

基于 K 均值的软件测试集用例约简算法研究

高丑光, 林 都, 鲜 浩

(中北大学 计算机与控制工程学院, 山西 太原 030051)

摘 要: 本文提出了一种基于模糊 K 均值的软件测试集用例约简算法, 尽可能的发现相似的用例, 旨在进行最大化的精简. 实验结果表明, 此算法能够最小化约简用例集, 并且对比分析其他算法, 用例集覆盖范围最广泛, 错误率检测较高.

关键词: K 均值; 软件测试; 用例约简; 复杂度; 最大化精简; 错误率监测; 模糊聚类

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2016)05—0133—04

Research of Software Testing Case Reduction Algorithm Based on K Means

GAO Chou-guang, LIN Du, XIAN Hao

(Computer and Control Engineering College, North University of China, Taiyuan 030051, China)

Abstract: In this paper, a software testing based on fuzzy K-means set case reduction algorithm, as far as possible find similar cases, aimed at maximizing streamlined. Experimental results show that the algorithm can minimize the reduction case set, and the analysis of other algorithms, the use case set covers the most extensive.

Key words: K means; software testing; use case reduction; complexity; maximize streamline; error rate monitoring; fuzzy clustering

1 引言

随着软件规模增大, 逻辑复杂程度上升, 因此对软件可靠性要求相对较高. 影响软件可靠性的一个关键因素是软件中隐藏的错误, 这些错误可以导致计算机软件运行错误, 造成灾难性后果. 传统软件错误采用设置断点等人工测试方法, 定位难度大, 耗时长, 因此快速准确的定位和消除软件错误, 已经得到了广泛的关注. 测试用例数量决定测试成本和有效性, 覆盖度全面的测试用例能够精确定位错误. Yu 等人提出基于向量的用例约简算法^[1]. Chen 等人分析软件错误之间的联系, 提出一种轻量级的用例约简方法^[2]. Zhang 等人在测试用例约简中引入相对冗余思想, 提出均衡覆盖有助于错误定位^[3]. 郝丹等人基于语句覆盖提出采用多种约简策略, 采用故障

定位提高语句覆盖集合, 定位错误^[4]. 周冲波等人定义一种不会改变测试需求与测试覆盖关系的布尔运算, 基于矩阵行列变化得到最精简的测试需求集和用例集^[5]. 苏小红等人结合测试用例约简和联合依赖概率, 提出了一种软件错误自动定位算法, 统一用例约简与软件错误定位为一个整体, 实现精准定位错误语句^[6].

软件测试用例集约简已经得到了广泛的研究, 但是仍然存在以下问题: 如何充分利用软件测试用例集之间的关系, 提高错误定位精准程度; 实现一种自动用例约简算法, 简化参数设置. 为了解决上述问题, 本文提出了一种基于模糊 K 均值的软件测试用例约简算法, 该算法引入模糊划分思想, 结合测试需求集, 从各个簇中抽取测试用例, 以便能够得到简化的用例集, 同时保持较高的测试精确度.

2 背景理论

随着测试用例集约简问题的研究和发展,很多学者不断地将聚类思想引入其中. 模糊聚类方法,将样本对各个类隶属度分配为区间 $[0,1]$ 连续数值,体现了样本与样本之间的相互关系,实现了数据样本对类的不确定性描述,可以完成对类与类之间有相同数据集做有效聚类,能更好地表达数据和更好的聚类效果,其聚类效果也比硬聚类优越.

假设数据集包含 n 个样本,使用 $X = \{x_1, x_2, \dots, x_n\}$ 表示,每一个样本都可以使用 m 个特征刻画,则第 j 个样本的特征向量使用 $x_j = \{x_{1j}, x_{2j}, \dots, x_{mj}\}^T$,则数据集可以使用矩阵进行描述,如式(1)所示.

$$X = \begin{bmatrix} x_{11}, x_{12}, \dots, x_{1n} \\ x_{21}, x_{22}, \dots, x_{2n} \\ \dots, \dots, \dots, \dots \\ x_{m1}, x_{m2}, \dots, x_{mn} \end{bmatrix} \quad (1)$$

式中, x_{ij} 表示样本 j 的特征 i , $j = 1, 2, \dots, n$, $i = 1, 2, \dots, m$. 在模糊聚类处理过程中,为了能够消除数据集 m 个特征值之间的量纲差别,需要对其进行归一化处理,具体处理过程如式(2)所示.

$$r_{ij} = \frac{x_{ij} - x_{i\min}}{x_{i\max} - x_{i\min}} \quad (2)$$

式中, $x_{i\max}$ 表示第 i 个指标特征的最大取值; $x_{i\min}$ 表示第 i 个指标特征的最小取值; r_{ij} 表示归一化后 x_{ij} 的取值.

归一化之后,数据集的描述矩阵 X 可以使用矩阵 R 表示,如式(3)所示:

$$R = \begin{bmatrix} r_{11}, r_{12}, \dots, r_{1n} \\ r_{21}, r_{22}, \dots, r_{2n} \\ \dots, \dots, \dots, \dots \\ r_{m1}, r_{m2}, \dots, r_{mn} \end{bmatrix} = (r_{ij})_{m \times n} \quad (3)$$

假设数据集 X 拥有 C 个类别,则数据集 X 的模糊识别矩阵如式(4)所示:

$$U = \begin{bmatrix} u_{11}, u_{12}, \dots, u_{1n} \\ u_{21}, u_{22}, \dots, u_{2n} \\ \dots, \dots, \dots, \dots \\ u_{m1}, u_{m2}, \dots, u_{mn} \end{bmatrix} = (u_{hj})_{c \times n} \quad (4)$$

式中, u_{hj} 表示数据集中的样本 j 归属于 h 类的隶属度, $h = 1, 2, \dots, C$, 并且满足以下约束条件: $0 \leq u_{hj} \leq 1$, $\sum_{h=1}^c u_{hj} = 1$, $\sum_{j=1}^n u_{hj} > 0$.

假设类别 h 的 m 个特征值称为类的聚类中心,则 c 个类别的特征值可以使用模糊聚类中心矩阵如式(5)所示:

$$S = \begin{bmatrix} s_{11}, s_{12}, \dots, s_{1c} \\ s_{21}, s_{22}, \dots, s_{2c} \\ \dots, \dots, \dots, \dots \\ s_{m1}, s_{m2}, \dots, s_{mc} \end{bmatrix} = (s_{hi})_{m \times c} \quad (5)$$

式中, s_{hi} 表示类别 h 指标 i 的归一化特征值, $0 \leq s_{hi} \leq 1$.

为达到不同的聚类要求,在模糊聚类执行过程中,可以设置不同的特征权重,一般能够优先突出较为重要的特征向量,特征权重向量如式(6)所示:

$$W = \{\omega_1, \omega_2, \dots, \omega_m\}, \sum_{i=1}^m \omega_i = 1 \quad (6)$$

通过以上分析,可得到模糊聚类的目标函数如式(7)所示:

$$\min\{F(u_{hj}) = \sum_{j=1}^n \sum_{h=1}^c \{u_{hj} [\sum_{i=1}^m (\omega_i (r_{ij} - s_{hi}))^p]^{\frac{1}{p}}\}^2\} \quad (7)$$

在实际应用过程中,具体算法可以采用迭代执行的过程,以此来求取目标函数的最优解.

3 基于模糊 K 均值的软件测试用例约简算法

在软件测试用例约简过程中,假设测试用例集 T 与测试需求集 R 之间的二元关系形式化描述为 $S(T, R) = \{(t, r) \in T \times R: \text{测试用例 } t \text{ 满足测试需求 } r\}$, 其表示测试用例 $t \in T$ 与测试需求 $r \in R$ 的满足关系. 假设测试需求集为 $R = \{r_1, r_2, \dots, r_m\}$, 测试用例集 $T = \{t_1, t_2, \dots, t_n\}$ 满足所有的测试需求 r_i . 测试用例约简问题是指在测试用例集 T 中寻找一个最简化的子集 T' , 并且满足 T' 的任一真子集 T'_1 都无法充分测试需求集, 因此 T' 是最简化的测试用例集, 使用 $\text{Req}(T')$ 、 $\text{Req}(T'_1)$ 表示测试用例集 T' 和 T'_1 的测试需求集合, 测试用例约简问题可以形式化描述为 $\text{Req}(T') = R, \forall T'_1 \subset T', \text{Req}(T'_1) \neq R$.

为了更好地实现软件测试用例约简, 本文采用模糊 K 均值聚类算法进行操作, 模糊 K 均值形式化描述如下: 假设测试用例集为 $T = \{t_1, t_2, \dots, t_n\}$, 数据集的簇数目为 K 个, m_i 表示第 i 个簇的中心, $i = 1, 2, \dots, k$. $u_j(t_i)$ 表示 K 均值聚类过程中第 t_i 个样本对第 j 类的隶属度, 模糊 K 均值的目标函数可以使用式(8)描述:

$$F(u_j(t_i)) = \sum_{i=1}^k \sum_{m=1}^n [u_j(t_i)]^b \|t - m_i\|^2 \tag{8}$$

式中, b 是一个可以控制模糊聚类结果的模糊度常数,通过对模糊 K 均值隶属度函数求导数,即可得到 K 均值算法的最优解,如式(9)、(10) 所示。

$$m_j = \frac{\sum_{i=1}^n [u_j(t_i)]^b t_i}{\sum_{i=1}^n [u_j(t_i)]^b}, \tag{9}$$

$$j = 1, 2, 3, \dots, k$$
$$u_j(t_i) = \frac{(1/ \|t - m_j\|^2)^{1/(b-1)}}{\sum_{i=1}^k (1/ \|t - m_i\|^2)^{1/(b-1)}}, \tag{10}$$

$$i = 1, 2, 3, \dots, n;$$
$$j = 1, 2, 3, \dots, k$$

在聚类算法执行过程中,对式(9)、(10) 进行迭代执行,得到具体的模糊 K 均值聚类算法,在实际的数据集划分过程中进行使用。

本文基于模糊思想的 K 均值聚类算法如下所示。

输入:簇数目 K , 参数 b , 包含 N 个数据对象的数据集。

输出: K 个簇。

步骤:(1) 采用随机初始法为数据集设定 K 个簇,并指定每一个簇的中心为 m_i ;

(2) 计算数据集中每一个数据对象的隶属函数,计算方法为式(10);

(3) 基于步骤(2) 的隶属度函数计算各个簇的中心值 m_i ,计算簇中心时候采用式(9);

(4) 遍历数据集中每一个数据对象,当隶属度不再发生任何变化时,算法终止;否则返回步骤(2)。

4 实验及结果分析

4.1 数据集

实验数据采用 Siemens Corporate Research 的研究学者研究程序控制流和数据流覆盖准则的错误检测能力的公用数据集。每一个数据集的原始测试用例集及错误数表 1 所示。

4.2 评价标准

为了更好地验证本文算法的有效性,正确性和稳定性。仿真实验采用约简率、错误检测率和错误检测丢失率三个方面来评价,评价标准详细描述如下:
 $|T|$ 表示原测试用例集中测试用例数目; $|T'|$ 表示

表 1 软件测试用例约简数据集

实验数据集	初始测试用例数	程序中的错误数
Print_tokens1	30	20
Print_tokens2	30	18
Replace	30	22
Schedule1	32	24
Schedule2	32	32
tacs	32	24
tot_info	32	28

约简后测试用例集包含的用例数; $|F|$ 表示原测试用例集检测的错误数; $|F'|$ 表示约简的测试用例集检测的错误数;SR 表示约简率即原测试用例集中测试用例数目与约简后测试用例集中测试用例数目的差与原测试用例集中测试用例数目的比值(如式(11) 所示);FDE 表示错误检测率即算法检测到的错误个数与实际错误个数的比(如式 12 所示);FL 表示误检测丢失率即原测试用例集检测到的错误数与约简后测试用例集的检测到的错误数的差与原测试用例集检测的错误数的比值(如式 13 所示)。

$$SR = \frac{|T| - |T'|}{|T|} \times 100 \tag{11}$$

$$FDE = \frac{|F'|}{|F|} \times 100 \tag{12}$$

$$FL = \frac{|F| - |F'|}{|F|} \times 100 \tag{13}$$

由以上三个公式可以比较三种方法的约简率、错误检测率和误差检测丢失率。

4.3 实验结果分析

为了验证模糊 K 均值用例约简算法有效性,本文算法与基于矩阵行列变换的测试用例约简算法^[7]、基于关联模式的回归测试用例约简算法^[5] 进行比较分析。

(1)约简率比较. 通过分析,基于矩阵行列变换的算法约简率优于关联模式算法,本文所采用算法则表现的最好,分别高于其余两种算法,其中最高的提高了 19%,从而大大提高了软件测试效率,降低了软件测试的复杂度,达到了该算法设计的基本目标,最大化地精简了软件测试用例集。三种算法约简率如图 1 所示。

三种约简算法约简完毕之后,最简化用例集数量如表 2 所示。

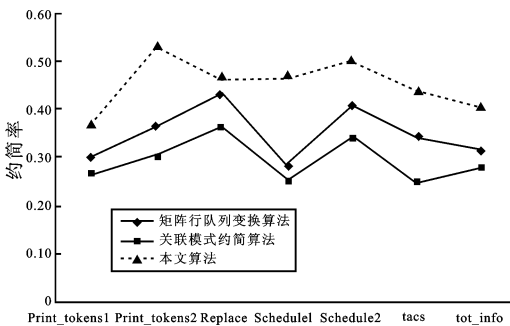


图 1 三种算法约简率比较分析图

表 2 三种算法约简后用例数

实验数据集	初始测试用例数	矩阵行列变换算法	关联模式约简算法	本文算法
Print_tokens1	30	21	22	19
Print_tokens2	30	19	21	14
Replace	30	17	19	16
Schedule1	32	23	24	17
Schedule2	32	19	21	16
tacs	32	21	24	18
tot_info	32	22	23	19

(2)错误检测率比较分析. 通过分析,与其他两种算法相比,本文算法约简之后能够检测出更多的错误数,提高了处理软件错误的效率,进而提高软件的可靠性与正确性,另一方面也能够提高测试的效率,如图 2 所示.

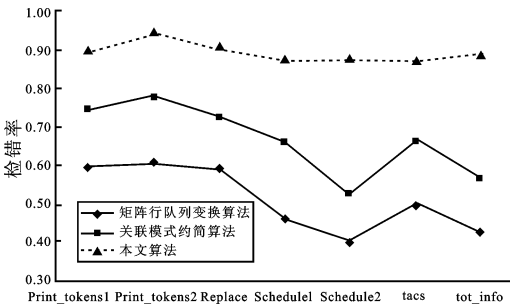


图 2 三种算法检错率

三种算法约简后检测出来的错误数如表 3 所示. 由表中可知,本文所采用的算法明显优于前两种算法.

表 3 三种算法约简后检测错误数

实验数据集	程序中的错误数	矩阵行列变换算法	关联模式约简算法	本文算法
Print_tokens1	20	12	15	18
Print_tokens2	18	11	14	17
Replace	22	13	16	20
Schedule1	24	11	16	21
Schedule2	32	13	17	28
tacs	24	12	16	21
tot_info	28	12	16	25

(3)错误检测丢失率比较分析. 错误检测丢失率表示用例约简之后遗漏的错误数量,错误检测丢失率越低越好. 三种算法的错误检测丢失率如图 3 所示. 图中显示本文算法的错误检测丢失率最低,在一定程度上体现了所约简的测试用例集覆盖度较高.

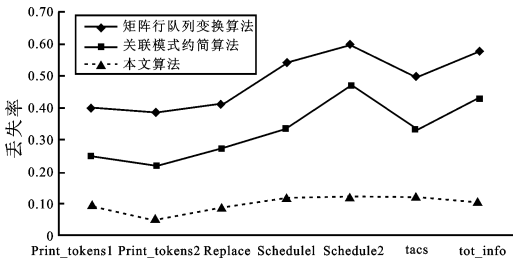


图 3 三种算法的有效性

5 结束语

本文基于模糊 K 均值算法提出了一种新的聚类算法,将其应用于软件测试用例集约简,可以在保持用例集的覆盖范围和错误检测有效性的基础上,实现用例集最简化,算法实验结果证明其具有较好的约简率和错误检测率,实现了通过约简后达到覆盖度高、规模精简的测试用例集的目的,并且体现出模糊聚类算法在聚类效果上表现出来的优势.

未来工作的重点包括两个关键内容:一是引入具体的模糊控制因子,以便更好地控制模糊 K 均值算法的输入参数,更加精准的控制算法的运行结果,实现软件测试用例集约简;二是针对算法测试集进行泛化,将算法引入到具有更多分支的程序中,检验并且改进算法的稳定性.

参考文献:

[1] Yu Y, Jones J A, Harrold M J. An empirical study of the effects of test-suite reduction on fault localization [C]// Software Engineering, ACM/IEEE 30th International Conference on. US: IEEE, 2008:201-210.

[2] Chen Z, Xu B, Zhang X, et al. A Novel Approach For Test Suite Reduction Based On Requirement Relation Contraction[C]// Proceedings of the Acm Symposium on Applied Computing Fortaleza, Geara, Brazil: IEEE, 2008:390-394.

[3] Zhang X, GuQ, Chen X, et al. A study of relative redundancy in test-suite reduction while retaining or improving fault-localization effectiveness. [C]// Proceedings of the 2010 ACM Symposium on Applied Computing. [s. l.]:ACM, 2010:2229-2236.

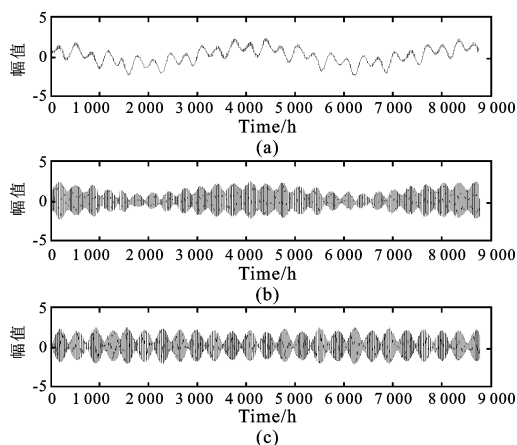


图4 经分类 PSO-ICA 处理之后信号波形图

根据杜森对重力固体潮信号中含有的主要潮汐波的角频率分析^[12],并由 $f = \frac{\omega}{(2\pi \times 3600)}$ 可知,本文改进算法处理所得分量信号频率具有与重力固体潮信号所含理论频率值相对应的特点,实现了将混合重力固体潮信号分解为独立信号分量的功能。

6 结束语

本文采用一种分类粒子的改进粒子群优化算法对独立分量分析的目标函数进行优化.改进粒子群优化算法将粒子进行分类,针对不同种类的粒子采用不同的速度及位置更新公式对粒子进行速度及位置的更新.同时,改进算法中加入的随机扰动速度使得粒子可以跳出局部最优,继续寻找最优粒子.这就使得改进粒子群优化算法在与 ICA 结合时可以跳出局部最优并提高算法的收敛速度.仿真实验结果表明,本文提出的改进粒子群优化算法与 Fast-ICA 及 PSO-ICA 相比具有分离精度高、计算速度快的特点,是一种分离盲源信号的有效方法.同时,由改进算法在重力固体潮信号中的应用研究分析可知,本文改进算法可以有效的将重力固体潮信号(混合信号)分解为含有与各个谐波成分理论值相一致的

独立信号分量.所以,本文改进算法具有一定的应用研究价值.

参考文献:

- [1] 郭武.面向图像处理的独立分量分析方法[D].长沙:国防科技大学,2008.
- [2] Common P. Independent component analysis, a new concept[J]. Signal Process, 1994,36(8):287-314.
- [3] 马建仓,牛亦龙,陈海洋.盲信号处理[M].北京:国防工业出版社,2006:1-7.
- [4] 李刚磊.基于改进粒子群的 ICA 算法[J].科技信息,2011,26(10):81-82.
- [5] Kennedy J, Eberhart R. Particle swarm optimization [C]// IEEE International Conference on Neural Networks. Washington: Bur of Labor Stat, IEEE, 1995: 1942-1948.
- [6] 张希文,郑茂.基于粒子群优化的独立分量分析算法研究[J].科学技术与工程,2010,10(8):1866-1873.
- [7] 席志红,边峦剑,晋野.基于改进粒子群的盲分离算法研究[J].应用科技,2010,37(1):12-22.
- [8] 孙路路.基于 ICA 的混合图像盲分离算法研究[D].南京:南京邮电大学,2010.
- [9] 李丽,牛奔.粒子群优化算法[M].北京:冶金工业出版社,2009.
- [10] 杨福生,洪波.独立分量分析的原理与应用[M].北京:清华大学出版社,2006.
- [11] Reju V G, Koh S N, Yann S I. Partial separation method for solving permutation problem in frequency domain blind source separation of speech signals [J]. Neurocomputing,2008,71(10/12):2098-2112.
- [12] 王谦身.重力学[M].北京:地震出版社,2003.

作者简介:

李巧燕 女,(1989-),硕士.研究方向为信号与信息处理、智能优化与决策. E-mail:lgy1373@qq.com.

全海燕 男,(1970-),博士,副教授.研究方向为信号与信息处理、智能优化与决策.

(上接第 136 页)

- [4] Hao D, Xie T, Zhang L, et al. Testinput reduction for result inspection to facilitate fault localization. [J]. Automated Software Engineering, 2010, 17(1):5-31.
- [5] 周冲波,楼俊钢,程龙.基于矩阵行列变换的测试用例约简算法[J].计算机应用研究,2013,30(3):779-782.
- [6] 苏小红,龚丹丹,王甜甜.结合用例约简与联合依赖概率建模的错误定位[J].软件学报,2014,25(7):1492-1504

- [7] 陈静,杨美红,王鲁.基于关联模式的回归测试用例约简模型[J].计算机工程,2011,37(2):63-65.

作者简介:

高丑光 男,(1989-),硕士研究生.研究方向为系统建模研究与仿真. E-mail:51932808@qq.com.

林都 男,(1957-),硕士生导师,教授.研究方向为系统建模与鲁棒控制.