

软件可靠性预测的核函数方法

楼俊钢 申 情 沈张果

(湖州师范学院信息与工程学院 湖州 313000)

摘 要 通过合理的假设和抽象,把软件可靠性预测问题转化成从高维空间向低维空间的非线性映射,采用核函数回归估计理论对软件失效时间数据之间的对应关系进行建模,在新建立的模型中,随着软件失效不断发生,模型参数将不断自动调整以适应失效过程的动态变化,从而实现软件可靠性的自适应预测,并对模型中核函数以及回归估计算法的选择进行了研究。最后,对 14 个软件失效数据集进行了实验分析,并对新建模型与部分其它模型的预测能力和适用能力进行了比较。

关键词 核函数,核函数回归算法,软件可靠性模型,非齐次泊松过程模型

中图分类号 TP302.7 **文献标识码** A

Kernel Methods of Software Reliability Prediction

LOU Jun-gang SHEN Qing SHEN Zhang-guo

(Department of Computer Science and Technology, Huzhou Teachers College, Huzhou 313000, China)

Abstract The prediction of future failure data from observed data sets can be transformed into a problem of nonlinear regression, and the kernel functions method is very efficient for solving nonlinear regression problems. A kernel functions-based generic model adaptive to the characteristic of the given data sets is used for software failure time prediction, and it is applied to learn and recognize the inherent internal temporal property of software failure sequence in order to capture the most current feature hidden inside the software failure behavior. The experimental results based on fourteen real data sets show that the proposed model has better prediction and applicability than that of some other conditional software reliability prediction models.

Keywords Kernel functions, Kernel regression, Software reliability model, Nonhomogeneous poisson process

软件的失效行为带有不确定性,对其进行精确刻画异常困难,研究人员多采用随机性^[1]、混沌性^[2]、模糊性^[3]、未确知理论^[3]等来描述软件失效的不确定性。目前对随机过程类软件可靠性模型的研究绝大部分集中于模型的统一、对已有模型的改进、可靠性成本模型等^[4]。模型改进的出发点是提出更加合理的假设以提高模型预测精度,如考虑测试环境与实际运行环境的差别、故障的相关性、不完美调试以及测试者学习能力、测试效用函数以及考虑软件不同失效过程偏差^[5]等。然而随机过程类软件可靠性模型的适用性较差,不能满足各类型软件的失效过程描述^[4]。人工神经网络以其强大的并行处理机制、任意函数的逼近能力和学习能力以及自组织和自适应能力等在软件可靠性建模领域得到了较好的应用。Karunanith 等人最早把人工神经网络理论应用到软件可靠性评估和预测中^[6],神经网络同时被用来估计模型的参数和仿效模型本身预测未来结果。神经网络模型受网络结构复杂性和样本复杂性的影响较大,容易出现“过学习”或低泛化能力,特别是神经网络学习算法缺乏定量的分析与完备的理论基础支持。根据已知软件失效时间数据预测未知失效数据可以看作是从高维空间向低维空间的非线性映射, Tian 等人率先把 SVM 引入到软件可靠性建模中^[8],此后 Yang 等人^[9]对 SVM 用于可靠性建模进行了适当的改进,本文作者及合作者使用

相关向量机在软件可靠性建模方面也做了一些工作^[10],然而这些研究才起步不久,所得结果也各有优劣,加之这是一个多学科高度交叉的领域,还有大量的工作需要进一步探索。

本文采用核函数回归估计理论,对软件失效时刻与在它之前的 m 个失效时刻数据进行学习,从而捕捉失效时刻之间的内在依赖关系,由此构建基于核函数回归估计的软件可靠性预测模型。这一思想跳出了传统软件可靠性建模过程中关于失效强度变化的各种统计分布的假设,因此该模型具有较高的预计精度和较好的适用性。在新建立的模型中,随着软件失效不断发生,模型参数将不断自动调整以适应失效过程的动态变化,从而实现软件可靠性的自适应预测。通过核函数技术的应用,把软件可靠性预测问题转化为一个回归估计问题,并应用了核主成分回归(Kernel Principal Components Regression, KPCR)^[11]、核偏最小二乘回归算法(Kernel Partial Least Squares, KPLS)^[12]、支持向量回归算法(Support Vector Regression, SVR)^[7]、相关向量回归算法(Relevance Vector Regression, RVR)^[13,14]等多类算法来解决这一问题。在多组公开发表的数据集上进行了实验比较与分析,发现新模型较之一些广泛使用的 NHPP 模型在预测能力及适用性上有较大的提高,并发现了部分适合于可靠性预测的核函数结构以及回归估计算法。

到稿日期:2011-10-22 返修日期:2012-02-24 本文受国家自然科学基金(61103051)以及浙江省自然科学基金(Y1101237)资助。

楼俊钢(1982-),男,博士,讲师,CCF 会员,主要研究领域为软件可靠性建模、可信计算、系统性能评测等。

1 软件可靠性预测转化为核函数回归估计

假设已发生的软件失效时间为 t_1, t_2, \dots, t_n , 软件可靠性预测问题可以描述为: 从已知失效时间间隔数据序列 $\{t_i\}_{i=1}^n$ 预测未知的软件失效时间间隔数据 t_{i+d} , 使用核函数对软件失效时间与在它之前的 m 个失效时间数据建模, 以此捕捉失效时间内在的依赖关系, 一般地, 令 $t_i = f(t_{i-m}, t_{i-m+1}, \dots, t_{i-1})$, 则 t_i 服从固定但未知的条件分布函数 $F(t_i | t_{i-m}, t_{i-m+1}, \dots, t_{i-1})$, 使用核函数回归估计算法对软件失效时间数据进行学习, 可以捕捉失效时间内在的依赖关系。假设用于学习的失效时间序列为 $t_1, t_2, \dots, t_k (k > m)$, 则在 t_1, t_2, \dots, t_k 已知条件下对 t_{k+1} 进行预测变为: 已知 $k-m$ 个观测 $(T_1, t_{m+1}), (T_2, t_{m+2}), \dots, (T_{k-m}, t_k)$ 和第 $k-m+1$ 个输入 T_{k-m+1} 的情况下, 估计第 $k-m+1$ 个输出值 t_{k+1} , 其中 T_i 表示 m 维向量 $[t_i, t_{i+1}, \dots, t_{m+i}]$ 。把 $[t_2, t_{i+1}, \dots, t_{k+1}]$ 作为输入, 则可以预测 t_{k+2} , 同理可以预测得到 $t_{k+3}, \dots, t_{k+d}, \dots$ 。

基于核函数回归估计理论的软件可靠性动态模型的学习及预测过程如图 1 所示, 首先对观测序列 $(T_1, t_{m+1}), (T_2, t_{m+2}), \dots, (T_{k-m}, t_k)$ 进行学习, 并使用回归估计算法确定模型的参数值, 随着失效数据不断增加, 核函数参数不断动态调整, 学习结束后, 便可对 t_{k+1} 进行预测。对可以获得的部分系统输出数据即失效数据进行分析, 将其蕴涵的系统失效特征用核函数进行学习并表达出来, 构造软件故障过程的等价系统, 从而完成对软件系统失效行为的刻画, 并依据所建模型完成对系统未来失效行为的预测。

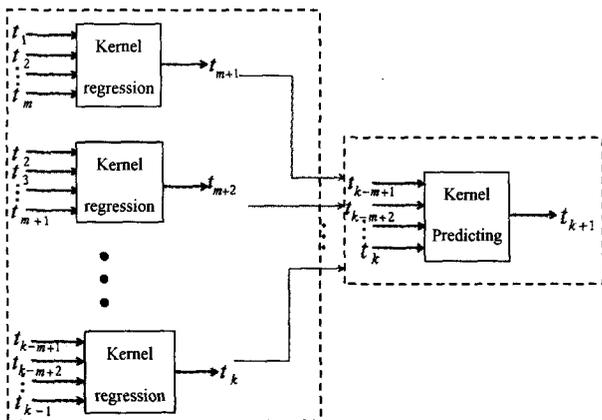


图 1 基于核函数回归估计的软件可靠性预测

2 模型实现步骤

模型的具体实现步骤如图 2 所示, 具体如下。

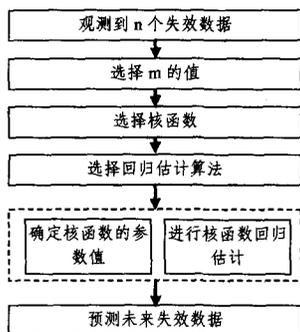


图 2 模型实现步骤

2.1 选择 m 的值

在前期工作中^[15], 作者通过实验研究发现, 早期失效数据对预测未来失效数据作用较小, 与很久之前观测的失效间隔数据现时失效间隔数据能相比, 更好地用以预测未来, 在本文研究中, 采用核函数回归估计理论对软件失效时间数据与发生在其之前的 m 次失效时间数据之间的关系进行建模, 根据文献^[15], m 的取值为 5~15 之间时, 能得到最好的预测效果, 在本文后面的实验部分, m 的取值均为 8。

2.2 使用的核函数

核函数、映射函数以及特征空间是一一对应的, 确定了核函数也就隐含地确定了映射函数和特征空间, 核函数的改变实际上是隐含地改变映射函数, 从而改变样本数据子空间分布的复杂程度, 数据子空间的复杂程度决定了能达到的最小经验误差。所以, 只有首先选择合适的核函数, 将数据投影到合适的特征空间, 才可能得到较好的预测能力以及适用能力。在本文中, 将通过选用多种不同的核函数进行性能分析对比, 寻找适合于描述软件失效过程的核函数结构, 使用的核函数包括:

线性核函数 $\kappa(x, z) = x^T z + c$;

高斯核函数 $\kappa(x, z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$;

sigmoid 核函数 $\kappa(x, z) = \tanh(ax^T z + c)$;

多项式核函数 $\kappa(x, z) = (ax^T z + c)^d$ 。

2.3 核函数估计算法

核函数回归问题求解可以描述为: 给定一群向量 $\{x_i\}_{i=1}^M$ 与对应的目标值 $\{t_i\}_{i=1}^M$ 作为输入, 想要找出 x_i 与 t_i 之间的对应关系, 使得在遇到一个新的向量 x 时, 能够预测出它所对应的目标值 t , t_i 是任意实数。假设 x 与 t 的对应关系符合以下的函数: $t = y(x, w) = \sum_{i=1}^M w_i k(x, x_i) + w_0$, 其中 $k(x, x_i)$ 为核函数, 核函数回归估计算法的目的是找到合适的 w_i 。

本文选取核主成分回归算法、核偏最小二乘回归算法、支持向量回归算法、相关向量回归算法应用于软件可靠性建模, 并对其进行了实验研究。

2.4 确定核函数参数的值

核函数参数选择问题, 其实质就是一个优化问题, 首先将失效数据集按 1:2 分成两份子集, 前面部分作为核函数学习, 而将另外一份数据作为测试数据, 然后采用网格搜索法进行核函数参数选择, 比如在用 SVM 分类时, 采用 RBF 核函数, 此时需要确定两个参数即惩罚因子 C 与核函数参数 σ 。基于网格法将 $C \in [C_1, C_2]$, 变化步长为 C_s , 而 $\sigma \in [\sigma_1, \sigma_2]$, 变化步长为 σ_s , 针对每对参数 (C, σ) 进行训练, 选取效果最好的一对参数作为模型参数。

例如, 使用相关向量机回归估计算法以及高斯核函数时, 为了确定核函数宽度值 σ 对预测精度的影响, 在 DATA-1-DATA-4 上采用不同的 σ 值进行了实验研究, 结果如图 3 所示。实验表明, 随着参数值的增加, 预测精度首先增高, 超过一定值以后, 精度开始下降, 参数的值对预测有很大影响, 但它的最佳取值与具体问题有很大关系, 一般来说, 用于训练的数据量越大, 训练结果对参数的变化越不敏感。在其它回归估计算法和核函数参数的确定中也采用类似的方法。

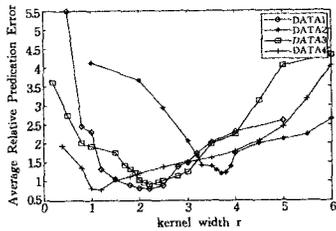


图3 不同核函数宽度条件下 RVM 单步预测的 AE 值变化趋势

3 实验分析与比较

3.1 用于比较的模型

为检验本文所提出的建模思路的有效性,并验证建立的软件可靠性模型的拟合能力及预测能力,对 G-O model、Yamada Delayed S-Shaped、SRGM With Logistic TEF、SRGM With Rayleigh TEF^[15]、Delayed S-Shaped Model With Logistic TEF^[16]、Delayed S-Shaped Model With Rayleigh TEF^[16]等 6 个使用广泛的模型进行了拟合能力、预测能力的比较,用于比较的模型及其均值函数如表 1 所列。

表 1 用于实验比较的 NHPP 模型及其均值函数

模型	均值函数
SRGM With Logistic TEF	$a(1-e^{-bt})$
SRGM With Rayleigh TEF	$a(1-(1+bt)e^{-bt})$
Delayed S-Shaped Model With Logistic TEF	$a(1-e^{-b(\frac{N}{1+Ae^{-\alpha}}-\frac{N}{1+A})})$
Delayed S-Shaped Model With Rayleigh TEF	$a(1-e^{-b(N \times (1-e^{-\alpha^m}))})$
G-O model	$a(1-(1+b(\frac{N}{1+Ae^{-\alpha}}-\frac{N}{1+A}))e^{-b(\frac{N}{1+Ae^{-\alpha}}-\frac{N}{1+A})})$
Yamada Delayed S-Shaped	$a(1-(1+b(N \times (1-e^{-\alpha^m})))e^{-b(N \times (1-e^{-\alpha^m}))})$

3.2 使用的数据集

为了对所建立的模型进行合理的比较与分析,必须使用尽可能多的真实失效数据集。在这一部分,采用 14 个来自不同类型软件的失效数据集^[6]对所提出的模型进行了实验分析,见表 2。这些数据集描述了各个软件系统的失效过程,每个数据点包含两种观测统计集合:累计执行时间和累计失效次数。在实验中,训练集包括从测试开始后完整的系统失效过程,为了让核函数进行充分的学习,在实验过程中,取所有数据集的前三分之一作为学习数据,并对后面三分之二数据进行预测后与真实数据进行比较。

表 2 实验数据集

数据集	代码行数	发生失效数	软件类型
1	1000	27	Class Compiler Project
2	21700	136	Realtime Command and Control
3	40000	46	On-line Data Entry
4	1317000	328	Database Application Software
5	35000	279	Hardware Control Software
6	240000	3207	Application Software
7	870000	535	Realtime Control Application
8	200000	481	Monitoring and Realtime Control
9	14500	55	Railway Interlocking System
10	90000	198	Monitoring and Realtime Control
11	10000	118	Flight Dynamic Application
12	22500	180	Flight Dynamic Application
13	38500	213	Flight Dynamic Application
14	未知	266	Realtime Control Application

3.3 比较标准

所建立的软件可靠性模型为软件可靠性预测模型,使用软件可靠性预测模型性能分析中最常用的指标平均相对预测误差(Average relative prediction Error, AE)对模型进行评价,计算式子为^[6]: $AE = \frac{1}{k-m} \sum_{i=m+1}^k \left| \frac{t_i - \hat{t}_i}{t_i} \right|$ 。式中, k 表示用于核函数学习的失效次数, \hat{t}_i 表示第 i 次失效时间估计值, t_i 表示失效时间实测值,AE 的值越小说明模型的预测能力越强,模型的适用性通过对多个数据集上模型的预测性能进行综合分析得到。

3.4 模型性能分析

表 3 列出了在 14 个数据集上各个模型的 AE 值,其中模型 1-6 分别代表 SRGM With Logistic TEF、SRGM With Rayleigh TEF、Delayed S-Shaped Model With Logistic TEF、Delayed S-Shaped Model With Rayleigh TEF、G-O model、Yamada Delayed S-Shaped;模型 7-12 分别代表 KPCR、KPLS、1-范数 SVR、1-范数 SVR、v-SVR 和 RVR,a、b、c、d 代表采用的核函数分别为高斯核函数、线性核函数、sigmoid 核函数、多项式核函数。从表 3 可以看出:

1)相比于传统随机过程方法,核函数方法能够有效地提高软件可靠性模型的预测能力。在各个数据集上,模型 7-12 的 AE 值总体上比模型 1-6 要小,如在数据集 6 上,模型 1-6 的 AE 值范围为 187.12 至 553.30,而在模型 7-12 上,相应的数值为 5.27~58.99。这主要是因为传统随机过程模型中,模型的参数完全根据前三分之一的失效数据确定,参数不会随着失效数据的更新而更新。而在核函数可靠性模型中,模型参数随着失效数据的增加不断调整,以保证模型中可以捕捉到失效过程的动态特性。

2)在所有随机过程模型中,不存在在 14 个数据集上表现都较好的模型,如模型 SRGM With Logistic TEF 在数据集 9、10、11、14 上表现不佳,模型 SRGM With Rayleigh TEF 在 2、5、7 等数据集上表现不佳,模型 Delayed S-Shaped Model With Logistic TEF 在 2、4、13 等数据集上表现不佳,模型 Delayed S-Shaped Model With Rayleigh TEF 在 6、8、14 等数据集上表现不佳,G-O 模型在大多数数据集上表现都不好,模型 Yamada Delayed S-Shaped 在 1、9、12 等数据集上表现不佳。这说明随机过程模型的适用性较差,主要原因是不同的随机过程模型有不同的假设条件。失效数据集符合随机过程的假设条件时,表现会比较好,否则往往会出现较大偏差。

3)在本文选用的 4 种核函数中,采用高斯核函数和线性核函数时,在多数数据集上,7 种预测算法得到的 AE 值比采用 sigmoid 核函数和多项式核函数时的更小,由此可见,软件可靠性预测中,采用高斯核函数和线性核函数更为合适。

结束语 本文采用核函数回归估计理论,对软件失效时数据之间的对应关系进行建模,在新建立的模型中,随着软件失效不断发生,模型参数将不断自动调整以适应失效过程的动态变化,从而实现软件可靠性的自适应预测。在 14 个数据集上进行的实验比较与分析说明新模型的预测能力以及适用性要好于 6 个 NHPP 类软件可靠性模型。在许多 NHPP 类模型与其它随机过程可靠性模型的比较中已经说明,NHPP 类模型在拟合效果、结构和应用方面具有优势。因此,可以认定本文中所选择的模型有代表意义,新模型与其它随机过程模型相比,依然可以取得类似的结果。

表3 14个数据集上各个模型预测的AE值比较

	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	Data11	Data12	Data13	Data14	
1	3.42	5.37	2.32	8.34	10.22	286.14	18.23	24.25	4.13	3.14	4.22	2.57	6.24	12.12	
2	2.53	6.24	1.35	6.98	13.82	187.12	29.12	27.53	2.02	2.56	3.89	1.65	7.89	7.42	
3	1.43	8.90	4.56	12.23	11.34	327.19	22.21	19.01	3.13	2.87	2.98	1.86	9.34	8.25	
4	1.24	4.43	2.14	9.17	8.42	401.12	15.24	30.23	2.47	2.19	2.66	1.64	7.32	13.56	
5	4.67	3.54	2.67	11.45	16.17	553.30	31.46	28.00	5.52	3.24	5.98	3.33	13.76	18.34	
6	3.55	7.23	1.26	3.64	11.23	298.01	12.36	17.53	4.19	5.15	2.12	3.89	8.34	14.44	
7	a	2.43	5.64	2.48	5.78	7.36	24.45	8.32	6.45	2.76	1.99	1.37	2.02	3.94	6.90
	b	2.87	6.32	2.04	4.36	10.24	35.16	11.35	4.17	2.35	1.00	2.98	1.76	4.05	8.76
	c	3.02	6.44	3.19	6.43	8.10	43.11	6.23	3.59	3.18	2.04	2.88	4.33	1.54	10.32
	d	3.17	4.54	1.98	7.20	4.89	22.12	4.98	12.34	5.23	4.19	1.30	2.09	6.03	13.18
8	a	2.22	4.09	1.03	6.34	11.32	39.31	12.76	5.35	3.77	1.04	0.83	0.85	4.54	6.00
	b	2.12	3.67	1.67	5.28	4.22	47.32	9.02	7.13	3.04	1.35	2.33	2.52	3.38	10.27
	c	4.54	4.03	1.46	4.98	2.30	29.24	14.37	17.44	3.87	2.84	3.09	2.90	5.98	10.33
	d	1.89	3.76	1.83	11.24	8.17	54.33	11.14	14.23	6.20	0.79	5.76	3.99	2.32	7.28
9	a	1.23	2.19	1.48	2.74	3.39	12.24	5.95	3.54	1.87	0.73	0.36	1.02	1.55	3.29
	b	1.02	1.98	0.48	3.89	4.78	10.34	7.32	3.88	1.35	0.48	0.65	1.35	1.20	2.09
	c	2.14	2.03	0.46	3.28	2.65	15.23	4.33	4.19	1.90	0.89	1.73	1.89	0.53	2.45
	d	2.00	3.96	0.76	4.17	6.38	9.38	6.27	5.53	1.44	2.63	2.63	1.27	1.83	4.99
10	a	1.23	2.54	1.03	2.54	7.18	9.34	4.98	2.98	1.32	0.62	0.67	0.67	1.29	1.98
	b	1.05	2.67	0.78	2.19	3.27	8.26	1.24	3.41	1.53	1.36	0.89	0.83	1.76	2.35
	c	2.16	2.73	0.62	3.98	3.18	11.02	5.84	4.18	3.28	0.99	3.45	1.33	2.39	3.66
	d	2.00	5.77	0.91	2.12	3.11	18.25	8.26	9.65	8.45	4.77	2.88	1.72	1.38	2.09
11	a	1.23	1.78	1.53	3.45	4.29	7.36	2.18	1.28	1.00	1.12	0.57	1.00	1.20	1.03
	b	1.05	3.46	1.21	4.23	2.87	12.53	4.32	3.67	1.10	1.13	1.64	1.65	0.99	1.27
	c	2.16	2.03	0.46	3.87	2.10	10.11	5.28	5.44	5.38	2.10	0.89	4.90	1.54	2.85
	d	2.00	2.66	2.76	2.96	3.10	13.14	8.99	3.98	7.45	0.18	1.36	2.08	1.35	3.00
12	a	1.11	0.93	0.36	1.36	1.32	6.21	1.67	1.98	0.58	0.45	0.45	0.39	0.87	1.02
	b	0.98	1.37	0.47	1.78	1.65	5.27	3.42	0.57	1.27	0.36	1.89	0.12	1.29	0.75
	c	1.99	2.02	0.76	2.48	4.23	11.25	0.78	2.21	0.99	1.83	0.77	0.56	1.03	1.98
	d	0.56	2.23	0.59	2.12	3.19	21.37	4.56	4.29	3.27	1.00	0.93	1.92	4.62	0.52

进一步的研究工作包括:

- 1)采用统计学方法对模型中核函数的构建和选择策略进行研究;
- 2)研究模型回归估计算法以及核函数参数的自动赋值算法,以进一步提高模型的预测能力和适用性。

参 考 文 献

[1] Musa J D. Software reliability engineering[M]. New York, McGraw Hill,1999

[2] 邹丰忠,李传湘. 软件可靠性混沌模型[J]. 计算机学报,1999,24(3):281-291

[3] 张勇强,孙胜娟. 基于未确知理论的软件可靠性建模[J]. 软件学报,2006,17(8):1681-1687

[4] 楼俊钢,江建慧,帅春燕,等. 软件可靠性模型研究进展[J]. 计算机科学,2010,37(9):13-19

[5] 楼俊钢,江建慧,靳昂. 考虑样本与均值差别的软件可靠性模型[J]. 计算机学报,2010,30(7):1279-1287

[6] Karunanithi N, Whitley D, Malaiya Y K. Prediction of software reliability using connectionist models[J]. IEEE Trans. Software Engineering,1992,18(7):563-574

[7] Vapnik V. The nature of statistical learning theory[M]. New York, Springer Verlag,1995

[8] Tian L, Noore A. Dynamic software reliability prediction: An approach based on support vector machines[J]. International Journal of Reliability, Quality and Safety Engineering,2005,12(4):309-321

[9] Yang B, Li X. A study on software reliability prediction based on support vector machines[C]// Proc IEEE International Conference on Industrial Engineering and Engineering Management. 2007:1176-1180

[10] Lou J G, Jiang J H, Shuai C Y. Software Reliability prediction based on Transduction inference[C]// IEEE 19th Asian Test Symposium. Shanghai,2010:77-80

[11] Rosipal R, Girolami M, Trejo L J, et al. Kernel PCA for feature Extraction and De-noising in Non-linear Regression[J]. Neural Computing & Applications,2001,10(3):231-243

[12] Rosipal R, Trejo L J. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space [J]. Journal of Machine Learning Research,2001(2):97-123

[13] Tipping. Sparse Bayesian learning and the relevance vector machine[J]. Journal of Machine Learning Research,2001(1):211-244

[14] John S T, Nello C. Kernel Methods for Pattern Analysis[M]. New York:Cambridge University Press,2004

[15] 楼俊钢,江建慧. 软件失效时间数据相关性研究[J]. 计算机应用,2010,30(3):600-603

[16] Kan S H. Metrics and Models in Software Quality Engineering [M]. New Jersey:Addison-Wesley,2003

[17] Huang C Y, Kuo S Y, Lyu M R. An Assessment of Testing-Effort Dependent Software Reliability Growth Models[J]. IEEE Transactions on Reliability,2007,56(2):198-211