高技术通讯 2012 年 第22 卷 第12 期:1219 ~ 1224

doi:10.3772/j.issn.1002-0470.2012.12.001

基于混沌免疫谱聚类的软件缺陷预测①

慕晓冬②* 常瑞花③** 宋国军* 宋洪军***

(*第二炮兵工程大学403教研室 西安 710025)

(**武警工程大学科研部 西安 710086)

(***第二炮兵青州士官学校计算机室 青州 262500)

摘 要 为提高无标识软件缺陷预测的准确性,提出一种谱聚类与混沌免疫相结合的软件缺陷预测方法。该方法首先将谱聚类算法引入到软件缺陷预测领域中,然后针对谱聚类算法中K-Means局部收敛的缺点,用一种混沌免疫聚类算法来替换K-Means算法。同时,在免疫克隆选择算法的框架下,借鉴混沌和免疫理论,设计免疫克隆聚类适应度函数计算方法,并给出分层混沌变异算子,以实现种群多样性的增加,促进无标识软件缺陷数据预测精度的提高。在Iris和3组商业软件模块数据集上进行了仿真实验,实验结果验证了该方法的有效性。

关键词 无标识数据,免疫,谱聚类,混沌,软件缺陷预测

0 引 言

随着计算机技术的发展以及软件所占比例的不断增大,复杂系统的可靠性越来越依赖于其软件的可靠性。软件缺陷是导致系统出错、失效和崩溃的潜在根源[1],因而软件测试是提高软件可靠性常用的手段之一。但研究表明,高达50%的软件测试资源用在了无缺陷模块上[2],造成极大浪费,因而采用准确的软件缺陷预测方法则显得更加必要。多年来,国内外学者一直从事这方面的研究,提出许多软件缺陷预测方法[3-5],并取得了不错的预测效果。但是传统研究主要集中在有监督的学习方法上,预测性能依赖于完整标记的训练样本,现实中存在大量标记不完整的训练样本(无标识数据),另外获得完整标记的样本需要化费大量的人力和财力。因此,如何利用无标识数据进行准确的软件缺陷预测成为当前一个亟待解决的问题。2004年Zhong等人[6]首次提出利用K-Means和Neural-Gas方法对无标识软件数据进行聚类,但该方法需要具有丰富机器学习和软件工程经验的人员对聚类模块进行类别标识,从而限制了其广泛应用。2007年,Seliya等人[7]以K-Means法为基聚类算法,提出了一种半监督学习方法,然而该方法也需要专家知识来递归标识聚类结果,这也增加了算法的运行时间,该方法在处理大数据集时聚类的数目和迭代次数也随之增加。为克服对专家经验的依赖,2009年Catal等人[8]提出了一种基于度量边界和X-Means的方法,可以看出,这些研究主要是利用K-Means或其扩展算法进行聚类预测,然而该类算法在凸样本空间分布上性能很好,当样本空间不为凸时,算法易于陷入"局部"最优。

近年来,出现了一种极具竞争力的谱聚类算法[9,10]。它基于谱图划分理论,可以在任意形状的样本空间上聚类,有效克服了K-Means聚类算法及其扩展算法的缺点。谱聚类已经成功应用于图像分割[11]、文本识别[12]等领域,然而在软件缺陷预测领域的研究还很少。鉴于此,为解决无标识软件数据的聚类预测问题,克服传统聚类算法的不足,本研究将谱聚类算法引入到软件缺陷预测中,并针对谱聚类算法K-Means处理无标识软件数据易于陷入局部最优的不足,提出了一种新的混沌免疫聚类方法,进而提出了基于混沌免疫谱聚类的软件缺陷预测方法,仿真实验及结果分析验证了该方法的有效性。

① 863计划(2010AA7010213),国家自然科学基金(61179005,61179004)和十一五国防预研(513270104)资助项目。

② 男,1965年生,教授,博士生导师;研究方向:软件缺陷预测,软件可靠性,计算机兵力生成与仿真;E-mail:blackeye_min@126.com

③ 通讯作者,E-mail:sxwcrh@163.com

(收稿日期:2011-12-06)

1 谱聚类算法及存在的问题

根据特征向量使用方法的不同,可以产生不同的谱聚类算法,经典的有 Shi 和 Malik 在 2000 年提出的 Normalized Cuts 算法[9]和 Ng 等人在 2002 年提出的 Ng-Jordan-Weiss (NJW) 算法[10]。它们的基本原理是类似的。本文主要考虑 NJW 算法。NJW 算法是一种流行的谱聚类算法,是一种简单而有效的多类聚类方法。NJW 算法利用拉普拉斯矩阵最大特征值所对应的特征向量,相应的相似性矩阵根据不同数据点间的距离度量来构造。

NJW 算法的步骤描述如下[10]:

步骤 1:计算矩阵 $\boldsymbol{L}_{\text{sym}}$ 的前 k 个最大特征值所对应的特征向量 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_k$ (必要时需作正交化处理),构造矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k]\in\boldsymbol{R}^{n\times k}$。

步骤 2:将矩阵 $\boldsymbol{X}$ 的行向量转变为单位向量,得到矩阵 $\boldsymbol{Y}$,即 $\boldsymbol{Y}_{ij}=\dfrac{\boldsymbol{X}_{ij}}{\sqrt{\sum_j \boldsymbol{X}_{ij}^2}}$。

步骤 3:将矩阵 $\boldsymbol{Y}$ 的每一行看作是 $\boldsymbol{R}^k$ 空间中的一个点,对其使用 K-Means 算法或任意其他经典算法,得到 k 个聚类。

步骤 4:将数据点 $\boldsymbol{y}_i$ 划分到聚类 j 中,当且仅当 $\boldsymbol{Y}$ 的第 i 行被划分到聚类 j 中。

由上述步骤可以看出,由于 NJW 算法采用 K-Means 算法完成最后的聚类,且是用迭代的方法寻找最优解,因而不能保证寻找到全局最优解[11]。为了克服其不足,本文对此展开研究,在谱聚类的框架下,提出了混沌免疫聚类算法。

2 混沌免疫聚类算法

免疫克隆选择算法[11,13]是一种模拟自然免疫系统的智能方法,它具有学习记忆功能,为软件度量数据的处理提供了新的思路。与进化算法不同的是,其抗体适应度值是根据抗原与抗体的亲和度、抗体之间的亲和度进行评价的,若某抗体与抗原之间的亲和度越大,且与其他抗体之间的亲和度越小,则该抗体的适应值就越大。这种适应值评价方式能保持个体的多样性,提高算法在局部解空间的搜索效率,并能有效摆脱局部最优点,但该适应度值的函数表达式难以确定,往往需要反复的试探[14]。文献[11]将免疫克隆选择算法与谱聚类相结合,很好地解决了图像分割问题,其亲和度函数采用了文献[15]的方法计算。本文在文献[11,15]基础上,在适应度函数计算中增加了聚类中心相似度的计算,使得聚类簇之间的相似度更小,从而提高了聚类的性能。

2.1 免疫聚类的适应度函数计算

假设 $\boldsymbol{T}=(t_{ij})_{n\times m}$ 为抗原群体,$\boldsymbol{W}=(w_{ji})_{k\times m}$ 为抗体群体,抗体新的适应度计算方法如式

$$R_m(\boldsymbol{C})=-\sum_{i=1}^{n}\Big(\sum_{j=1}^{k}D(\boldsymbol{t}_i-\boldsymbol{w}_j)^{1/(1-m)}\Big)^{1-m}-Sim(\boldsymbol{w}_{k1},\boldsymbol{w}_{k2}) \tag{1}$$

所示,其中,n 为抗体数目,$D(\boldsymbol{t}_i-\boldsymbol{w}_j)$ 为第 i 个抗体样本点 $\boldsymbol{t}_i$ 到第 j 个聚类中心 $\boldsymbol{w}_j$ 的欧式距离,$Sim(\boldsymbol{w}_{k1},\boldsymbol{w}_{k2})$ 表示第 k_1 个聚类中心 $\boldsymbol{w}_{k1}$ 和第 k_2 个 $\boldsymbol{w}_{k2}$ 之间的相似度。当 $Sim(\boldsymbol{w}_{k1},\boldsymbol{w}_{k2})$ 越小表示聚类中心间相似性越小,其中 $Sim(\boldsymbol{w}_{k1},\boldsymbol{w}_{k2})$ 的计算如下式所示:

$$Sim(\boldsymbol{w}_{k1},\boldsymbol{w}_{k2})=\frac{\sum_{k1,k2=1}^{m}\boldsymbol{w}_{k1}\boldsymbol{w}_{k2}}{\sum_{k1=1}^{m}\boldsymbol{w}_{k1}+\sum_{k2=1}^{m}\boldsymbol{w}_{k2}-\sum_{k1,k2=1}^{m}\boldsymbol{w}_{k1}\boldsymbol{w}_{k2}} \tag{2}$$

免疫克隆聚类的任务是优化 $R_m(\boldsymbol{C}^*)=\max_{v}R_m(\boldsymbol{C})$,当亲和度最大时,就寻找到了最优的聚类中心 $\boldsymbol{C}^*=[c_1,c_2,\cdots,c_k]$,$m\in[1,\infty]$,本文选取参数 $m=2$。

2.2 分层混沌变异算子

2.2.1 Logistic 混沌序列

混沌现象在自然界中普遍存在,研究表明免疫系统中也存在混沌现象[16,17]。混沌的随机性、遍历性和规律性等性质,可以有效地对其过程进行优化。在免疫算法中,变异是抗体进化的主要操作,为进一步提高免疫聚类算法搜索的性能,本文引入了混沌思想,给出了一种基于分层的混沌变异算子。

下面给出本文使用的典型混沌系统 Logistic 映射[16]:

$$x_{n+1}=\mu x_n(1-x_n),\quad n=0,1,2,\cdots \tag{3}$$

式中,μ 为控制参量。μ 值确定后,由任意初值 $x_0\in[0,1]$,可迭代出一个确定的时间序列 $x_1,x_2,x_3,\cdots$。随着 μ 值的增加,式(3)序列将呈现下述不同的性质[16]:

(1)当 $0<\mu\leqslant 1$ 时,系统的形态十分简单,除了不动点 0 外,没有其他周期点。

(2)当 $1<\mu<3$ 时,系统形态也比较简单,不动点 0,$1-1/\mu$ 为仅有的两个周期点。

(3)当 $3 \leqslant \mu \leqslant 4$ 时,系统的形态十分复杂,系统由倍周期通向混沌。

(4)当 $\mu > 4$ 时,系统更为复杂。

Logistic 序列进入混沌的过程如图 1 所示。

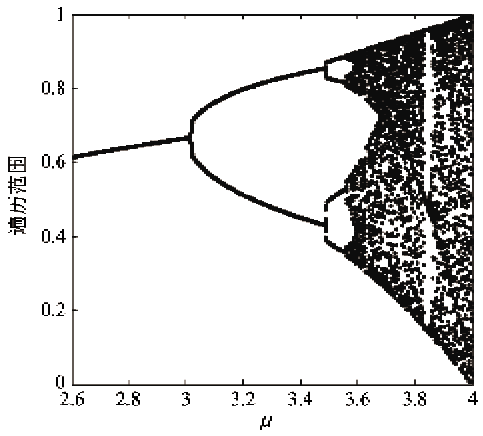

图 1 Logistic 序列进入混沌的过程

2.2.2 分层混沌变异算子

为了保存最优个体同时促进种群的多样性,提出了分层变异的策略。具体为:首先对抗体种群依据适应度函数值进行降序排列,然后将抗体种群分为高、中和低三层,对于不同的层次采用不同尺度的混沌变异概率。抗体所在层次越高,变异尺度越小,抗体所在层次越低,变异尺度越大,同时在进化初期采用较大的变异尺度进行搜索,在进化后期采用逐渐缩小的变异尺度,以提高求解的精度。

当变异概率 $p_m > 0.5$ 时,抗体变异计算如下式所示:

$$a'_{ij} = a_{ij} + \gamma_i \exp\left(-\frac{gen}{Gen}\right) \tag{4}$$

其中,γ_i 为分层控制参数,当 $i = 1,2,3$ 时,γ_i 依次取值为{0.1,0.3,0.6},分别表示高、中和低三层的权值;gen 为当前进化代数;Gen 为总进化代数;a_{ij} 为变异前第 i 个抗体第 j 个基因位对应的值。p_m 值由混沌序列式(3)产生。

3 混沌免疫谱聚类算法

限于篇幅,本文不单独给出混沌免疫聚类算法,而在谱聚类算法的框架下,将混沌免疫聚类算法嵌入其中,直接给出混沌免疫谱聚类(chaotic immune spectral clustering,CISC)算法,即:给定包含 n 个 d 维无标识软件数据集 $\boldsymbol{X} \in \boldsymbol{R}^{n\times d}$,首先通过谱聚类算法降维,得到新数据集 $\boldsymbol{Y} \in \boldsymbol{R}^{n\times k}$,再通过混沌免疫克隆聚类寻找该新数据集的最优聚类中心,最后将每一个数据划分到离它最近的聚类中心所在的类别中去,从而完成整个聚类过程。假设聚类数目为 k,具体步骤如下:

输入:无标识软件数据集 $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$;

输出:无标识软件数据的聚类预测结果。

步骤 1:计算相似矩阵 $\boldsymbol{W}$,$W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$,其中 σ 为事先确定的参数。

步骤 2:计算矩阵拉普拉斯矩阵 $\boldsymbol{L}$,$\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$。

步骤 3:计算 $\boldsymbol{L}$ 的 k 个最大特征值所对应的特征向量,并将其归一化得到矩阵 $\boldsymbol{Y}$。

步骤 4:将 $\boldsymbol{Y}$ 的每一行作为 $\boldsymbol{R}^k$ 空间中的一个点,将这些点作为抗原。

步骤 5:产生混沌的初始化种群。由于初始化取值越多,多样性就越丰富,聚类就越细致。混沌的遍历性能确保变量在[0,1]范围内不重复地遍历所有状态,因此,$gen = 0$ 时,利用式(2)所示的 Logistic 映射作为混沌吸引子,产生初始抗体群如下式所示:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1l} \\ a_{21} & a_{22} & \cdots & a_{2l} \\ \vdots & & \vdots & \\ a_{n1} & a_{n2} & \cdots & a_{nl} \end{bmatrix} \tag{5}$$

步骤 6:判断是否满足迭代条件:若满足 $gen \geqslant Gen$,则当前种群中的最佳抗体即为算法最终寻找到的聚类中心,退出,否则,$gen = gen + 1$ 并继续。

步骤 7:克隆,对当前的第 t 代父本种群中,具有最高亲和度的抗体进行克隆,得到 $\boldsymbol{A}'(t)$。

步骤 8:对 $\boldsymbol{A}'(t)$ 分层混沌变异得到 $\boldsymbol{A}''(t)$。

步骤 9:选择,选择具有最大适应度值的抗体作为新的聚类中心,转步骤 6。

步骤 10:计算 $\boldsymbol{Y}$ 中每一个点到最终得到的所有聚类中心的距离,将 $\boldsymbol{Y}$ 划分到具有最近距离的那一类聚类中心所在的类别中。

步骤 11:将原始的点根据 $\boldsymbol{Y}$ 的聚类结果划分到与之对应的相应的类别中去,具体对于初始的个体 x_i 当且仅当 $\boldsymbol{Y}$ 矩阵的第 i 行属于 j 类时,$\boldsymbol{x}_i$ 也被划分到 j 类中。

步骤 12:返回聚类结果。

4 仿真实验与结果分析

为了验证方法的可行性,分别使用来自 UCI[18]

和 PROMISE[19] 的数据集进行 2 组仿真实验，并对所有数据集的类别标签进行删除处理，作为算法处理的无标识数据。实验环境为：Pentium(R) 3.2G CPU，1GDDR 内存，Windows XP 操作系统的 PC 机，MatLab 7.4.0 (R2007a)。

算法参数设置为：克隆规模为抗体规模的 5 倍，最大迭代次数为 200 代。为了更准确地衡量算法的性能，给出了算法 20 次独立运行后的平均预测值。

4.1 实验 1

首先在 Iris 数据集上进行仿真实验 1。Iris 数据集包含 150 个样本，该数据集用萼片的长度、宽度和花瓣的长度、宽度来区分 3 种不同的花，每一种花占的比例均为 1/3，使用主成分分析(principal component analysis，PCA)方法将数据降为三维，给出 Iris 数据的分布结构图，如图 2 所示。

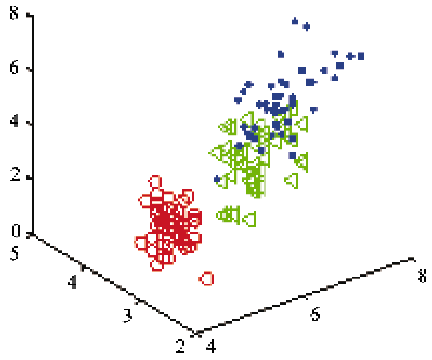

图 2 Iris 数据分布图

从 Iris 数据分布图可以看出，圆形表示的簇易于识别，另外两类有重叠不易于聚类识别。为了从整体上衡量算法的聚类效果，选择准确率(Accuracy)进行算法评价，定义 $Accuracy = N_1/N_2$，其中，N_1 为正确聚类的样本数，N_2 为全部样本数。

σ 为事先确定的参数，下面给出混沌免疫谱聚类(CISC)算法在不同 σ 下的多次运行结果，如图 3 所示。

由图 3 可以看出，CISC 算法当 σ 取值 2.8 时性能更佳。图 4 给出了当 $\sigma = 2.8$ 时，CISC 算法随着进化代数的增加时的聚类性能。从图 4 中可以看出，随着进化代数的增加，CISC 算法的预测准确值也在不断增加，在 75 代左右取得最优解并收敛。

为了进一步比较算法性能，选择经典 K-Means 算法、期望最大化(expectation - maximization，EM)算法及标准谱聚类算法 NJW 算法，在准确率评价指标下进行对比实验，结果如表 1 所示。

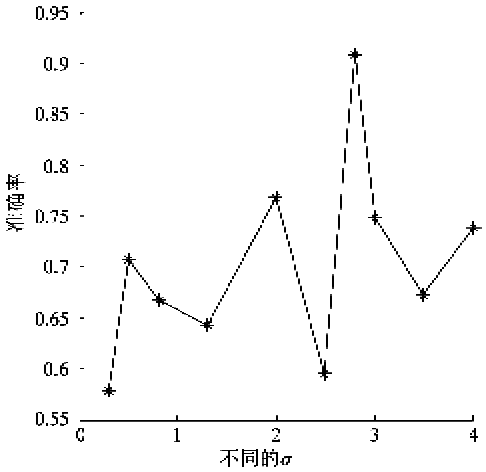

图 3 不同 σ 对应的结果

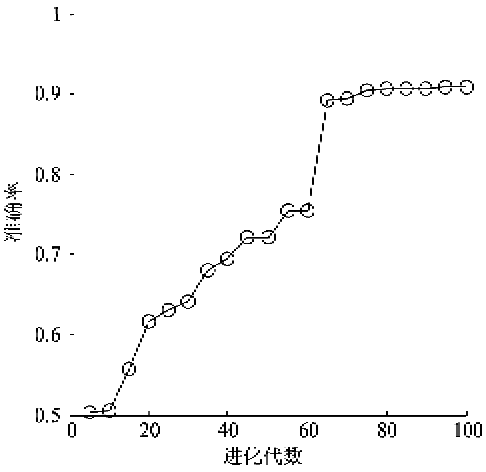

图 4 CISC 进化代数

表 1 不同算法之间的比较结果

Iris	CISC	K-Means	EM	NJW
准确率	0.9074	0.7867	0.7637	0.8208

从表 1 可以看出，CISC 算法优于 K-Means 算法和 EM 算法，取得了一个最高的预测值(0.9074)，表明 CISC 算法克服了传统 K-Means 算法和 EM 算法仅在紧凑的超球形分布的数据集合上有不错性能的不足。与标准谱聚类相比，算法聚类精度也得到了明显提高，主要的原因是混沌免疫谱聚类将进化搜索、全局搜索以及局部搜索相融合，以及分层混沌变异算子增加了抗体种群的多样性，使得算法不易陷入局部最优并以更加大的概率收敛到全局最优，取得很好的聚类效果。

4.2 实验 2

为了验证 CISC 算法在无标识软件数据缺陷预测的性能，利用 3 组商业软件模块（AR3，AR4 和

AR5)进行了进一步的对比实验。软件模块的基本特征如表2所示。同时在表3中,给出了本实验中用于描述软件模块特征的软件度量元。

表2 缺陷数据集

数据	模块	缺陷率	数据描述	语言
AR3	63	12.7%	电冰箱模块	C
AR4	107	18.69%	洗衣机模块	C
AR5	36	22.22%	洗碗机模块	C

表3 软件度量元

度量元	类型	度量元	类型
$V(g)$	McCabe	T	DHalstead
$EV(g)$	McCabe	B	DHalstead
$IV(g)$	McCabe	*UniqOp*	BHalstead
LOC	McCabe	*UniqOpnd*	BHalstead
N	DHalstead	*TotalOp*	BHalstead
V	DHalstead	*TotalOpnd*	BHalstead
L	DHalstead	*UniqOp*	BHalstead
D	DHalstead	*LOCcode*	Line Count
I	DHalstead	*LOCComment*	Line Count
E	DHalstead	*LOCBlank*	Line Count

在实验2中,选择了当前缺陷预测领域中处理无标识软件数据常用的K-Means算法和EM算法进行对比实验,同时为了检验CISC算法改进的有效性,标准谱聚类算法(NJW)也被选为对比算法,CISC算法和NJW算法中尺度参数σ均设为2.8。另外,由表1可以看出,这3组数据严重趋于不平衡性,因此选择软件缺陷领域常用的评价指标F测量(F-measure)进行算法评价[20,21]。具体的实验结果如表4所示。

表4 软件缺陷预测结果

	K-Means	EM	NJW	CISC
AR3	0.8730	0.6825	0.8570	0.8889
AR4	0.6636	0.8037	0.7248	0.8100
AR5	0.8611	0.8611	0.8713	0.8856

从仿真结果可以看出,上述3组数据集在F测量(F-measure)的评价指标下,混沌免疫谱聚类(CISC)方法的性能均优于传统软件缺陷预领域中处理无标识数据的算法,这表明了CISC算法保持了传统谱聚类算法处理非凸数据分布的优势,克服了传统聚类算法局部最优的不足;同时与标准谱聚类算法NJW相比,CISC方法的F-measure值也得到了明显的提高,尤其对于AR4数据集。这主要得益于混沌免疫聚类的抗体适应度评价方法,它提高了算法在局部解空间的搜索效率,分层混沌变异因子促进了个体的多样性,从而有效摆脱了局部最优点,使得预测结果得到了有效的提高,这也说明了在谱聚类算法的框架下结合混沌免疫聚类是可行的。

5 结 论

本文提出了一种基于混沌免疫谱聚类的无标识软件缺陷预测方法,它利用谱聚类算法克服传统聚类方法局部收敛的不足,并将混沌因子和免疫克隆选择算法引入谱聚类算法,设计了混沌克隆聚类适应度函数计算方法和分层混沌变异算子,以进一步提高谱聚类算法对无标识缺陷数据聚类预测的性能。两组仿真实验的结果表明该算法的性能优于传统聚类算法,从而为解决无标识软件缺陷数据的预测问题提供了新的思路。下一步将展开CISC算法处理大规模无标识数据的性能以及CISC算法中参数优化设置的研究。

参考文献

[1] 王青,伍书剑,李明树. 软件缺陷预测技术. 软件学报, 2008,19(7):1565-1580

[2] Shull F, Basili V, Boehm B, et al. What we have learned about fighting defects. In: Proceedings of the 8th IEEE Symposium on Software Metrics, Washington D C, USA, 2002. 249-258

[3] Gondra I. Applying machine learning to software fault-proneness prediction. *Journal of Systems and Software*, 2008, 81 (2):186-195

[4] Menzies T, Greenwald J, Frank A. Data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering*, 2007, 33 (1):2-13

[5] André B C, Aurora P, Silvia R V. A symbolic fault-prediction model based on multiobjective particle swarm optimization. *Journal of Systems and Software*, 2010,83:868-882

[6] Zhong S, Khoshgoftaar T, Seliya N. Unsupervised learning for expert-based software quality estimation. In: Proceeding of the 8th International Symposium on High Assurance Systems Engineering, Washington D C, USA, 2004. 149-155

[7] Seliya N, Khoshgoftaar T M. Software quality analysis of unlabeled program modules with semisupervised clustering. *IEEE Transactions on Systems, Man, and Cybernet-*

ics, *Part A*: *Systems and Humans*, 2007, 37(2): 201-211

[8] Catal C, Sevim U, Diri B. Clustering and Metrics Thresholds based Software Fault Prediction of Unlabeled Program Modules. In: Proceedings of the 6th International Conference on Information Technology: New Generations, Software Engineering Track, Washington D C, USA, 2009. 199-204

[9] Shi J B, Malik J. Normalized cuts and image segmentation. *IEEE Trans On Pattern Analysis and Machine Intelligence*, 2000, 22(8):888-905

[10] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm. In: Proceeding of Advance of Neural Information Processing System, Cambridge, USA, 2002. 849-856

[11] 张向荣, 骞晓雪, 焦李成. 基于免疫谱聚类的图像分割. 软件学报, 2010, 21(9):2196-2205

[12] 吴锐,黄剑华,唐降龙等. 基于灰度直方图和谱聚类的文本图像二值化方法. 电子与信息学报, 2009, 31(10):2460-2464

[13] 焦李成,杜海峰. 人工免疫系统进展与展望. 电子学报, 2003, 31(9):73-80

[14] 张雷,李人厚. 人工免疫 C-均值聚类算法. 西安交通大学学报, 2005, 39(8):836-839

[15] Hall L O, Ozyurt B, Bezdek J C. Clustering with genetically optimized approach. *IEEE Trans On Evolutionary Computation*, 1999, 3(2):103-112

[16] 杜海峰,公茂果,刘若辰等. 自适应混沌克隆进化规划算法. 中国科学, 2005, 35(8):87-829

[17] Du H F, Jiao L C, Wang S N. Clonal operator and antibody clone algorithms. In: Proceedings of the 1st International Conference on Machine Learning and Cybernetics, Washington D C, USA, 2001. 506-510

[18] Asuncion A, Newman D. UCI machine repository. http://archive.ics.uci.edu/ml, 2012

[19] Sayyad S J, Menzies T J. The PROMISE repository of empirical software engineering databases, University of Ottawa. http://promise.site.Uottawa.ca/SERepository, 2012

[20] Zhang H Y, Zhang X Z. Comments on "data mining static code attributes to learn defect predictors". *IEEE Transactions on Software Engineering*, 2007, 33(9): 35-637

[21] 常瑞花,慕晓冬,李琳琳等. 基于模糊聚类非负矩阵分解的软件缺陷预测. 宇航学报, 2011, 32(9):2059-2064

Software defect prediction based on chaotic immune spectral clustering

Mu Xiaodong*, Chang Ruihua**, Song Guojun*, Song Hongjun***

(* Staff Room of 403, The Second Artillery Engineering University, Xi'an 710025)

(** Scientific Research Department, The Engineering University of Armed Police Force, Xi'an 710086)

(*** Staff Room of Computer, The Second Artillery Petty Officer College, Qingzhou 262500)

Abstract

To improve the accuracy of defect prediction for unlabeled software data sets, a novel software defect prediction method based on the combination of spectral clustering and chaotic immune is presented. The method first introduces the Ng-Jordan-Weiss (NJW) algorithm, a spectral clustering algorithm, into the field of software defect prediction, and then uses a new chaotic immune clustering algorithm go replace the K-Means algorithm to overcome the K-Means's problem of easily getting trap local optima in spectral clustering. And under the framework of immune clone selection, it designs a new affinity function for immune clone clustering and gives the layered chaotic mutation operator based on the immune and chaotic theory to diversify the antibodies and improve the accuracy of software defect prediction. Two case studies are used to validate the method on the Iris and three commercial software data sets. The experimental results illustrate the effectiveness of the proposed method.

Key words: unlabeled data, immune, spectral clustering, chaos, software defect prediction