基于概率统计的软件测试方法研究

周 珺
（兰州交通大学数理学院　兰州甘肃，730070）

摘　要：为了缩短软件测试周期，本文把马尔可夫链模型运用于软件可靠性测试中，提出了这一技术进行软件可靠性测试的方法。在测试过程中使用了新的评判准则分析测试结果，通过实例证明了该评判准则的实用性和有效性。

关键词：软件可靠性；软件测试方法；测试用例；马尔可夫链；马尔可夫链模型

Abstract: In this paper,in order to shorten the period of software testing ,Markov Chain Model is used in software reliability testing .It is introduced the basic method of this technology in testing software reliability. In the testing process ,it is used that new judge standard to analyze the testing result. The judge standard are the practicability, efficiency through examples.

Key words: Software reliability ; Software testing technology ; Test case ; Markov chain ; Markov chain model

中图分类号：TP311　　**文献标识码：**A　　**文章编号：**1001-9227(2011)04-0001-03

0　引 言

软件测试是软件工程领域必不可少的过程，在软件生存周期中占有非常重要的位置。据统计，软件开发总成本中，用在测试上的开销要占30%到50%，特殊情况下，对可靠性要求很高的软件，其测试费用甚至高达所有其他软件工程阶段费用总和的3～5倍[1]。在对传统测试用例的生成方法学习的基础上，探索出了一种实用性强的方法，该方法在马尔可夫链模型的基础上，提出了一种软件可靠性测试模型，并通过实例证明了这种模型的实用性和正确性。

1　马尔可夫链的基本概念[2-4]

1.1　状态和状态转换

基于马尔可夫链的软件使用模型是由软件的状态和边组成。状态表示软件在使用过程中的内部环境。状态转换是指当软件在某一状态经输入激励，从该状态转换到另一个状态。

马尔可夫链是满足下面两个假设的一种随机过程：

t+1时刻系统状态的概率分布只与t时刻的状态有关，与t时刻以前的状态无关；从t时刻到t+1时刻的状态转移与t的值无关。一个马尔可夫链模型可表示为M=(S，P，Q)，其中各元的含义如下：

(1)S是系统所有可能的状态所组成的非空的状态集，有时也称之为系统的状态空间，它可以是有限的、可列的集合或任意非空集。本文中假定S是可数集(即有限或可列)。用小写字母i，j(或S_i，S_j)等来表示状态。

(2) $p=\left[p_{ij}\right]_{n\times n}$ 是系统的状态转移概率矩阵，其中p_{ij}表示系统在时刻t处于状态i，在下一时刻t+1处于状态j的概率，N是系统所有可能的状态的个数。对于任意i∈S，有$\sum_{j=1}^{N}p_{ij}=1$。

(3)Q=[q1，q2，…，qn]是系统的初始概率分布，q_i是系统在初始时刻处于状态i的概率，满足$\sum_{j=1}^{N}q_{ij}=1$。

2.2　输入激励与状态转换概率

软件处于某一稳定的内部状态，外界环境有相应的输入激励，激励可以是不同的输入变量或者相同的输入变量取不同的值。不同的输入激励将导致软件不同的状态转换。当软件在稳定的使用环境下，不同的输入激励的出现是遵循一定的统计分布的，因而导致软件状态转换间也存在相应的概率分布，这个概率就称为状态转换概率。如果遵循软件的状态转换概率分布抽样产生测试输入序列，则体现了统计意义上的软件使用方式。

2.3　软件的使用链

软件的使用链是用马尔可夫链描述的软件载誉期使用环境中的状态转换模型，用U表示。定义为：U={S，ARC}，其中S代表软件的状态集，有S={s_1，s_2，…，s_n}；而ARC表示软件状态之间转换关系，有

$$ARC=\{arc_{11},arc_{12},\cdots arc_{1n},arc_{21},arc_{22},\cdots,arc_{2n},\cdots,arc_{n1},arc_{n2},\cdots arc_{nn}\} \quad (1)$$

而其中的每个状态转换关系又是一个二维向量。用D表示引起软件状态转换的输入激励域，p表示软件状态转换率，则有

$$arc_{ij}=(d_{ij},p_{ij})\ (d_{ij}\in D;i=1,2\cdots,n;j=1,2,\cdots,n) \quad (2)$$

U={S，ARC}中，S_i表示时刻t，ARC表示测试用例集D发生的概率P，应用2中的结果分析准则，要使使用链的状态转换概率为1，必须满足2中的①、②，即使用链在时刻t的测试用例集D应满足完全覆盖。

3　基于马尔可夫链的软件测试方法

3.1　结果分析准则

收稿日期：2011-02-02
作者简介：周珺(1963-)，男，甘肃靖远人，副教授，主要研究方向为数理统计，软件测试。

马尔可夫链使用模型(Markov Chain Usage Model)也是一种很好的软件使用方式的方法[4-5]。在软件测试过程中，对于被测软件，基于逻辑覆盖和基本路径测试方法生成的测试用例，覆盖了程序的所有测试，保证了在测试中程序的每个可执行语句至少执行一次[6]，由此可知：这些测试用例构成了一个完整的测试实验样本空间；生成的每个测试用例一定发生并且发生的概率相同。因此，可以把测试用例的生成及其发生概率和2中的马尔可夫链模型联系起来。把一个完全正确的逻辑结构或模块对应的状态集看成在时刻t处于的状态集，则状态集中的每一个状态对应一个测试用例，从而构成一个完整的满足马尔可夫链模型的测试用例集；同样，把接下来要测试的逻辑结构或模块所处的状态看作下一时刻t+1处于的状态。通过以上对模型的分析，可以初步构造一个基于该模型的软件可靠性测试的结果分析准则，描述如下：

（1）s是被测逻辑结构或模块所对应的测试用例集，它应该是有限的、可列的集合或任意非空集。

(2)Q=[q_1, q_2, …, q_n]是时刻t测试用例的发生概率分布，q_i是在t时刻测试用例i的发生概率，满足$\sum_{j=1}^{N} p_{ij}=1$。其中，$q_1=q_2=\cdots=q_n$，即在t时刻，测试用例集中的每个测试用例发生概率相同。

结果分析准则的概率表示形式为Pz=(D，H)。

（1）D是被测逻辑结构或模块所对应的测试用例集，它应该是有限的、可列的集合或任意非空集。

(2)H是时刻t测试用例的发生概率分布。其中H=[h_1、$h_2\cdots h_k$]，P({h_i})表示t时刻第i个测试用例发生的概率，满足：

$$P(\{h_1\})=P(\{h_2\})=\cdots=P(\{h_k\})=1/k \quad (1)$$

$$P(\{h_1\})+P(\{h_2\})+\cdots+P(\{h_k\})=1 \quad (2)$$

即在t时刻，测试用例集中的每个测试用例全都发生，并且发生概率相同。

3.2 测试方法

在基于马尔可夫使用模型的软件统计测试过程中，首先要根据软件需求规范建立软件的使用链；然后根据使用链进行序列抽样，产生测试用例；利用测试用例集驱动程序运行，获得输出数据后，利用评判准则进行评判：如果所有测试用例都发生，并且测试数据与预期数据相符，则说明测试成功，可进行下一时刻的测试。在进行每一时刻的测试中，都应满足评判准则。

在进行测试过程中，如果测试结果显示$\sum_{j=1}^{N} q_{ij}\neq 1$，说明被测软件逻辑结构不满足完全覆盖性，测试无需再进行。否则，再进行第二步判定，如果测试结果数据与预期数据不相符，说明被测软件的逻辑结构或结构中的语句本身不正确，则对其重新进行检查。

产生测试用例时，从初态开始，在每一个状态都生成一个0～1之间的随机数，根据这个随机数选择这个状态的一条出边，转移到下一个状态，周而复始，直到终态，于是，测试用例便生成了。由于这样产生的测试用例是严格遵循软件的使用链并按照状态转移概率随机生成的，所以能够很好地体现软件的真实使用方式。

当然，在软件可靠性增长测试过程中，利用基于马尔可夫使用模型的统计测试所产生的失效数据作为软件可靠性增长模型的输入，可以获得相应的可靠性估计或预计。在软件可靠性验证测试过程中，直接利用马尔可夫使用模型产生一定量的测试用例，通过对失效数据的观察，便可以获得对软件的可靠性验证。所以这里只关注如何建立相应的马尔可夫软件使用模型，以及相应的测试用例的抽取方法。

4　一个简单的实例分析

运用以上提出的测试方法，对某一软件在进行了测试后得出的有关各类错误数据资料整理分析得出了错误分类统计表[7]，如表1所示。

表1　错误分类统计表

错误分类	错误数	百分比
需求错误	1317	8.1
功能和性能错误	2624	16.2
结构错误	4082	25.2
数据错误	3638	22.4
实现与编码错误	1601	9.9
软件集成错误	1455	9.0
系统结构错误	282	1.7
测试定义与测试集成错误	447	2.8
其他类型错误	763	4.7
总计错误	16209	100.0

根据表1的资料可设α_1表示需求错误，α_2表功能和性能错误，α_3表结构错误，α_4表示数据错误，α_5表实现与编码错误，α_6表软件集成错误，α_7表系统结构错误，α_8表测试执行错误，α_9表其它类型错误。由此得出“性质错误”类$\Omega_1=\{\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,\alpha_7,\alpha_8,\alpha_9\}$，其总体分布为$P(\alpha_1)=p_1$，$P(\alpha_2)=p_2$，$P(\alpha_3)=p_3$，$P(\alpha_4)=p_4$，$P(\alpha_5)=p_5$，$P(\alpha_6)=p_6$，$P(\alpha_7)=p_7$，$P(\alpha_8)=p_8$，$P(\alpha_9)=p_9$。由表1得出“性质错误”样本，其样本Ω1’总体容量为16209。由大数定理可以得$p_1\approx0.081$，$p_2\approx0.162$，$p_3\approx0.252$，$p_4\approx0.224$，$p_5\approx0.099$，$p_6\approx0.09$，$p_7\approx0.017$，$p_8\approx0.028$，$p_9\approx0.047$。

“性质错误”类Ω_1包括9类错误，总计错误的百分比100%，得出：这9类错误构成了完整的测试实验样本空间Ω_1。由于每类错误中的测试用例数目不同，所以其错误数的百分比不同，根据测试用例的概念[5]可知：所有类的各个测试用例是相互独立的，并且每个测试用例都得发生，即每个测试用例发生的概率是相同的。所以这个例子满足了提出的Pz=(S，Q)这个式子。

在根据测试模型产生测试用例时，要给出预期的测试结果。如果测试数据与预期数据相符，则测试通过。否则再根据3.2中测试策略进行查错分析。

5　结束语

本文基于马尔可夫链模型，提出一种软件可靠性测试模型：根据软件需求规范建立软件的使用链；然后根据

（下转第5页）

式(7)，得到PID控制器的三个参数P、I、D。

(5)计算PID控制器的输出$u(k)$，参与控制和计算；

(6)根据式(11)、(12)、(13)分别计算修正输出权值、节点基宽参数，节点中心值；

(7)取$k=k+1$，返回步骤(2)。

3 仿真实例

为了验证RBF网络对PID整定的控制效果，以加热炉的物料出口温度-炉膛温度串级系统的控制为例，对应的控制原理方框图如图4所示，选取主、副对象的传递函数[10]分别为

$$G_{1(S)}=\frac{18}{70S^2+19S+1}e^{-12S}\quad G_{2(S)}=\frac{6}{15S+1}e^{-12S}$$

取采样周期T=3s，将以上两个传递函数离散化，应用Matlab中的Simulink进行仿真，同时与传统PID串级控制方法进行比较。输入信号取单位阶跃，RBF网络结构选取3-4-1，学习速率η_i、η_p、η_i、η_d分别取0.4，动量因子α=0.05。传统PID控制的参数取P=0.15，I=0.3，D=0.08，RBF神经网络和传统PID控制输出仿真图如图3所示。

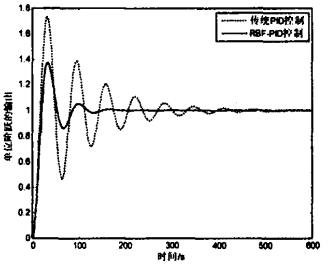

图3 两种控制对加热炉温度系统的单位阶跃输出曲线

从图3可以看出，应用RBF神经网络对PID控制参数进行自适应调整，系统阶跃响应曲线在历经170秒，系统输出处于稳定状态。而用传统PID控制，在近500秒的时候，系统的输出才趋于稳定状态。综上所述，引入RBF神经网络的PID控制系统比传统PID控制系统有较小的振荡幅度和更短的振荡时间。在实际应用中，采用RBF神经网络的PID控制可以使锅炉温度值经过控制参数的训练调整，更迅速更稳定地达到设定值。

4 结 论

RBF神经网络算法具有良好的全局搜索寻优特性，可以任意精度逼近非线性函数，且操作方便，速度快，尤其是不需要建立精确的数学模型，因而适用于大多数工业过程控制中。本文采用RBF神经网络算法在线优化PID控制参数效果很明显，大大地改善了传统PID控制器的性能，有效地提高了系统的响应速度，且减小了振荡幅度。仿真结果表明，基于RBF的PID控制较传统PID控制有较高的适应能力。

参考文献

[1] 刘金琨. 先进PID控制Matlab仿真(第2版)[M]. 北京:电子工业出版社, 2004.

[2] 赵瑞军, 王先来. 模糊-PID控制器在空调温度控制中的应用[J]. 计算机仿真, 2006, 23(11):311-313.

[3] 李 文等. 智能控制及其应用综述[J]. 重庆邮电学院学报, 2006, 18(3):376-381.

[4] 刘寅虎, 李绍铭. 基于动态RBF神经网络在线辨识的单神经元PID控制[J]. 系统仿真学报, 2008, 20(3):627-630.

[5] 焦竹青, 屈百达, 徐保国. 基于RBF神经网络的多变量系统PID解耦控制[J]. 系统仿真学报, 2002, 23(6):530-533.

[6] 武国庆等. 基于径向基神经网络的不确定非线性系统的鲁棒自适应控制[J]. 航空学报, 2002, 23(6):530-533.

[7] 张 雅等. RBF网络模型参考自适应控制在温度控制中的仿真研究[J]. 系统仿真学报, 2008, 20(2):429-432.

[8] R Somakumar, J Chandrasekhar. Intelligent anti-skid brake controller using a neural network[J]. Control Engineering Practice, 1999(7):611-621.

[9] 俞金寿. 工业过程先进控制技术[M]. 上海:华东理工大学出版社, 2008.

[10] 蒋宗礼. 人工神经网络导论[M]. 北京:高等教育出版社, 2001.

[11] 陈书谦, 张丽虹. BP神经网络在PID控制器参数整定中的应用[J]. 计算机仿真, 2010, 27(10):171-174.

(上接第2页)

使用链进行序列抽样，产生测试用例进行测试，利用提出的评判准则对测试结果进行分析。通过实例证明，这种方法可以降低测试的复杂度，解决了简化测试难度的问题。

参考文献

[1] 杨季文等. 80x86汇编语言程序设计教程[M]. 北京:清华大学出版社, 2004, 180-181.

[2] 赵艳丽, 孙志挥. 基于静态马尔可夫链模型的实时异常检测[J]. 计算机应用, 2004, 24(11):30-32.

[3] YE N. A Markov chain model of temporal behavior for anomal detec—tion[A]. 2005 IEEE Sytem, Man, and Cybernetics Information As—surance and Security Workshop [C]. West Point, NY, 2006.

[4] JHA S, TAN K, MAXION R. Markov chains, classifiers, and intrusion detection[A]. Computer Security Foundation Workshop, the 14th IEEE[C]. Cape Breton, Novia Scotia, Canada, 2001.

[5] ju WH, VARDI Y. A HybridHigh—order Markov Chain Model for Computer Intrusion Detection[R]. Technical Report 92, National Institute for Statistical Sciences, Research Triangle Park, North Caro-lina 27709-4006, 2007

[6] 朱少民. 软件测试方法和技术[M]. 北京:清华大学出版社, 2005, 283-289.

[7] 李兴南, 葛 玮, 董云卫等. 一种基于大数定律的软件测试方法[J], 微机发展, 2005, 15(2):15-17.