基于EMD和GEP的软件可靠性预测模型

张德平[1] 汪 帅[1] 周吴杰[2]
(南京航空航天大学计算机科学与技术学院 南京 210016)[1]
(东南大学计算机科学与工程学院 南京 210096)[2]

摘 要 基于经验模态分解和基因表达式编程算法提出了一种软件可靠性预测模型。通过对软件失效数据序列进行经验模态分解得到不同频段的本征模态分量和剩余分量,消除失效数据中的噪声,运用基因表达式编程算法的灵活表达能力,把分解得到的不同频段的各本征模态分量及剩余分量中所对应的不同失效时间序列作为样本来分别进行预测,重构各本征模态分量和剩余分量中相对应的预测结果,将其作为软件失效的最终预测值。基于两组真实软件失效数据集,将所提出的方法与基于支持向量回归机以及单纯使用基因表达式编程的软件可靠性预测模型进行比较分析。结果表明,该软件可靠性预测模型具有更为显著的模型拟合能力与精确的预测效果。

关键词 经验模态分解,基因表达式编程,软件可靠性预测,可靠性模型
中图法分类号 TP311 **文献标识码** A

Software Reliability Forecasting Model Based on Empirical Mode Decomposition and Gene Expression Programming

ZHANG De-ping[1] WANG Shuai[1] ZHOU Wu-jie[2]
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)[1]
(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)[2]

Abstract A forecasting method based on empirical mode decomposition (EMD) and gene expression programming (GEP) was presented and applied to software reliability forecasting. Firstly, the software failure samples were handled in order to eliminate the pseudo-data, and the intrinsic mode functions (IMFs) and the residue of different frequency bands were obtained according to EMD. Then the corresponding failure data series in the IMFs and the residue were chosen as the training samples. By means of the flexible expressive capacity of GEP, the models of each IMF and the residue were forecasted. Finally, the ultimate forecasting result was obtained by reconstructing the forecasting results of each IMF and the residue. The method of EMD overcomes the shortcomings that it's difficult to select proper wavelet function for wavelet transform, and the final result indicates that the IMFs can reflect the characteristic of software failure. After comparing with the results forecasted by means of combination of SVR and GEP, it proves that the effect of the forecasting method of EMD&GEP in software reliability forecasting is better.

Keywords Empirical mode decomposition (EMD), Gene expression programming (GEP), Reliability prediction, Software reliability model

1 引言

随着现代计算机软件系统的规模越来越庞大、越来越复杂,其可靠性保障的难度也越来越大,计算机软件系统的可靠性已为社会所广泛关注[1]。软件可靠性分析是实施软件可靠性工程、保证软件产品质量的重要实际过程,它贯穿软件开发的所有阶段。软件可靠性分析以数学为工具,综合运用随机过程、数理统计和数据分析等方面的数学知识,对软件的各种质量指标进行各种评估或预测。软件可靠性是评价软件质量的一个重要指标,如何准确无误地对其进行预测是当前软件可靠性工程研究中的热点问题之一[2]。

从20世纪70年代开始,人们在软件可靠性建模、分析、评估和预测等方面进行了大量的研究,提出了近百种软件可靠性模型(Software Reliability Models, SRMs)。软件可靠性模型旨在根据软件失效数据,通过建模给出软件的可靠性估计值或预测值。它不仅是软件可靠性分配、预测与评价的最强有力的工具,还为改善软件质量提供了指南。软件可靠性模型一般可分为分析模型(Analytical Model)和数据驱动模型(Data-driven Model)两类[3]。分析模型即为传统的软件可靠性增长模型(Software Reliability Growth Model, SRGM),主要包括NHPP模型、Markov模型等,这类模型一般需要对软件内部错误、失效及排除过程的特性做出相应的假设,然后

到稿日期:2012-06-20 返修日期:2012-09-10 本文受中央高校基本科研业务费专项资金(NS2012072)资助。
张德平(1973-),男,博士,讲师,主要研究方向为软件测试技术、软件可靠性、数理统计等,E-mail:depingzhang@nuaa.edu.cn(通信作者);汪 帅(1982-),男,硕士生;周吴杰(1973-),男,博士生,讲师,主要研究方向为软件测试、错误定位。

利用某种随机过程进行建模，分析软件可靠性。这些假设包括：软件失效相互独立，立即完全排错，不引入新的软件缺陷，每个软件内部错误对软件失效率的贡献相同等。尽管各种可靠性增长模型的假设条件不尽相同，但这些假设都或多或少与软件失效及排错过程的实际情况有所不符，因此，模型在实际的软件可靠性评估与预测中的适用性和准确性分析都受到了较大的影响[3]。

基于此，近年来数据驱动的软件可靠性预测方法受到了越来越多的重视，国内外学者已提出了一些基于数据驱动的软件可靠性模型。这类模型不需要对软件的失效和排错过程做任何假设，而是将观测到的软件失效数据视为一个时间序列，借助于机器学习技术，对失效数据序列中蕴含的失效过程信息进行挖掘和分析，进而得到软件可靠性的评估和预测结果。基于数据驱动的软件可靠性模型不需要对软件内部错误、失效和排错过程做任何不符合实际的假定，而是直接对失效数据本身建模，因而其适用范围较传统的软件可靠性增长模型更为广泛，具有很好的自适应与自学习能力，并且不存在实际应用中的模型选择问题。相对于传统的可靠性增长模型，基于数据驱动的可靠性模型的适用性和预测精度均有显著的提高，已经得到了广泛的关注和研究，成为当前软件可靠性建模研究的热点。目前，基于数据驱动的软件可靠性模型主要包括基于时间序列分析的 ARIMA 模型[4,5]、人工神经网络模型[6]、支持向量机(Support Vector Machine，SVM)模型[7,8]、灰色理论(Grey Model，GM)模型[9]以及基于基因表达式编程算法的模型[10]等。已有的数据驱动模型中大多数对于平稳的失效数据序列具有较高的模型精度，对于非平稳的失效数据有时并不尽如人意。

由于软件可靠性测试过程的复杂性与不确定性，导致收集到的失效数据集中不可避免地包含噪声信息。研究表明[1,5]，软件发生缺陷的间隔时间具有较大的波动性。对于此类复杂的非线性、非平稳时间序列，使用单一可靠性模型将难以对其进行准确预测。如果采用一种适当的数据处理方法，将蕴含多种成分的软件失效时间序列数据分解为若干个规律性较强的不同频率子时间序列分量，并针对其时频特性选择适当的数学工具建立预测模型，通过预测子序列使预测风险分散化，则可进一步提高预测精度。因此，本文利用经验模态分解(Empirical Mode Decomposition，EMD)方法[11]将软件可靠性数据分解成独立的两部分数据。一部分描绘软件可靠性数据的总体趋势；另外一部分描绘软件可靠性数据随时间的波动趋势。对两部分数据分别利用基因表达式编程(Gene Expression Programming，GEP)算法进行预测并加以组合，从而得到最终的可靠性结果。

2 基本概念及相关算法

2.1 经验模态分解

经验模态分解(Empirical Mode Decomposition，EMD)是一种基于信号局部特征的信号分解方法。该方法吸取了小波变换多分辨的优势，同时克服了小波变换中需选取小波基与确定分解尺度的困难，因此更适于非线性、非平稳信号分析，是一种自适应的信号分解方法，已被成功应用于诸多研究领域[12]。经验模态分解作为一种新的时频分析方法，从本质上讲是对一个信号进行平稳化处理，其结果是将信号中不同尺度的波动或趋势逐级分解开来，产生一系列具有不同特征尺度的数据序列，每一个序列代表一个本征模式函数 IMF(Intrinsic Mode Function)。分解出的各个 IMF 分量突出了数据的局部特征，对其进行分析可以更准确有效地把握原始数据的特征信息。本征模态函数必须满足两个条件：①极值点与过零点数目必须相等或至多相差一点；②在任意点由局部极大值点构成的包络线与局部极小值点构成的包络线的平均值为零。数据序列 $x(t)$ 的经验模态分解步骤如下[13]：

1)找出 $x(t)$ 的所有局部极大值点和局部极小值点。

2)对所有局部极大值点和局部极小值点分别用三次样条函数拟合出数据序列的上、下包络线 $u_0(t)$ 和 $v_0(t)$。

3)记上、下包络线的平均包络线为 $m_0(t)$：

$$m_0(t)=\frac{u_0(t)+v_0(t)}{2} \tag{1}$$

记原数据序列 $x(t)$ 与 $m_0(t)$ 之差为：

$h_0(t)=x(t)-m_0(t)$

4)判断 $h_0(t)$ 是否满足 IMF 的两条性质。若满足，则 $h_0(t)$ 为 IMF；否则，记 $h_0(t)$ 为 $x(t)$，重复步骤 1)－3)；直到得到一个 IMF，记为 $c_1(t)$。

记 $r_1(t)=x(t)-c_1(t)$ 为新的数据序列，重复步骤 1)－4)，得到第二个 IMF，记为 $c_2(t)$。这样一直重复下去，直到余项 $r_n(t)$ 是一个单调数据序列或 $r_n(t)$ 的值小于预先给定的阈值，分解结束。

这样，最终可得到 n 个 IMFs $c_1(t),c_2(t),\cdots,c_n(t)$，余项为 $r_n(t)$。因此，原始数据序列 $x(t)$ 可表示为：

$$x(t)=\sum_{i=1}^{n}c_i(t)+r_n(t) \tag{2}$$

非平稳的失效数据序列经过 EMD 分解以后，各 IMF 分量都基本趋于平稳，这对预测是有利的。而且每一 IMF 分量都是对软件失效样本数据特征的一种真实反应，通过对失效数据序列的各个特征分别进行预测处理，然后对各预测结果分量进行重构，就可以得到更加理想的预测效果。

2.2 基因表达式编程算法

基因表达式编程算法是一种基于基因型组和表现型组的新型自适应演化算法，它是遗传算法和遗传程序设计[11]相结合的产物，它集遗传算法和遗传程序设计的优点于一体，把个体编码成易于进行遗传操作的固定线性串，然后将其表达成长度和形状不同的表达式树形式。在基因表达式程序设计中，终结点集和函数集中元素的选取和遗传程序设计没有太大区别，但是基因的构成分为头部和尾部两部分。头部既可以是终结点集(Terminals)中的元素，也可以是函数集(Functions)中的元素，而尾部元素只能限制在终结点集(Terminals)中。

GEP 与 GA 和 GP 在算法上相差不大，根据给定问题，首先要定义终结点集 T 和初始函数集 F，然后确定适应度评价方法，并给定运行控制量，最后需要确定终止运行的标准。进化过程中染色体被表达成表达式树，通过一系列遗传操作生成新的个体，当遗传代数或适应度值达到预定值时，终止进化过程。在基因表达式程序设计中，染色体通常由多个等长的基因构成，且基因个数和基因头部的长度都是预先选定的。每一个基因被表达成一个子表达式树，子表达式树间相互作用构成更复杂的多子树表达式树，这样一些复杂的问题就可以被表示出来了。基因表达式程序设计中的遗传操作主要包

括选择、变异、变换和重组，变换操作实际上就是变异操作，而重组操作可以看作是交叉操作。其具体步骤如下[10]：

Step1　初始化种群：随机生成数目一定的个体集合作为初始种群，个体编码为线性符号串。

Step2　计算种群中各个体的适应值：如果获得满意的适应值或满足其它终止条件（例如总的进化代数达到预定值或在一定的进化代数内，个体的适应度值没有发生明显的变化），保存结果，退出；否则，进入下一步。

Step3　精英保留：保存最优个体。

Step4　选择：根据适应值的大小进行选择，适应值大的被选中的概率高，适应值小的被选中的概率低。

Step5　复制：将被选中个体直接复制到下一代。

Step6　有修饰复制：对被选中个体，按照一定的概率进行遗传操作，如变异、重组、转座，形成新个体。

Step7　形成新种群：由经过选择复制和其它遗传操作后的个体形成新种群，返回 Step2。

3　基于 EMD 和 GEP 的可靠性模型

不同的软件错误、缺陷及其故障在表现形式、性质乃至数量方面可能大相径庭，这使得原始的软件失效数据序列具有很大的非平稳性。引起这种数据现象的主要原因为：在不同功能区选取测试用例进行测试会得到不同的失效数据分布；软件可靠性测试的输入域数据的分布与选取会影响时间域内的失效数据分布趋势；测试人员的变更与学习过程会对失效数据的分布趋势有一定影响；另外，随机测试与重点测试的混合测试策略也是失效数据产生突变的原因之一。通过对大量软件失效数据的研究分析发现，由于原始的软件失效数据间隔时间的非平稳性，导致其最终预测结果产生极大的误差，特别是在波峰、波谷处。如何描绘其波动性趋势，构建软件可靠性数据的波动模型，是解决问题的关键。

为解决上述问题，这里采用 EMD 方法将软件失效数据分解成独立的两部分数据：一部分描绘软件可靠性数据的总体趋势；另外一部分描绘软件可靠性数据随时间的波动趋势。记 t 时刻的软件失效数据为 $x(t)$，则 $x(t)=P(t)+Q(t)$，其中 $P(t)$ 用来描绘软件可靠性数据随缺陷出现的波动趋势，$Q(t)$ 用来描绘软件可靠性数据的总体趋势。具体地，我们采用经验模态分解方法将软件可靠性数据分解为一系列的本征模态分量 $c_i(t)$ 和 1 个剩余分量 $r_n(t)$，即

$$P(t)=\sum_{i=1}^{n}c_i(t),Q(t)=r_n(t) \tag{3}$$

对失效样本数据序列进行经验模态分解后得到本征模态分量、剩余分量，且各 IMF 分量都基本趋于平稳，这对预测是有利的。而且每一 IMF 分量都是对失效样本数据特征的一种真实反应，通过对失效数据序列的各个特征分别进行预测处理，再对各预测结果分量进行重构，就可以得到更加理想的预测效果。这样不仅可以使失效数据的特征在不同的分辨率下显露出来，而且由于这种分辨率是自适应的，因此与小波的多分辨率分析相比，采用 EMD 分解信号具有更好的效果。

对软件失效样本数据进行 EMD 分解以后，对每一 IMF 分量分别利用 GEP 进行预测，并重构各分量预测结果，以得到最终预测结果。具体的预测过程是首先对失效样本数据序列进行经验模态分解，分解的结果是得到一系列的本征模态 IMF 分量 $c_i(t)$ 和 1 个剩余分量 $r_n(t)$。对于每一个 IMF 分量 $c_i(t)$ 和 $r_n(t)$ 分别运用 GEP 算法进行演化训练。GEP 算法的参数设置如表 1 所列[10]。

表 1　基因表达式编程的参数设置

参　数	设定值	参　数	设定值
种群大小	100	倒位概率	0.1
进化代数	150000	RIS 变换概率	0.1
染色体数目	80	基因变换概率	0.1
基因头部长度	15	单点重组概率	0.3
基因数目	5	两点重组概率	0.3
各基因间连接函数	+	基因重组概率	0.1
变异概率	0.044	随机常数个数	10
IS 变换概率	0.1		

这里将经典软件可靠性增长模型中使用的初等函数加入到函数集，函数集 $F=\{+,-,*,/,\log,\exp,pow(x,y)\}$，其中 log 表示自然对数，exp 表示 e 的指定次幂，$pow(x,y)$ 表示 x 的 y 次幂等。终结点集选取 $T=\{t,r,s\}$，t 为累积失效次数（以下简称为失效次数）；r 为 0～9 之间的随机整数，运行过程中随机生成，s 为失效间隔。这里采用在可靠性领域中常用的 4 种拟合或预测评价指标作为 GEP 的适应值函数[1,6]：

(1) 均值误差平方和（mean square error，MSE）

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-y_i')^2 \tag{4}$$

(2) 回归曲线方程的相关指数（R-Square 或 R 值）

$$R\text{-}Square=1-\frac{\sum_{i=1}^{n}(y_i'-y_i)^2}{\sum_{i=1}^{n}(y_i-y_{ave})^2} \tag{5}$$

(3) 均值误差（average error，AE）

$$AE=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-y_i'}{y_i}\right|\times 100 \tag{6}$$

(4) 均方百分比误差（MSPE）：

$$MSPE=\frac{1}{n}\sqrt{\sum_{i=1}^{n}\left(\frac{y_i-y_i'}{y_i}\right)^2} \tag{7}$$

以上各式中，y_i 表示数据的实际值，y_i' 表示数据的预测值，y_{ave} 表示观测数据 y_i 的均值。显然，AE 值、MSE 值和 MSPE 值越小，R-Square 值越接近 1，表明预测值与实际值越接近，模型拟合或预测性能越好。

这样，每个 IMF 分量 $c_i(t)$ 和剩余分量 $r_n(t)$ 都分别会演化出一个预测模型，将各个预测模型分别代入变量 t 的值即可得到各分量所对应失效次数的软件可靠性预测结果；然后对各 IMF 分量 $c_i(t)$ 和 $r_n(t)$ 所对应的可靠性预测结果进行重构，则可得到最终的软件可靠性预测结果。为了比较的方便，本文称之为 EMD&GEP 预测法，预测模型可表示为：

$$F(t)=\sum_{i=0}^{n}C_i(t)+R(t) \tag{8}$$

式中，$F(t)$ 由两部分构成，前者为 EMD 分解后各本征模态分量对应的相应时刻的预测结果之和，后者为剩余分量所对应的相应时刻的预测结果；变量 n 表示本征模态分量的总数；i 表示第 i 个本征模态分量。

4　实例及灵敏度分析

在若干组真实失效数据集上分别对 EMD&GEP 与单纯的 GEP（以下简称 GEP 模型）以及 SVR（支持向量回归）进行对比分析。所选取的对比模型均为该类研究中最新或具有代

表性的典型成果。每个实例中所选取的失效数据集均为公开发表的，且常被用于软件可靠性模型评估或预计性能比较的经典范例。为方便与已有研究成果进行比较，用于模型训练（利用历史数据来确定模型参数的估计值，同时计算模型对历史数据的拟合程度）与模型预计（在模型训练基础上，对未来失效行为进行预计）的失效数据比例均与对比模型相同。

4.1 数据预处理

由于软件可靠性测试过程的复杂性与不确定性，导致收集的失效数据集中不可避免地包含一些不良数据。不良数据虽有些不属于错误数据，但随机性太强，会很大程度上影响模型精度和程序进化速度，因此在进行软件可靠性建模之前，要对原始数据中的不良数据值进行必要的预处理。

根据软件失效数据的趋势分析和时序特征，这里采用层次聚类的凝聚法[14]进行同类故障密度的失效数据的聚类，找到单点类的异常数据点，然后利用三次样条曲线去拟合异常数据前后 4 个数据，再确定异常点上的失效数据值。如对实例 1 中的失效数据集，采用凝聚法对真实的失效数据聚类，容易发现失效次数为 21 和 24 的失效间隔可能存在异常。实验分析表明，数据预处理强度的大小会直接影响 EMD 算法对原始数据平稳化处理时收敛速度的快慢。运行环境操作系统：Windows 7，编程语言：Java，CPU：AMD A6，内存容量：2GB DDR3。

4.2 实例分析

实例 1　表 2 的数据来自一个大型通信软件项目的失效数据集[15]，记录了软件的失效次数以及失效间隔。

表 2　失效数据集

失效次数	失效间隔	失效次数	失效间隔
1	2036	21	220
2	11779	22	35580
3	40933	23	81000
4	34794	24	643095
5	17136	25	47857
6	148446	26	154800
7	7995	27	170460
8	1636	28	108540
9	15830	29	73800
10	21932	30	1860
11	2485	31	336600
12	11000	32	268140
13	2880	33	74880
14	61182	34	286200
15	4800	35	25320
16	38005	36	7080
17	16200	37	59820
18	6000	38	87900
19	1000	39	76200
20	10000	40	89280

通过对失效数据集进行预处理，利用经验模态分解方法，得到失效数据集的经验模态分解结果，如图 1 所示。

通过 EMD 分解将原始数据分为 6 个本征模态分量和 1 个剩余分量，相较于原始数据，每个本征模态分量明显趋于平稳，且频率依次降低。选取每个分量的前 31 个作为训练数据，后 9 个作为预测数据，并用 GEP 对其进行演化训练，这样就会得到 7 个分量的预测模型。最后将这 7 个分量模型进行重构，便可得到较为理想的预测模型。

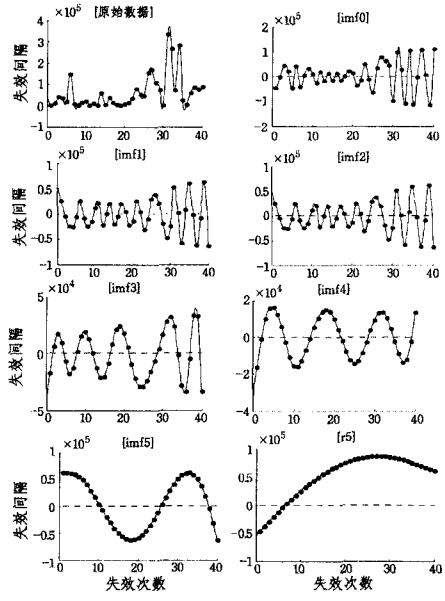

图 1　经验模态分解结果

表 3 给出了分别利用 EMD&GEP、GEP 与 SVR 预测方法对表 2 的失效数据进行预报时得到的预测结果及相应评价指标值。

表 3　各模型的预测结果

数据集		模型		
失效次数	失效间隔	SVR	GEP	EMD&GEP
32	268140	275100	268801	287824
33	74880	71654	68909	76398
34	286200	299905	273465	276485
35	25320	23650	24465	24615
36	7080	——	——	——
37	59820	60980	60558	61316
38	87900	81354	67005	85467
39	76200	75980	75681	76488
40	89280	88450	87762	88953
R-Square		0.9655	0.9943	**0.9973**
AE		3.61	5.43	**2.69**
MSPE		0.154	0.032	**0.012**
MSE		**3.67E7**	7.98E7	6.16E7

注：(1)“——”表示该组数据的预测效果远小于同列的数值；
(2)加粗部分为该行中的最佳结果。

由表 3 可以看出在本组数据集上，EMD&GEP 模型的预测结果要优于 GEP 和 SVR 模型的预测结果。其中有 3 种评价指标值（R-Square 值为 0.9973，AE 值为 2.69，MSPE 值为 0.012）均为最优，仅在 MSE 值上劣于 SVR 模型。由此可证明，EMD&GEP 模型在整体上有比前二者更为理想的预测性能。

为了更好地验证模型的稳定性，可分别对 3 种预测模型进行 20 次建模（其中 EMD&GEP 和 GEP 算法终止条件是算法演化代数达到 100000），再对各组模型的预测结果进行对比。分析结果如图 2 所示。

由图 2 可得出，从总体来看 EMD&GEP 模型比 SVR 模型要稳定，且在相同的设置条件下比 GEP 模型更为稳定，即 EMD&GEP 模型在 4 种预测评价指标值上的波动性更小。

实例 2　本实验以软件可靠性评测领域的权威 J. D Musa 发布的 Sys1. dat 失效数据[1]为基础。SYS1. dat 是一个记录有 136 个失效的失效间隔时间的数据集。为了评判采用

EMD&GEP 方法所得到的表达式的预测能力，这里取前 129 个失效作为训练数据，留下第 130 号到 136 号失效数据作为预测结果的对比数据。

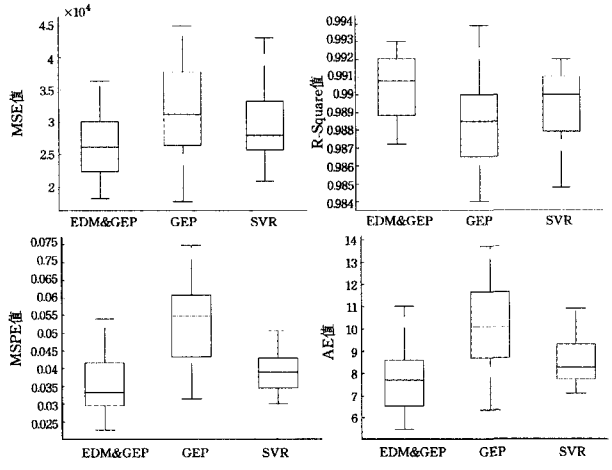

图 2　模型稳定性对比

图 3 给出了数据集 SYS1. dat 通过 EMD 分解将原始数据分为 8 个本征模态分量和 1 个剩余分量的曲线图。

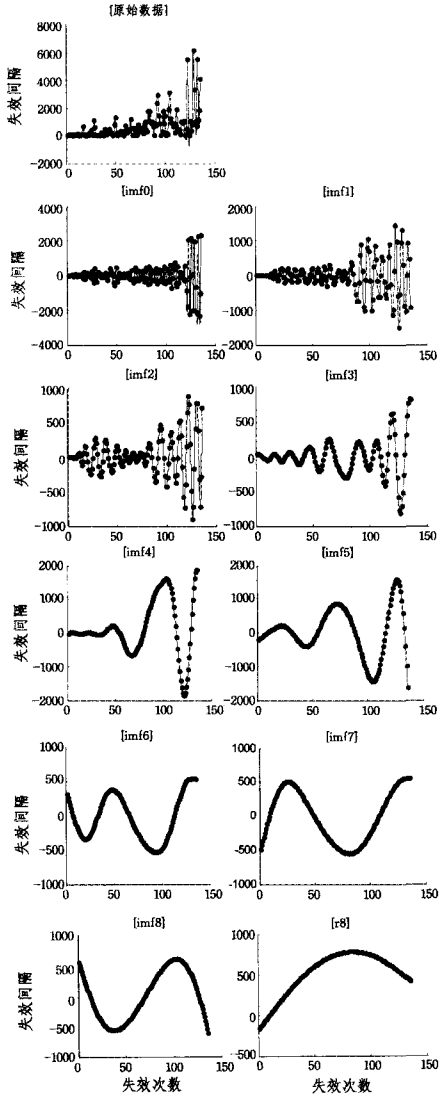

图 3　SYS1 数据集的经验模态分解结果

表 4 给出了分别利用 EMD&GEP、GEP 与 SVR 方法对 SYS1. dat 数据进行预测的结果及相应评价指标值。

表 4　SYS1 数据集上各模型的预测结果

数据集		模型		
失效次数	失效间隔	SVR	GEP	EMD&GEP
130	3321	3551	2978	3173
131	1045	1387	853	1180
132	648	785	530	697
133	5485	5290	5121	5318
134	1160	972	1286	1310
135	1864	1945	1723	1674
136	4116	3667	3823	4370
R-Square		0.976	0.979	**0.990**
AE		13.69	11.30	**8.18**
MSPE		0.064	0.046	**0.034**
MSE		67166	60362	**27647**

EMD 算法能够完成对原始数据的特征提取，又由于 GEP 算法有着特殊的编码方式和遗传算子，使得 EMD&GEP 算法能够在绝大多数数据集上有着不错的拟合和预测结果。由表 4 可得出，在本组失效数据集上，EMD&GEP 模型的预测结果均优于对比模型，即 AE 值、MSPE 值和 MSE 值最小，R-Square 值最接近于 1。另外由图 3 可看出，未结合 EMD 的 GEP 模型其预测的稳定性较差且稍劣于 SVR 模型，而 EMD&GEP 的效果要明显优于 SVR 模型。

为了更好地验证模型的稳定性，分别对 3 种预测模型进行 20 次建模（其中 EMD&GEP 和 GEP 算法的终止条件是算法演化代数达到 100000），图 4 为各组模型的预测结果的盒状图。

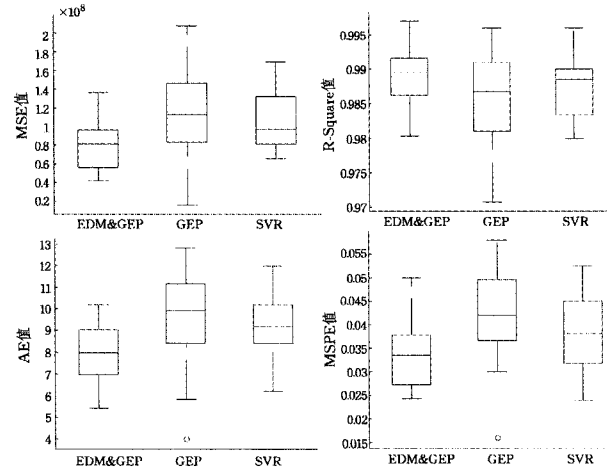

图 4　SYS1 数据集上模型稳定性对比

由图 4 可得出，从总体来看 EMD&GEP 模型比 SVR 模型要稳定，且在相同的设置条件下比 GEP 模型更为稳定，即 EMD&GEP 模型在 4 种预测评价指标值上的波动性更小。

结束语　经验模态分解（EMD）解决了小波变换中需选取小波基与确定分解尺度的困难，是一种自适应的信号分解方法。其可以对非线性非稳定的失效数据序列进行逐层分解，获得若干个 IMF 分量和 1 个剩余分量，完成对原始数据的特征提取。基因表达式编程（GEP），是借鉴生物遗传的基因表达规律提出的一种新的效率较为理想的演化算法，能够根据与问题相关的终结点集和函数符集，生成与历史数据相拟合的预测模型。本文将 EMD 和 GEP 算法相结合，用 EMD 算法对真实的软件失效数据进行分解得到一系列的本征模态

（下转第 184 页）

表 4　聚类算法结果的精确度

ur_{kj} 取值	结果的精度
0	0.7855
1	0.6855
2	0.8959
3	0.7337
4	0.8959
聚类精度平均值	0.7993

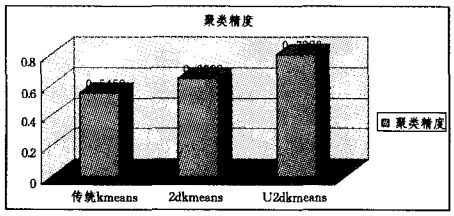

图 2　数据集 Breast Cancer Wisconsin (Diagnostic)的对比图

为了更加直观地对比各个算法的效果，将各个算法在 Breast Cancer Wisconsin (Diagnostic)数据集上的聚类效果用直方图的形式描述出来，如图 2 所示。显然 U2d-Kmeans 算法对 Breast Cancer Wisconsin (Diagnostic)数据集的聚类效果既客观又合理，效果优于其他两种。

结束语　传统聚类算法大多没有考虑数据对象的不确定因素，只是简单地消除数据的不确定成分，这种数据预处理方式会影响聚类效果。为了得到真实的聚类结果，本文考虑到数据对象的不确定成分，提出一种不确定域环境下基于 DKC 值改进的 K-means 聚类算法。另一方面，该算法还借鉴了 2d-Kmeans 算法中对孤立点和初始点的处理方法，分 3 步进行：(1)计算每个数据对象的 DKC 值；(2)根据 DKC 值对原始样数据集剔除孤立点；(3)对 DKC 值排序，根据累积距离的方法确定初始聚类中心。实验结果表明，该算法比传统的 K-means、2d-Kmeans 聚类算法有更好的聚类效果。但是，值得注意的是，聚类算法中引入数据对象的不确定性因素会给算法带来复杂性问题，这是今后研究的重点。

参 考 文 献

[1] Han Jia-wei, Kamber M. Data Mining: Concepts and Techniques [M]. Morgan Kaufmann Publishers, 2001

[2] 李光宇. 基于改进的 CLARANS 算法在数据挖掘中的研究[J]. 中南林业科技大学学报，2010，3：142-145

[3] 原福永，张晓彩，罗思标. 基于信息熵的精确属性赋权 K-means 聚类算法[J]. 计算机应用，2011，31(6)：1675-1677

[4] 姚丽娟，罗可，孟颖. 一种基于粒子群的聚类算法[J]. 计算机工程与应用，2012，13

[5] 储岳中，徐波. 动态最近邻聚类算法的优化研究[J]. 计算机工程与设计，2011，32(5)：1687-1690

[6] 杨臻. 基于 2k-距离的孤立点算法研究[J]. 福建电脑，2009，2：77-78

[7] 陈福集，蒋芳. 基于 2d-距离改进的 K-means 聚类算法研究[J]. 太原理工大学学报，2012，43(2)：114-118

[8] 刘位龙. 面向不确定性数据的聚类算法研究[D]. 济南：山东师范大学，2011

[9] Pfoser D, Jensen C S. Capturing the Uncertainty of Moving-ObjectRepresentations[C]// Proceedings of the 6th International Symposium on Advances inSpatial Databases. 1999：111-132

[10] UCI Machine Learning Repository [DB/OL]. http://archive.ics.uci.edu/ml/, 1992-07-16

[11] Ahmad A, Dey L. A K-mean clustering algorithm for mixed numeric and categorical data[J]. Data and Knowledge Engineering, 2007, 63：503-527

[12] 王茜，张鲲鹏. 隐私保护数据挖掘算法 MASK 的改进[J]. 重庆理工大学学报：自然科学版，2012，26(6)：63-66

(上接第 168 页)

分量和剩余分量，然后用 GEP 算法对上述的各个分量进行演化训练，最后将产生的各个预测模型进行重构，得到最终预测模型。根据对比实验可知，EMD&GEP 模型在预测精度上效果更为理想，稳定性能上比 GEP 模型更为优秀。

参 考 文 献

[1] Lyu M R. Handbook of software reliability engineering[M]. New York: McGraw Hill, 1996

[2] 赵亮，王建民，孙家广. 统计测试的软件可靠性保障能力研究[J]. 软件学报，2008，19(6)：1379-1385

[3] Yang B, Li X, Xie M, et al. A generic data-driven software reliability model with model mining technique[J]. Reliability Engineering and System Safety, 2010, 95：671-678

[4] Raja U, Hale D P, Hale J E. Modeling software evolution defects: a time series approach[J]. J. Softw. Maint. Evol.: Res. Pract., 2009, 21：49-71

[5] 贾治宇，康锐. 软件可靠性预测的 ARIMA 方法研究[J]. 计算机工程与应用，2008，44(35)：17-19

[6] Su Y S, Huang C Y. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models[J]. The Journal of Systems and Software, 2007, 80：606-615

[7] Moura M C, Zio E, Lins I D, et al. Failure and reliability prediction by support vector machines regression of time series data [J]. Reliability Engineering and System Safety, 2011, 96：1527-1534

[8] Lo J H. A study of applying ARIMA and SVM model to software reliability prediction[C]// 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering. 2011：141-144

[9] 李海峰，陆民燕，王智新. 基于灰色系统理论的软件可靠性综合评价框架[J]. 北京航空航天大学学报，2008，34(11)：1261-1265

[10] 李海峰，陆民燕，曾敏，等. 基因表达式编程在软件可靠性建模中的应用[J]. 计算机科学与探索，2011，5(6)：534-546

[11] Huang N E, Shen Z, Long S R. Anew view of nonlinear waves: the hilbert spectrum[J]. Annual Review of Fluid Mechanics, 1999, 31：417-457

[12] 玄兆燕，杨公训. 经验模态分解法在大气时间序列预测中的应用[J]. 自动化学报，2008，34(1)：97-101

[13] Dong Y, Wang J Z, Jiang H, et al. Short-term electricity price forecast based on the improved hybrid model[J]. Energy Conversion and Management, 2011, 52：2987-2995

[14] 马飒飒，陈自力，赵守伟. 基于聚类的软件失效数据预处理[J]. 计算机工程与应用，2006，11：106-109

[15] Bandara P K, Wikramanayake G N, Goonethillake J S. Software reliability estimation based on cubic splines[C]// Proceedings of the World Congress on Engineering. 2009：12-15