示在随机过程 概率分布条件执行  $n$  组测试用例的情况。尽管  $K(U, T)$  不能从随机过程  $U$  和  $T$  中直接计算，但它可以按照 Markov 链  $U$  和  $T$  计算如下：

$$K(U, T) = \sum_{ij} \pi_i * u_{ij} \log \frac{u_{ij}}{t_{ij}} \quad (6)$$

这里， $\pi_i$  是  $U$  的固定分布， $u_{ij}$  是  $U$  中从状态  $i$  到  $j$  的过渡的概率， $t_{ij}$  是  $T$  中的相应概率。 $K(U, T)$  非负，并且当且仅当对所有的  $i, j$  都有  $u_{ij} = t_{ij}$  时  $K(U, T)$  为 0。

### 3.2 测试链到使用链的收敛

如果  $K(U, T)$  很小的话，模型  $U$  和  $T$  所形成的路径概率分布几乎是相同的。 $K(U, T)$  的值趋近于 0 说明以下几点：(1) 它保证了每个路径使用状态在测试历史中以正确的概率出现，正如在  $U$  的固定分布中计算的概率，测试历史的发展变化趋势近似匹配  $U$  的这些概率；(2) 通过不断的修正程序，它强迫  $T$  中的失效出现趋向 0。这意味着在  $K(U, T)$  小到可以接受的程度时，通过测试链的每条路径的可信度即可获得。(3) 它为测试这一软件规定了使用链的限制。当测试链的统计和使用链匹配时，使用链本身也不能在统计测试过程中增加额外信息。

**定义 6** 当  $U$  和  $T$  二者的差别落到预定义的下限，即  $K(U, T)$  小于某一预定义值，并且随着测试的进展，测试经历变化很小时，额外的测试用例不再会显著影响测试模型的统计，此时测试终止。

近似相等概率和 Kullback 判别式可以联合使用。测试链收敛时，Kullback 判别式用于评估当前测试经历匹配软件预期使用的程度，而近似相等概率用于评估测试记录将在未来显著变化的机会<sup>[4,5]</sup>。

**定理 1** 路径测试 Markov 链  $T$  到路径使用 Markov 链  $U$  的收敛是确定的。

**证明** 当对程序  $P$  进行失效修正后，测试链中弧的频率需要重新设置。测试链模型可能根据计算得到的概率发生变化，如在使用链中存在的某条弧，若在测试链中计算得到概率为 0 的弧，那么将它去掉，从直观意义上来说使用链上有一条路径不可达。反之，若使用链中不存在的弧（即概率为 0 的弧）出现在测试链中，意味着有程序中不允许出现的情况发生。如果没有检测到错误， $T$  的最终概率分布会趋向于  $U$ 。只要  $m$  足够大，Markov 链的大数定律保证了这些测试用例  $s_1, s_2, \dots, s_m$  将成为  $U$  的代表性的统计模型。因为  $T$  的弧上的相对频率必然收敛到  $U$  的弧上的概率，所以这意味着  $T$  到  $U$  的收敛是确定性的，定理 1 得证。

在实际测试过程中，往往采用多停止标准共用的准则：(1) 可靠性估计的可信度；(2) 测试经历已经覆盖软件预期使用的程度；(3) 测试中状态、弧、路径覆盖程度的模块覆盖准则等。这样做的好处是可以综合评价软件的可靠性和成熟度。

## 4 实例验证

### 4.1 路径使用 Markov 链

图 3 的程序路径使用 Markov 链，状态空间为 {1, 2, 3, 4, 5, 6}，状态转移概率如图所示。当随机输入测试数据后，有 65% 的概率进入循环，而只有 35% 的概率不执行循环，或从循环退出。对于从状态 2 到状态 5 之间的分支选择，采用出射弧的均匀分布方法。

### 4.2 状态和弧的覆盖百分比

表 2 给出了一个关于状态和弧的覆盖百分比的实例。测试中，通常要关注在测试中产生的软件的一些特定属性的覆盖率。而黑盒统计测试的缺点就是在给定测试用例时，软件代码执行的覆盖率很难得知。可以利用测试过程中获得的数据用来评估测试用例对应用模型状态和弧的覆盖率。

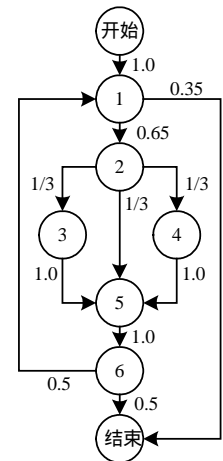


图 3 某程序的路径使用 Markov 链

表 2 关于状态和弧的覆盖百分比的实例

输入测试用例的个数	状态的覆盖率估计	路径弧的覆盖率估计
1	18.75	86.32
2	62.50	87.89
5	75.00	91.79
10	87.50	94.92
15	100.00	95.31
35	100.00	98.82
36	100.00	100.00

## 5 总结

本文将统计学理论用于软件的白盒测试，从模块的内部结构出发构造基于路径的 Markov 链使用模型，并利用统计结论进行分析软件的实际测试情况。基于路径使用模型的软件统计测试可以使得统计学的许多优点应用于软件测试：(1) 它提供了一种形式化的方法，将测试过程表示为一个统计课题；(2) 它有助于追踪测试过程，并记录测试结果；(3) 它提供了可以使用统计方法进行测试停止的判断一个基础。这种统计测试方法的应用，同时也方便了测试计划的制定，这也反过来使得它可能在系统开发的每一步中优化测试质量。相信随着研究的不断深入，基于路径使用模型的软件统计测试方法一定会得到广泛的应用。

### 参考文献

- Whittaker J A, Thomason M G. A Markov Chain Model for Statistical Software Testing[J]. IEEE Transactions on Software Engineering, 1994, 20(10): 812- 824.
- Poore J H, Trammell C J. Application of Statistical Science to Testing and Evaluating Software Intensive Systems, in Statistics, Testing, and Defense Acquisition[M]. Washington, D.C: National Academy Press, 1998: 40-57.
- 盛 骤, 谢式千. 概率论与数理统计[M]. 北京: 高等教育出版社, 1989: 31-32.
- Kullback S. Information Theory and Statistics[M]. New York: Wiley, 1958: 10-83.
- Sayre K, Poore J H. Stopping Criteria for Statistical Testing[J]. Information and Software Technology, 2000, 12(42): 851-857.