

一种改进的实值负选择算法

刘锦伟^{1,2}, 唐 俊^{2,3}

(1. 湖南科技大学计算机学院, 湖南 湘潭 411201; 2. 湖南城建职业技术学院信息工程系, 湖南 湘潭 411101;
3. 同济大学软件学院, 上海 200092)

摘 要: 通过分析已有实值负选择算法检测率不高的原因, 提出一种通过鉴别边界自体样本的改进负选择算法, 以提高对检测黑洞的覆盖率。给出算法的改进思想、具体实现过程及优势分析。采用人工合成数据集 2DSyntheticData 和实际 Biomedical 数据集对算法进行验证。实验结果表明, 该算法检测率较高, 所需的检测器数量较少, 综合性能较优。

关键词: 人工免疫系统; 负选择算法; 异常检测; 实值; 数据集

Improved Real-value Negative Selection Algorithm

LIU Jin-wei^{1,2}, TANG Jun^{2,3}

(1. College of Computer, Hunan University of Science and Technology, Xiangtan 411201, China; 2. Department of Information Engineering, Hunan Urban Construction College, Xiangtan 411101, China; 3. School of Software, Tongji University, Shanghai 200092, China)

【Abstract】 By analyzing the reasons for the low detection rate of the existing real-value negative selection algorithms, an improved negative selection algorithm is proposed with the identification of boundary samples to improve the coverage of holes. Detailed realization and advantages of the algorithm are given. The experiments of synthetic 2DSyntheticData and real biomedical data sets are done to test the algorithm. Experimental results show that the algorithm has higher detection rate and needs less detector numbers. It has optimum overall performance.

【Key words】 artificial immune system; negative selection algorithm; anomaly detection; real-value; data set

DOI: 10.3969/j.issn.1000-3428.2011.14.065

1 概述

负选择算法是人工免疫系统的主要算法之一, 已经在不同的领域得到了广泛的应用^[1]。任何基于负选择算法的异常检测都希望得到较高的检测率。已有研究表明^[2-5], 负选择算法检测率不高的原因主要在于存在检测黑洞。黑洞是无法根据匹配规则和自体集合被检测器集合匹配的非自体样本, 主要发生在自体和非自体的边界处。由此可见, 为了提高负选择算法的检测率, 必须减少检测黑洞, 而黑洞的存在主要在于检测器的表示和匹配机制。

检测器的表示方法有二进制表示和实值表示。已有研究者采用二进制表示方法对此问题进行了研究。由于实值表示更适合实际问题的描述, 因此得到了更多的关注。文献[3]提出了实值表示的负选择算法, 采用定长检测器; 文献[4]提出了变长的检测器生成算法, 提高了检测率, 并且在不同的异常检测领域得到广泛应用。

本文基于实数编码的可变长检测器, 提出一种改进的负选择算法, 主要处理边界处的自体样本。

2 实值负选择算法改进思想

在已有的负选择算法中^[3-5], 只有既不被自体半径覆盖又不被所有检测器半径覆盖的样本点才有可能成为候选检测器。也就是说, 随机产生的一个点成为检测器必须经过与自体集合和检测器的二阶段负选择。

这种机制将在自体区域与非自体区域的边界处带来一些无法被检测器覆盖的黑洞, 如图 1 所示。其中, 结果使用交叉形状的自体区域, 自体半径为 0.1, 期望覆盖率为 99%^[4]。黑框表示该区域存在黑洞, 4 个小区表示自体区域, 圆环表示已经存在的检测器集合。从图 1 可以看出, 很多非自体无

法被检测器集合有效覆盖, 产生了覆盖黑洞。

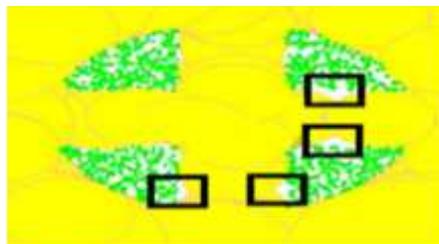


图 1 黑洞检测示意图

此外, 从图 1 也可以看出, 所有的黑洞都存在于边界自体的自体半径范围之内。因此, 本文采用了一种更细致的方法处理检测器: 把检测器由(检测中心, 检测器半径)的二维表示改变为(检测中心, 检测器半径, 最近自体点)的三维表示。实际上, 最近的自体点就是边界自体。因此, 改进的算法将包含所有已检测到的边界自体。在改进的算法中, 新的样本点只要满足以下情况就可以成为候选检测器: 样本点没有被已有的检测器集合覆盖, 或者样本点检测到了新的边界自体。

3 本文算法基本步骤和分析

3.1 检测器生成算法基本步骤

本文检测器生成算法的基本步骤如下:

步骤 1 产生候选检测器 x 。

基金项目: 湖南省教育厅科研基金资助项目(08D030, 10C0082)

作者简介: 刘锦伟(1979—), 女, 讲师, 主研方向: 计算智能, 网络与信息安全; 唐 俊, 讲师

收稿日期: 2010-12-21 **E-mail:** 954218364@qq.com

步骤2 计算 x 与检测器集合中已有的检测器 $d_i(i=1,2,\dots,n)$ 的欧氏距离 $L_{d_i} = \text{Euclidean}(d_i, x)$ 。

步骤3 如果距离 L_{d_i} 小于检测器 d_i 的半径 $r(d_i)$ ，说明点 x 已经被检测器覆盖，则计数器 $t = t + 1$ ；否则，转步骤5。

步骤4 如果 $t \geq 1/(1-a)$ ，说明覆盖率已经足够，结束算法；否则，转步骤1。

步骤5 计算点 x 与自体样本集合 S 中自体点 s_i 的欧氏距离 $L_i = \text{Euclidean}(s_i, x)$ ， x 的半径 r 由最近的自体元素决定，同时，记录最近的自体元素 s_{near} （即边界点），其与 x 的距离记为 L_{near} ，记检测器为 (x, r, s_{near}) ，其中， $r = L_{\text{near}} - r_s$ 。

步骤6 将 (x, r, s_{near}) 作为新的检测器加入检测器集合 D 。

步骤7 如果检测器集合 $|D|$ 达到最大检测器数量 N_{max} ，则结束；否则，转步骤1。

S 表示自体样本集合； N_{max} 表示预设的最大检测器数量； r_s 表示自体半径； a 为所期望的覆盖率； D 表示生成的检测器集， $d_i(i=1,2,L,n)$ 为检测器集合中的一个检测器； t 表示一个随机样本点被检测器覆盖的次数。

3.2 检测器对新样本的检测过程

在检测阶段，基本步骤如下：

步骤1 输入新的样本点。

步骤2 如果样本点与检测器集合 D 中的任何检测器都不匹配，则认为此样本点为自体；否则，转步骤3。

步骤3 如果样本点匹配其中任何一个检测器 $d_i(i=1,2,L,n)$ ，则判断是否属于边界元素，如果是，则认为是自体，否则，认为是非自体，转步骤4。

步骤4 算法结束。

3.3 算法特点及优势分析

本文算法特点如下：

(1)本算法与已有算法的最大不同之处在于^[3-5]：在生成检测器的过程中增加了记录边界自体的过程，在检测过程中增加了对边界自体的检测。

(2)与定长检测器相比^[3]，本算法中每个检测器的半径可变，覆盖检测黑洞的能力更强。

(3)本检测器生成采用集成了点估计的覆盖率估计方法作为算法的终止条件，而不是简单决定于预设的检测器最大数量。因此，本算法提供了以下2种结束方式：1)当估计的覆盖率达到时，算法结束，这是本算法独特之处，可以避免产生冗余的检测器。2)检测器的数目达到预设值。即使在这种情况下，因为检测器是变长的，所以算法仍然有较好覆盖黑洞的能力。

采用点估计(检测器生成算法中步骤4)的正确性证明如下：

(1)生成随机点作为候选检测器。

(2)如果它是一个非自体点并且未被覆盖，在它上面产生一个新的检测器；如果它被覆盖的自体点，则不将其作为候选检测器，但尝试的过程被记录在一个计数器 t 中，用于估计覆盖率。

(3)如果在非己空间中采样 m 个点，只有一个未被覆盖的点，通过点估计，未覆盖的区域部分为： $1/m$ ，则覆盖率估计值为 $a = 1 - 1/m$ 。

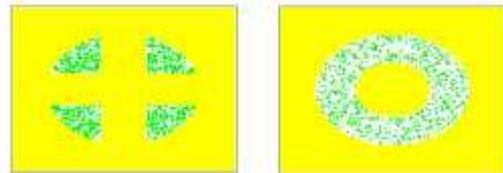
(4)当随机尝试 t 次但没有发现一个未覆盖的点(点全部被覆盖)时，估计实际的覆盖率已达到 a 。因此， t 无需预设，它由目标覆盖率 $t = 1/(1-a)$ 决定。

4 实验和结果分析

为了验证算法的性能，在合成数据和真实数据上均进行了实验验证^[6-7]，并与相关算法做了比较。

4.1 合成数据上的实验结果

先用合成的2维数据集验证本算法^[6]，用Java语言编程实现。整个空间是2维区域 $[0,1]^2$ 。此数据集包括了多种不同形状的自体分布。为了便于与V-detector算法比较，本文使用环形和交叉型自体分布，如图2所示。



(a)交叉型自体

(b)环形自体

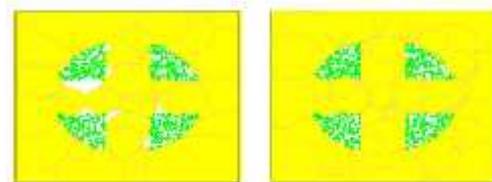
图2 自体分布示意图

假设训练数据点在整个空间以特定形状随机分布，使用1000个自体点。检测器产生后，用随机分布在整个空间的点评估性能。测试数据是1000个随机分布的数据点，包括自体与非自体。为便于与已有文献比较^[3-5]，实验中参数的选择如下：最大检测器数量为 $N_{\text{max}} = 1000$ ， $r_s = 0.1$ ， $a = 99\%$ 。实验结果是使用同样参数运行100次的平均结果。检测性能用检测率和虚警率衡量。对比算法为经典的V-Detector算法^[4]。如表1所示，本文算法检测率远高于V-Detector算法，平均达到92%，说明本文算法在检测黑洞的覆盖上是有效的。同时，本文算法在虚警率方面略有上升，但需要的检测器数量大幅下降。因此，本文算法总体上有一定的优越性。

表1 不同自体形状下算法检测性能比较

自体形状	算法	检测率/(%)	检测器数
交叉	本文算法	99.36	172
	V-Detector	97.61	510
环形	本文算法	98.29	181
	V-Detector	95.92	528

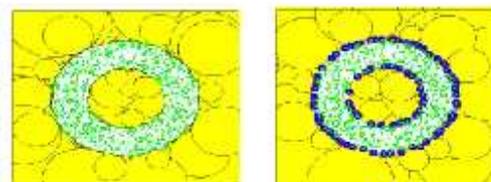
图3、图4是程序运行结果，主要是对2种算法对黑洞覆盖和所需检测器数量及边界自体元素的直观显示。



(a)V-Detector 算法

(b)本文算法

图3 黑洞覆盖示意图



(a)检测器

(b)边界自体

图4 边界自体元素示意图

从图3可以看出，本文算法有较好覆盖黑洞的能力，并且所需的检测器数量有所减少。在图4中，圆圈中黑点表示边界自体样本。

4.2 真实数据上的实验结果

为了验证本文算法的性能，在Biomedical数据集上进一步

(下转第199页)