

基于 DBSCAN 算法的测试用例优化方法^①

包晓安¹, 鲍超¹, 滕赛娜¹, 张唯¹, 张娜¹, 钱俊彦²

¹(浙江理工大学信息学院, 杭州 310018)

²(桂林电子科技大学广西可信软件重点实验室, 桂林 541004)

通讯作者: 张娜, E-mail: zhangna@zstu.edu.cn

摘要: 在软件测试研究领域, 测试用例约简一直以来都是研究的重点, 目前的一些研究利用测试需求之间复杂的相互关系得到约简的测试需求集, 在此基础上可以优化对应的测试用例集, 但单个测试需求所对应的测试用例集可能是一个密度分布且数量较大的集合. 对单个测试需求所对应的测试用例集合进行合理优化约简, 本文在这个方面做了深入的研究和探索, 提出了两种基于黑盒测试的类等价划分和类边界值分析策略. 基于 DBSCAN 算法提出了科学合理的参数取值方法, 提高了算法的适应问题程度和效率, 结合优化的算法和两种策略从而得到优化约简的测试用例集.

关键词: 黑盒测试; 约简; DBSCAN 算法; 参数取值

引用格式: 包晓安, 鲍超, 滕赛娜, 张唯, 张娜, 钱俊彦. 基于 DBSCAN 算法的测试用例优化方法. 计算机系统应用, 2018, 27(3): 105-111. <http://www.c-s-a.org.cn/1003-3254/6224.html>

Test Case Optimization Method Based on DBSCAN Algorithm

BAO Xiao-An¹, BAO Chao¹, TENG Sai-Na¹, ZHANG Wei¹, ZHANG Na¹, QIAN Jun-Yan²

¹(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 31008, China)

²(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In software testing field, test case reduction has been a research hotspot for a long time. Some researches currently use the complex relationship between test requirements for the test suites of test case, which can optimize the corresponding test suites on this basis. But the corresponding test case of a single test requirement may be a collection of density distribution in large quantities. This paper does an in-depth research and exploration on how to rationally optimize test case for the corresponding test suits of a single test requirement in the premise of test case. It proposes two classes based on black box testing equivalence partitioning and boundary value analysis strategy. Based on DBSCAN algorithm, it proposes a scientific and reasonable parameter selection method, and improves the adaptation degree and efficiency of algorithm. Combined with optimization algorithm and two strategies, it gets the optimal reduction set of test cases.

Key words: black box testing; reduction; DBSCAN algorithm; parameter choice

在测试工作过程中, 在满足测试需求的前提下应当尽可能的减少测试用例的数量同时确定测试用例的优先级^[1]顺序, 测试用例的数量和优先级跟测试的成本紧密相关, 如何优化测试用例一直是软件测试研究领域的重点. 在该研究领域约简方法和角度是多种多样

的, 并各有其特点. 例如陈军成、薛云志等人提出了一种基于事件处理函数的 GUI 测试用例集约简技术^[2]. 而在实际的测试工作当中我们可以将一个测试任务转化为具体的测试需求, 于是顾庆、唐宝等人提出了一种面向测试需求部分覆盖的测试用例约简技术^[3]. 测试

① 基金项目: 国家自然科学基金(61502430, 61379036, 61562015); 广西自然科学基金(2015GXNSFDA139038); 浙江理工大学521人才培养计划项目
收稿时间: 2017-05-25; 修改时间: 2017-06-16; 采用时间: 2017-06-19; csa 在线出版时间: 2018-02-09

需求之间除了部分覆盖的关系外还存在着复杂的重叠和包含关系,通过约简这些关系可以达到减小测试用例规模的目的,这就是徐宝文、聂长海等人提出的基于测试需求约简的测试用例约简技术^[4,5].约简过的测试需求集能在一定程度上缩减测试时测试的需求,这样就减少了需求对应的测试用例集的规模.测试需求约简模型只考虑了测试需求之间的冗余关系而并没有考虑到约简后的单个测试需求所对应的测试用例集合内部所存在的测试用例之间的冗余关系,这就是本文研究的重点.

关于用例约简的研究一直都是一个重点,1979年 VChvatal 提出了基于贪心算法^[6](Greedy, 简称 G)的测试用例约简方法. Harrold 等人提出了一种约简的启发式算法^[7](简称 H 算法),在 G 和 H 算法的基础上, Chen 和 Lan 提出了算法 GRE^[8,9],这些约简方法都有各自的特点和局限性,怎样去选择相应的约简方法还要根据具体需求情况. DBSCAN(基于密度的聚类算法)算法,它在很多领域得到了应用,例如郭世可、董槐林等人提出了一种结合密度聚类 and 区域生长的图像分割方法^[10].同时对于这一算法的研究和改进一直都在进行,于亚飞、周爱武提出了一种改进 DBSCAN 密度算法^[11],马帅、王腾蛟等人提出了一种基于参考点和密度的快速聚类算法^[12].当前关于该算法的研究虽然取得了一定的成果,但还是存在着很大的争议,尤其在参数取值这方面,科学合理的参数取值方法是提高该算法效率和适应度的关键.本文结合测试需求约简下测试用例集所存在的优化空间,在参数取值方面做了深入的研究,提出了两种有效的参数取值方法以及两种约简策略,结合改进的 DBSCAN 算法生成了更加精简的测试用例集.

1 研究背景

1.1 需求约简

徐宝文等人提出的需求约简模型确实起到了约简用例规模的效果同时消除了测试需求之间的复杂关系.本文的理论研究基础就是约简后的测试需求所对应的测试用例集合,目的就是为减少包含复杂关系的测试需求对优化结果的影响.单个测试需求所对应的用例集合中的测试用例存在属性和维度的上的相似性,这种相似性就是用例间冗余现象的基础,利用这种相似性我们可以完成用例覆盖替代即在相似性度较高的

测试用例间提取核心用例进行替代覆盖达到用例优化约简的效果.

如图 1 所示测试需求集 $R = \{R_A, R_B, R_C, R_D, R_E\}$, 经过约简得到约简后的测试需求集 $R = \{R_1, R_2, R_3\}$ 所对应的测试用例集 $T = \{T_1, T_2, T_3\}$ (T 是一个大集合, T_1, T_2, T_3 为子集). 在实际的测试过程中单个测试需求所对应的测试用例并不是单个的,在大规模的测试工程中是一个非常庞大的测试用例的集合,同时存在用例之间非常复杂的冗余现象,本文就是针对这一问题提出了一种解决方法.

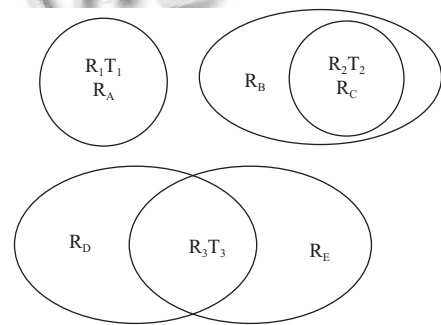


图 1 需求和测试用例集的关系

1.2 DBSCAN 算法思想

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 由 Ester 等人在 1996 年提出,是一个比较有代表性的基于密度的聚类算法.

DBSCAN 算法是通过利用密度连通性的原则去探索和发现不同形状类,这里的密度概念匹配于测试用例相似度概念,即密度值越高相似度越高.它的基本思路是这样:一个类中的每个数据对象在给定的半径 e 的领域中所包含的数据对象个数一定是不小于给定的某个数值,这个数值即领域密度阈值 ($MinPts$).为了去确定一个类,该算法首先从给定的数据集 D (测试用例集合) 中选择任意一个数据对象 O (测试用例),并搜索 D 中关于领域半径 e 和领域密度阈值 $MinPts$ 从 O 密度可以直接达到的所有数据对象.如果 O (例图中 A) 是核心数据对象 (核心测试用例),即在领域半径 e 中所包含的数据对象个数不少于 $MinPts$ 个测试用例,于是可以根据 DBSCAN 算法找到一个关于参数 e 和 $MinPts$ 的类.如果 O (例图中 a) 是一个边界点,则在以领域半径 e 的领域内所包含的数据对象个数少于 $MinPts$ 个,则将 O 被暂时标注为噪点在这里我们把它

定义为“流放用例”,然后 DBSCAN 继续处理 D 中的下一个未被处理的数据对象.图 2 为一个集合 D 中半径为 e , $MinPts$ 为 3 的 DBSCAN 算法模型.

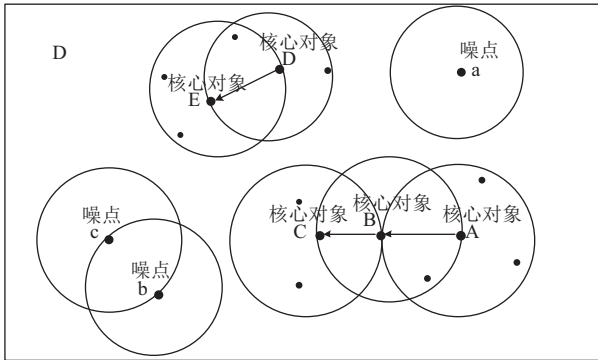


图 2 半径为 e , $MinPts=3$ 的 DBSCAN 算法模型

2 方法策略与算法改进

2.1 类黑盒测试策略

黑盒测试又称功能性测试是软件测试方法的重要组成部分,在软件测试中可以把程序看作是一个不能打开的黑盒子,并且不用考虑程序内部的情况和结构只在程序接口进行测试,它能反映程序是否能够按照需求规格说明书的要求正常使用,以及程序的输入能否按照正常的运行输出正确结果.黑盒测试的方法有很多种主要包括等价类划分法、边界值分析法、错误推测法、因果图法、判定表驱动法、正交试验设计法、功能图法、等等.本文针对等价类划分法以及边界值分析法提出了两种类似策略,能够很好的结合 DBSCAN 算法完成测试用例的优化约简效果.

2.1.1 类等价划分策略

等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的,并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数据.取得较好的测试结果.等价类是指某个输入域的子集合.在该子集合中,各个输入数据对于揭露程序中的错误都是等效的,并合理地假定:测试某等价类的代表值就等于对这一类其它值的测试.因此,可以把全部输入数据合理划分为若干等价类,在每一个等价类中取一个数据作为测试的输入条件,就可以用少量代表性的测试数

据.取得较好的测试结果.

本文中 DBSCAN 算法能够很好地聚类相似度极高的测试用例形成单个的类,这个类等价于等价划分中的等价类.本文提出一种类等价类划分策略即可选取该类中最具代表性的核心测试用例完成用少量代表性的测试数据完成测试取得测试结果的过程.

2.1.2 类边界值分析策略

大量的测试实践中证明许多错误是发生在输入和输出的范围边界内的,而不是在其内部,于是黑盒测试中边界值分析法就针对各种边界情况设计测试用例,进而发现更多的错误.由 DBSCAN 算法所生成的流放用例概念类似于边界值概念,它是游离于各个类边缘的数据,并不属于各个类介于类与类的之间或者集合的边界位置.本文提出一种基于边界值分析法的类边界值分析策略将流放用例作为测试的一个重点.

2.2 算法改进

该算法涉及两个非常重要的参数半径 e 和领域密度阈值 $MinPts$,这两个参数十分敏感,对聚类的效果以及测试用例集的优化起着关键的作用.在目前的研究当中大多数情况下都是凭借用户的经验而得到,或者进行逆向推导确定一个最优的值,但这样做的成本和时间复杂度相当的大.本文提出了两种效率较高的参数确定的方法,实验证明这两种方法所提供的参数值在改进后的算法中具有较高的问题适应度和工作效率,能充分的聚类相似度较高的用例获取核心用例以及流放用例.

2.2.1 领域半径

DBSCAN 算法本身就是基于密度的,该算法中的密度是一个基于一定范围内的概念.本文提出把每个测试需求所对应的测试用例集 T 看成是一个整体分布的大区间.例如 T_3 中的测试用例 t 由 (x, y) 确定,总体测试用例集 T_3 的范围是 $x \in (1, 1000)$, $y \in (10, 2000)$. T_3 这个大区间可以划分为许许多多的密集分布的子区间,例如可随机划分一个区间 T_a, T_a 即 T_3 中的一个子区间,范围为 $x \in (10, 20)$, $y \in (210, 220)$.在划分的无数的子区间中选取点分布数量较多较密集的几个区间作为本文确定半径 e 的采样区间.由于这些数据点之间具有相似性,即数据点越接近相似性越高,我们可以利用点与点之间的距离来表现这种相似性.计算每个采样区间内数据之间的距离标准差 (d_{sd}),选取最小的距离标准差 $d_{sd_{min}}$ 即相似性最高.其目的就是为了在更

大程度上使得到的类尽可能多的去覆盖分布相对密集即相似度较高的测试用例区域, 一个类囊括更多相似性较高的测试用例, 从而起到约简的效果降低测试的成本. 具体计算公式如下:

$$dsd = \sqrt{\frac{1}{n} \sum_{i=1}^N (\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} - \mu)^2} \quad (1)$$

其中, μ 为平均距离, N 为集合大小.

2.2.2 基于最小生成树的阈值 *MinPts*

通过确定半径 e 我们可以确定最终的采样区间 T_a . 子区间 T_a 中所有的点可以用线段连接起来, 我们可以把它看成是一个图 $G=(V, E)$ (如图 3) 其中 V 代表所有的点集, e 表示图 G 中点之间的关系 (边) 集合, 边的权值可以用 $d(u_i, u_{i+1})$ 表示即点 u_i 和 u_{i+1} 之间的距离, 具体的求解方法可以用欧式距离. 图 G 中边的权值确定后通过 Prim 算法或者 Kruskal 算法构造图 G 的最小生成树 *Tree*(图 3 粗线连接树), 具体算法的选择看图 G 边的稀疏程度, 边稠密的图用 Prim 算法, 边稀疏而点较多的图用 Kruskal 算法. 最小生成树 *Tree* 的权值和是唯一的且是最小的, 对于最小生成树中权值明显高于距离标准差值的进行断开 (图 3 中点 10 即为第一个断开点), 在子区间 T_a 内会形成许多最小生成树的分段树 T_s , 分段树中度值最大的数值 x 即为领域密度阈值 *MinPts*, 图 3 中 *MinPts* 值为 8(①②③④⑥⑦⑧⑨⑩). e 为距离标准差基本满足了分布相对密集点的间距, 实验表明在较密集的区间内以 O 为核心对象以 e 为半径覆盖点数数值基本满足了大于或者等于 x 的要求.

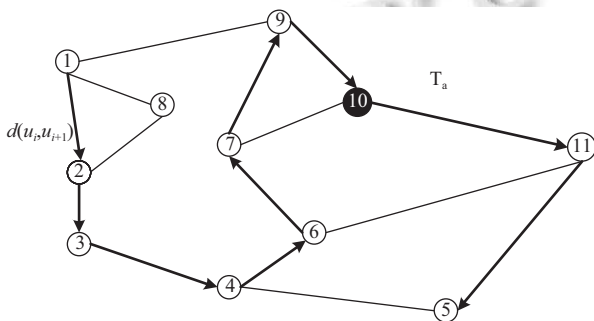


图 3 图 $G=(V, E)$ 和最小生成树 *Tree*

算法 1. *MinPts* 算法实现

```
1. Prim(V, int i){
```

```
2. T=∅ V; //初始化空树
3. U={u0}; //随机选取一个顶点
4. while((V-U)≠∅) { //若树中不含全部顶点
    (ui, ui+1) 使得 ui ∈ U 且 ui+1 ∈ (V-U) 且 d(ui, ui+1) 最小;
5. T=T ∪ [(ui, ui+1)]; //边归入树
6. U=U ∪ {ui+1}; //点归入树
}
7. Return U;
8. Return i; //返回 Tree 点集, 以及点集个数
}
9. MinPts(Ts, int i) {
10. Prim(U, int i);
11. Ts=U; j=i; //调用 Prim 算法的返回值
12. U={u0}; //任意选择 Tree 中的一个点
13. While((Ts-U)≠∅) {
14. If (d(ui, ui+1) >> dsd) {
15. Ts=Ts-(ui+1, uj); //如果某一边的权值远大于 dsd 进行断开
16. xi=Ts; xi 添加入 X //求分段树的度数并添加入集合 X
17. Ts=(ui+1, uj); U={ui+1}
} //继续搜索远大于 dsd 的权值
18. else
19. i++;
20. U=U ∪ ui;
21. 取 X 中的最大值 xi;
22. MinPts=xi; } //分段树中度数最大的数值即为领域密度阈值
```

MinPts 算法旨在解决领域密度阈值参数的取值问题, 本文提出的 *MinPts* 算法是在 Prim 算法的基础上实现的, 其根本的思想就是在最小的范围和成本内确定关键节点的数量问题, 并提高取值的科学性而不是根据经验取得, 保证最终算法的一个可靠和稳定, 从而解决了 *MinPts* 的取值问题. 本文提出的优化方法重点就是基于密集分布的集合进行约简优化, 所以在相对密集的集合中所取得 *MinPts* 值很具有代表性满足了解决问题的实质要求. 实验表明这一算法实现对实现优化约简的目的具有重要的作用和意义.

2.3 改进的 DBSCAN 算法

传统的 DBSCAN 算法^[13]作为一种有效的聚类算法, 主要作用是根据数据的属性进行分类分簇排除噪点的, 输出的对象主要是类, 从而达到聚类的效果. 本文对 DBSCAN 的改动主要是在算法中提取核心和噪点数据并消除输出对象类, 生成一个关于这些数据的集合, 同时利用新的参数取值方法提高算法的问题适应度, 从而达到本文的研究需求和提高解决问题的能力, 其输出的结果是包含核心测试用例和流放测试用例的数据集合. 具体改进算法实现框架如下.

算法 2. 改进的 DBSCAN 算法

输入: D : 测试需求 R_3 对应测试用例集合 T_3 所包含的 n 个数据对象;
 e : d_{sd} 值; $MinPts$: x_i ; T : 空集; N : 领域点集
 输出: T (核心数据对象、噪点集)

方法:

```

1. visited[D]=FALSE;
2. Do
3. visited[O]=TRUE, O∈D;
4. 在半径  $e$  内 If sizeof(NeighborPts[O])<MinPts;
5. T=T∪O
6. Create cluser C
7. Add O to C
8. For N 中每个数据对象 O
9.   If visited[O]=FALSE
10.    visited[O]=TRUE
11.    If O 的  $e$  半径范围内至少有  $MinPts$  个数据对象, 把这些
    数据对象添加入 N 中,
12.    T=T∪O
13.    If O ∉ VC
14.    C=C∪O
15.  End for
16.   Else mark O as noise
17.   T=T∪O
18. Until visited[D]=TRUE
  
```

测试用例集合我们可以看着是一个数据点集, 每个测试用例对应一个点, 点集越集中表示测试用例密度分布越密集即测试用例相似性越高, 改进的 DBSCAN 可以很好地将这些测试用例进行聚类划分, 从而确定核心用例和流放用例同时结合类等价划分和类边界值分析策略得到一个约简的测试用例集. 核心用例 O 在满足了测试需求的同时, 因具有高度的相似性可以替代类中其他非核心用例完成覆盖, 一个测试用例小集合(即一个类)中以核心测试用例作为代表进行软件测试, 即满足了测试需求同时也减少了测试用例的数量起到了优化测试用例集的作用. 而流放用例独立于任何类, 处于边界位置, 对于这类测试用例, 在通常的软件测试工作中也是测试的重点

3 实验仿真

3.1 实验设计

仿真实验的主要步骤如下: (1) 随机生成测试需求集 R , 在测试用例空间中生成测试用例集 T , 建立起测试需求和测试用例的满足关系. (2) 利用需求约简模型, 获得测试需求约简集 R' . (3) 基于 R' 利用 DBSCAN 获得约简后的测试用例集 DT . (4) 基于 R' 分别利用 $G, H,$

GRE 方法约简测试用例集获得约简测试用例集 $GT, HT, GRET(5)$ 测试约简集对比分析.

参数配置: $|R|$ 测试需求个数, 测试用例个数 $|T|$, 实验中的具体参数取值为 $|R|=30, |T|=3000$ 即初始测试需求集的个数为 30, 所对应的测试用例集中测试用例的个数为 3000. 嵌套阈值 $STRP$, $1-STRP$ 表示测试需求集 R 中包含关系的测试需求的生成概率, 即 $1-STRP$ 的数值越大生成包含关系测试需求的可能性就越大. $STRP$ 的取值根据文献[5]取值区间为 $[0.025, 0.900]$. 在实验的进行中随机生成一个实数 $\beta \in [0, 1]$, 如果 $\beta > STRP$ 则当前的测试需求 r 中生成一个被包含的 r , 否则在当前的 r 中停止生成被包含的 r . 由于 DBSCAN 算法跟密度分布有很大的关系所以在实验中设置一个密度参数 ρ, ρ 的取值范围为 $[0, 1]$, 当 ρ 的取值越接近 1 表示分布越密集, 越接近 0 表示分布越稀疏. 为了有效验证本文提出的理论, 实验中分别对 ρ 取三个值 0.1, 0.5, 0.9 即在密度稀疏, 密度中等, 密度密集的环境下进行实验.

3.2 实验分析

对于配置参数组 ($|R|, |T|, STRP, \rho$) 本文进行了 30 次的实验, 具体的实验平均结果如下表:

表 1 说明在区间内测试用例分布稀疏的环境下采用 DBSCAN 方法的效果并没有传统的方法 $G, H,$ GRE 有效果, 并不能达到优化的结果.

表 1 $\rho=0.1$, 各种方法获得的测试用例个数的比较

$STRP$	DT	GT	HT	$GRET$
0.025	8.0	5.3	5.3	5.3
0.050	12.2	10.4	10.4	10.4
0.075	14.0	13.0	13.0	13.0
0.100	17.3	15.9	15.9	15.9
0.125	18.0	16.5	16.5	16.5
0.150	17.8	16.8	16.8	16.8
0.175	19.5	18.5	18.5	18.5
0.200	20.4	18.9	18.9	18.9
0.250	19.0	18.2	18.2	18.2
0.300	19.5	17.7	17.6	17.6
0.400	18.3	15.8	15.6	15.6
0.500	15.0	12.2	12.0	12.0
0.600	16.0	11.3	11.0	11.0
0.700	11.2	9.0	8.9	8.9
0.800	10.3	7.5	7.2	7.2
0.900	9.5	7.2	6.8	6.8

表 2 在密度分布较为均匀的环境下, 方法 DBSCAN 和 G, H, GRE 方法的约简结果在不同的 $STRP$ 取值下各有优劣, 但总体上约简效果差异性并不大.

表2 $\rho=0.5$,各种方法获得的测试用例个数的比较

STRP	DT	GT	HT	GRET
0.025	5.2	5.3	5.3	5.3
0.050	10.2	10.4	10.4	10.4
0.075	12.9	13.0	13.0	13.0
0.100	15.3	15.9	15.9	15.9
0.125	16.5	16.5	16.5	16.5
0.150	16.8	16.8	16.8	16.8
0.175	19.0	18.5	18.5	18.5
0.200	18.4	18.9	18.9	18.9
0.250	18.0	18.2	18.2	18.2
0.300	17.5	17.7	17.6	17.6
0.400	15.8	15.8	15.6	15.6
0.500	12.0	12.2	12.0	12.0
0.600	11.3	11.3	11.0	11.0
0.700	9.2	9.0	8.9	8.9
0.800	7.3	7.5	7.2	7.2
0.900	7.2	7.2	6.8	6.8

表3是在分布密集的环境下进行的,可以看出无论STRP的取值如何DBSCAN算法在总体上都是优于传统的约简方法.可见在满足一定的条件下DBSCAN方法与传统方法相比还是具有一定的优越性.

表3 $\rho=0.9$,各种方法获得的测试用例个数的比较

STRP	DT	GT	HT	GRET
0.025	4.2	5.3	5.3	5.3
0.050	9.3	10.4	10.4	10.4
0.075	11.8	13.0	13.0	13.0
0.100	15.0	15.9	15.9	15.9
0.125	15.4	16.5	16.5	16.5
0.150	15.7	16.8	16.8	16.8
0.175	16.8	18.5	18.5	18.5
0.200	17.4	18.9	18.9	18.9
0.250	17.2	18.2	18.2	18.2
0.300	16.5	17.7	17.6	17.6
0.400	15.1	15.8	15.6	15.6
0.500	11.7	12.2	12.0	12.0
0.600	10.3	11.3	11.0	11.0
0.700	8.1	9.0	8.9	8.9
0.800	6.3	7.5	7.2	7.2
0.900	6.1	7.2	6.8	6.8

由于本文实验中所有涉及的优化方法均是在R的前提下,所以对于参数STRP只是起到一个辅助观察和研究的作用,具体的因为STRP的不同而体现出来传统约简方法之间的差异性本文不做赘述.上述表格总体上凸显出了DBSCAN方法的优劣性,在不同的环境下DBSCAN有不同的适用性,在测试用例集足够大且分布区间较密集的环境下DBSCAN方法具有一定的优越性.但在测试用例的分布较为稀疏的环境下该方法的适用性显然降低了很多,在何种情况下选择这种约

简方法还需要根据测试人员根据具体的测试工作环境进行选择.

图4表示在STRP分别取值0.025, 0.25, 0.9的情况下不同分布密度所对应的测试用例平均个数比较,横轴为分布密度取值,纵轴为用例个数取值.可以很清晰的看出随着 ρ 取值的递增,用例个数逐渐变小,充分显示了在高密度环境下DBSCAN约简方法的有效性.同时我们也发现STRP的取值的大小并不直接影响用例个数的多少,即在相同密度下用例个数与STRP不存在直接影响关系.

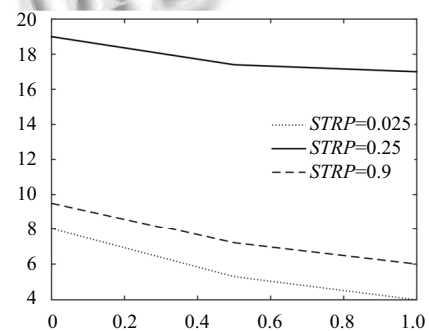


图4 不同阈值与密度下测试用例个数比较

综上所述,在一定的环境条件下DBSCAN用例约简方法比传统约简方法有效性更高,约简效果更佳,同时也表明DBSCAN约简法较适用于测试用例分布较为集中的测试用例集.

4 总结与展望

目前,软件测试用例约简方法大多数都是基于需求与测试用例的对应关系进行的,很少是基于测试用例之间的关系进行的.基于改进的DBSCAN算法的测试用例约简是在用例之间关系下提出的一种聚类约简方法,该方法具有一定的创新性,在一定环境下约简效果具有一定的优势.丰富了测试用例的约简方法,为今后的软件测试约简提供了一个新的思路.

基于改进的DBSCAN算法的约简方法需要在满足一定的条件下才能体现出其优越性,尤其是在测试用例集合根据一定属性分布比较密集的情况下才能体现出聚类的效果,从而达到更好的约简效果.如何在分布相对稀疏的环境下提高该方法的适应度是未来研究的一个重点和难点,同时对于参数取值方法的合理优化和改进也需要进行进一步的研究和探索.

参考文献

- 1 张娜, 姚澜, 包晓安, 等. 多目标优化的测试用例优先级在线调整策略. 软件学报, 2015, 26(10): 2451–2464. [doi: 10.13328/j.cnki.jos.004745]
- 2 陈军成, 薛云志, 陶秋铭, 等. 基于事件处理函数的 GUI 测试用例集约简技术. 软件学报, 2015, 26(8): 1871–1885. [doi: 10.13328/j.cnki.jos.004711]
- 3 顾庆, 唐宝, 陈道蓄. 一种面向测试需求部分覆盖的测试用例集约简技术. 计算机学报, 2011, 34(5): 879–888.
- 4 章晓芳, 陈林, 徐宝文, 等. 测试用例集约简问题研究及其进展. 计算机科学与探索, 2008, 2(3): 235–247.
- 5 章晓芳, 徐宝文, 聂长海, 等. 一种基于测试需求约简的测试用例集约化方法. 软件学报, 2007, 18(4): 821–831.
- 6 Chvatal V. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 1979, 4(3): 233–235. [doi: 10.1287/moor.4.3.233]
- 7 Harrold MJ, Gupta R, Soffa ML. A methodology for controlling the size of a test suite. *Proceedings of Conference on Software Maintenance*. San Diego, CA, USA, 1990. 302–310.
- 8 Chen TY, Lau MF. A new heuristic for test suite reduction. *Information and Software Technology*, 1998, 40(5-6): 347–354. [doi: 10.1016/S0950-5849(98)00050-0]
- 9 Chen TY, Lau MF. On the completeness of a test suite reduction strategy. *The Computer Journal*, 1999, 42(5): 430–440. [doi: 10.1093/comjnl/42.5.430]
- 10 郭世可, 董槐林, 龙飞, 等. 一种结合密度聚类 and 区域生长的图像分割方法. 计算机研究与发展, 2007, 44(S3): 420–423.
- 11 于亚飞, 周爱武. 一种改进的 DBSCAN 密度算法. 计算机技术与发展, 2011, 21(2): 30–33, 38.
- 12 马帅, 王腾蛟, 唐世渭, 等. 一种基于参考点和密度的快速聚类算法. 软件学报, 2003, 14(6): 1089–1095.
- 13 周水庚, 周傲英, 曹晶. 基于数据分区的 DBSCAN 算法. 计算机研究与发展, 2000, 37(10): 1153–1159.