

# 一种基于受体编辑的实值阴性选择算法

李贵洋 郭涛

(可视化计算与虚拟现实四川省重点实验室 成都 610066) (四川师范大学计算机科学学院 成都 610101)

**摘要** 受生物免疫受体编辑理论的启发,提出了一种基于受体编辑的实值阴性选择算法 RERNS(Receptor Editing-inspired Real Negative Selection Algorithm)。对于匹配自体的检测器,该算法采用定向受体编辑使之获得新生,而这些新生的检测器分布在自体与非自体的边界区域,从而增加了检测器的多样性,并改善了算法对边界区域的覆盖情况;对于不匹配自体的检测器,该算法采用识别相同最近自体的定向受体编辑,使检测器在包含原检测范围的情况下扩大了对非自体空间的覆盖。理论分析和实验验证表明,与实值阴性选择算法中具有代表性的 RNS 算法和 V-detector 算法相比,RERNS 算法生成的未成熟检测器更少,且检测性能更好。

**关键词** 人工免疫系统,阴性选择算法,受体编辑

**中图分类号** TP18 **文献标识码** A

## Receptor Editing-inspired Real Negative Selection Algorithm

LI Gui-yang GUO Tao

(Visual Computing and Virtual Reality Key Laboratory of Sichuan Province, Chengdu 610066, China)

(College of Computer Science, Sichuan Normal University, Chengdu 610101, China)

**Abstract** Inspired by theory of biological immune receptor editing, a receptor editing-inspired real negative selection algorithm(RERNS) was proposed. For the detector that matches self, algorithm uses directional receptor editing to make a new life. These new detectors are located in the area of self and non-self boundary, thereby the diversity of detector is increased and the boundary covered by the algorithm is also improved. For the detector that does not match self, algorithm uses direction receptor editing for identifying identical nearest self to expand coverage of no-self space under the circumstances of containing original scope of detector. Theoretical analysis and experimental verification show that RERNS algorithm generates less un-mature detectors and obtains better detection performance than the most representative RNS algorithm and V-detector algorithm.

**Keywords** Artificial immune system, Negative selection algorithm, Receptor editing

## 1 引言

由美国学者 Forrest 等<sup>[1]</sup>首先提出的阴性选择算法是人工免疫系统领域最重要的算法之一,常用于故障检测、病毒检测和网络安全检测等异常检测领域<sup>[1-4]</sup>。该算法的思想来源于传统免疫理论对 T 淋巴细胞胸腺耐受现象的解释。传统免疫理论认为:T 淋巴细胞的特异性抗原受体一旦产生就不会再发生变化,在胸腺耐受中,识别自身抗原的 T 淋巴细胞将凋亡或失活,而未识别自身抗原的 T 淋巴细胞经过一段耐受期后将成熟。基于此理论建立的经典阴性选择算法框架<sup>[1]</sup>中,模拟 T 淋巴细胞的检测器一旦产生就不再变化,其中匹配自体的检测器将被删除,而没有匹配自体的检测器则成为成熟检测器。

后续研究中,相继出现了位串阴性选择算法和实值阴性选择算法,在这两类算法研究中,如何扩大检测器对非自体空间的覆盖并减少“漏洞”的存在,始终是一个重要的研究方向。

在位串阴性选择算法的研究中,Balthrop 等<sup>[5]</sup>指出了 r-连续位匹配存在的“漏洞”问题,并提出了改进的 r-chunk 匹配机制;张衡等<sup>[6]</sup>提出了 r-可变阴性选择算法;何申等<sup>[7]</sup>提出了检测器长度可变阴性选择算法;在实值阴性选择算法的研究中,Gonzalez 和 Dasgupta 等<sup>[8]</sup>提出了检测器位置可演化的 RNS 算法;Zhou 和 Dasgupta 等<sup>[9-12]</sup>则提出了检测半径可变的 V-detector 算法。尽管这些改进的算法不同程度上减少了“漏洞”的存在,扩大了检测器对非自体空间的覆盖,并提高了算法的检测性能,但本质上这些算法的思想仍没有脱离传统免疫理论和经典阴性选择算法框架的束缚。

生物免疫系统的机制十分神秘而复杂,如何从生物免疫学的相关理论中提取灵感,从而改进甚至是构建新的免疫算法是目前人工免疫系统理论和应用研究的重要课题<sup>[4]</sup>。受体编辑理论是近年兴起的用于解释淋巴细胞耐受行为的生物免疫理论<sup>[13,14]</sup>。与传统免疫理论认为 T 淋巴细胞一旦产生就不再发生变化不同,受体编辑理论认为 T 淋巴细胞个体具有

到稿日期:2011-10-15 返修日期:2012-01-01 本文受四川省科技厅重点实验室项目(PJ201102)资助。

李贵洋(1975-),男,博士,副教授,主要研究方向为人工免疫与信息安全,E-mail:guiyang.li@gmail.com;郭涛(1967-),女,硕士,副教授,主要研究方向为数据挖掘与信息可视化。

一定程度的自适应能力,在胸腺耐受中,T淋巴细胞的抗原受体特异性在一定条件下会向其他特异性漂移或发生亲和力的变化,此抗原受体特异性变化被称为受体编辑。通过受体编辑,机体一方面可以给那些识别自身抗原的淋巴细胞一个“改过自新”的机会,使它们从针对自我转化为针对非我,以进一步丰富抗原受体的多样性;另一方面也可以给不识别自身抗原的淋巴细胞一个“改良”的机会<sup>[13,14]</sup>。

受受体编辑理论的启发,本文提出了一种基于受体编辑的实值阴性选择算法 RERNS(Receptor Editing-inspired Real Negative Selection Algorithm),该算法赋予了检测器一定的自适应能力,其中匹配自体的无效检测器通过受体编辑可成为不再匹配自体的有效检测器(改过自新),而没有匹配自体的检测器则通过受体编辑能进一步扩大对非自体空间的覆盖(改良)。

## 2 基于受体编辑的实值阴性选择算法

### 2.1 基本概念

**定义 1(系统状态空间)** 实值表达的特征值通过标准化映射到区间 $[0, 0, 1, 0]$ ,则  $n$  维系统状态空间中的任意一状态可表示为特征向量  $x^i = \{x_1^i, x_2^i, \dots, x_n^i\} | x_j^i \in [0, 0, 1, 0], j=1, \dots, n$ ,而  $n$  维系统状态空间  $U$  可表示为超立方体  $[0, 0, 1, 0]^n$ 。系统状态空间进一步由自体空间  $Self$  与非自体空间  $Nonsel$  两个互补的子空间构成,即  $Self \subset U, Nonsel = U - Self$ 。

**定义 2(距离度量)** 距离度量用于计算状态空间中任意两点之间的距离。在本文提出的 RERNS 算法中,采用闵可夫斯基公式度量距离。对于  $n$  维状态空间中的任意两点  $x \in U, y \in U$ ,其  $m$ -阶闵可夫斯基距离函数  $f$  为:

$$f(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^m \right)^{\frac{1}{m}} \quad (1)$$

也常将  $m$ -阶闵可夫斯基距离称为  $m$ -阶范数,1-阶范数即为曼哈顿距离,而 2-阶范数即为最常见的欧几里得距离。 $m$ -阶范数的特点是,采用不同的  $m$  值,状态空间中相同两点之间的距离则不同。

**定义 3(检测器)** 检测器采用二元组描述为  $d = \{c, r\} | c \in U, r \in R$ ,其中  $c$  为检测器在状态空间中的位置, $r$  为检测器的检测半径, $R$  为实数集合。训练阶段用自体数据集合  $S \subset Self$  进行耐受,设  $S$  中距离检测器  $d$  最近的自体用函数  $Nearest$  表示:

$$Nearest(d, S) = \{s_n | s_n \in S, \forall s \in S, f(d, c, s_n) \leq f(d, c, s)\} \quad (2)$$

设  $r_{self}$  为自体半径,则对于任意检测器  $d$ ,如果  $f(d, c, Nearest(d, S)) < r_{self}$ ,则检测器  $d$  匹配自体。

**定义 4(受体编辑)** 通过调整检测器的检测位置和检测半径,使得检测器在避免匹配自体的同时更好地覆盖非自体空间,称该调整过程为受体编辑。设随机产生未成熟检测器集  $D_U$ ,首先对匹配自体的检测器集  $D_S$  进行受体编辑,以避免其覆盖自体空间,其中匹配自体且不能进行受体编辑的检测器集  $D_A$  将被删除。然后对余下的检测器集  $D_P = D_U - D_A$  进行进一步的受体编辑,以扩大对非自体空间的覆盖并最终成为成熟检测器集  $D_M$ ,并且  $D_M$  中检测器  $d$  的检测半径用如下公式计算:

$$d, r = \begin{cases} f(d, c, Nearest(d, S)) - r_{self}, & \text{点感知} \\ f(d, c, Nearest(d, S)), & \text{边界感知} \end{cases} \quad (3)$$

式中,点感知(point-aware)和边界感知(boundary-aware)是文献[12]提出的两种检测半径确定技术。

**定义 5(检测)** 检测过程使用成熟检测器集  $D_M$  来判断被检数据是自体还是非自体。设待检数据  $x \in U$ ,检测过程用函数  $match$  表示:

$$match(x) = \begin{cases} 1, & \text{if } \exists d \in D_M, f(d, c, x) < d, r \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

函数  $match$  返回 1 表示被检数据是非自体,返回 0 则表示被检数据是自体。

### 2.2 两种受体编辑方法

在生物免疫系统中,T淋巴细胞受体编辑机制的具体解释目前仍然存在争议,部分免疫学家认为受体编辑是完全随机的,而部分免疫学家则认为受体编辑具有方向性<sup>[13,14]</sup>。本文借鉴受体编辑具有方向性的解释,并根据检测器是否匹配自体的条件分别采取两种受体编辑方法:定向受体编辑 DRE(Directional Receptor Editing)和识别相同最近自体的定向受体编辑 DREIINS(Directional Receptor Editing for Identifying Identical Nearest Self)。其中 DRE 应用于匹配自体的检测器,DREIINS 则应用于没有匹配自体的检测器。

#### 2.2.1 定向受体编辑

定向受体编辑 DRE 的目的是通过定向移动匹配自体的检测器使它不再匹配自体。设需要定向受体编辑的检测器为  $d$ ,则移动该检测器的检测位置,使其远离初始最近自体  $Nearest(d, S)$ ,移动方向向量的具体计算公式如下:

$$dir = \frac{d, c - Nearest(d, S)}{\|d, c - Nearest(d, S)\|} \quad (5)$$

“定向”是指移动方向向量  $dir$  一旦确定就不再改变,检测器  $d$  定向移动后的检测位置用如下公式计算:

$$d, c = d, c + dir * \eta \quad (6)$$

式中, $\eta$  为移动步长,检测器通过定向移动后,如果仍然匹配自体,则继续定向移动直至不再匹配自体或因超出范围无法移动而失败。图 1 所示为本文提出的 DRE 方法采用欧式距离度量在二维状态空间中的运行示意图。

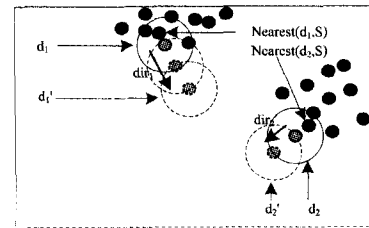


图 1 定向受体编辑

在图 1 中,黑色实心圆代表自体,空心圆  $d_1$  和  $d_2$  分别表示两个匹配自体的检测器, $Nearest(d_1, S)$  和  $Nearest(d_2, S)$  分别为这两个检测器的初始最近自体, $dir_1$  和  $dir_2$  分别为这两个检测器的移动方向向量,移动步长取值为自体半径  $r_{self}$ ,空心圆  $d_1'$  和  $d_2'$  则代表经过定向移动后不再匹配自体的检测器,其中  $d_1'$  经过两次定向移动,而  $d_2'$  经过一次定向移动。下面给出 DRE 过程的算法流程。

$d$  为检测器, $S$  为训练用自体集合, $\eta$  为检测器的移动步长。

- 步骤 1 采用式(5)确定检测器  $d$  的移动方向向量  $dir$ ;
- 步骤 2 采用式(6)移动检测器  $d$ ;
- 步骤 3 如果移动后的检测器超出范围,则受体编辑失败并退出;
- 步骤 4 如果移动后的检测器仍然匹配自体,则转到步骤 2,否则受体编辑成功并退出。

### 2.2.2 识别相同最近自体的定向受体编辑

与 DRE 避免检测器匹配自体的目标不同, DREIINS 的目标则是在覆盖原检测范围的情况下最大化没有匹配自体的检测器的检测范围。DREIINS 同样采用定向移动机制,其移动方向向量的计算公式同 DRE 一样,但与 DRE 不同的是, DREIINS 还需要覆盖原检测范围,因此需要移动后的检测器满足识别相同最近自体的限制条件。

**定义 6(识别相同最近自体)** 识别相同最近自体是指受体编辑前的检测器  $d$  和受体编辑后的检测器  $d'$  与自体集  $S$  中的同一自体距离最近,即满足  $Nearest(d, S) = Nearest(d', S)$ 。

本质上,可将 DREIINS 的目标看作约束最优化问题:在满足移动方向不变且识别相同最近自体的约束下,最大化其检测范围(检测半径)。图 2 给出了二维状态空间中采用欧式距离度量及边界感知技术<sup>[10]</sup>的 DREIINS 示意图。

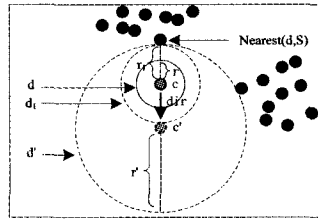


图 2 识别相同最近自体的定向受体编辑

在图 2 中,黑色实心圆代表自体,空心圆  $d$  代表需要进行 DREIINS 的检测器,  $Nearest(d, S)$  为检测器  $d$  的最近自体,  $dir$  为 DREIINS 的移动方向,空心圆  $d_1$  表示直接利用式(3)计算检测半径后的检测器。DREIINS 机制的目标是使检测器  $d(c, r)$  最终改进为检测器  $d'(c', r')$ ,由“定向受体编辑”可知检测器  $d$ 、检测器  $d'$  和自体  $Nearest(d, S)$  在同一直线上,由“识别相同最近自体”的约束可知  $Nearest(d, S) = Nearest(d', S)$ 。事实上,图 2 所示的 3 个检测器  $d(c, r)$ 、 $d_1(c, r_1)$  和  $d'(c', r')$  正好分别代表了采用固定检测半径<sup>[8]</sup>、可变检测半径<sup>[9-12]</sup>和 DREIINS 3 种机制生成的检测器,从图中能直观地看出,3 种机制各不相同, DREIINS 为最优(具体证明参见定理 2)。

作为约束最优化问题的 DREIINS 可以有多种算法实现,下面给出采用折半查找思想的算法流程。

$d$  为检测器,  $S$  为训练用自体集合,  $\eta$  为检测器的初始移动步长,  $\eta_{min}$  为最小移动步长。

- 步骤 1 采用式(5)确定检测器  $d$  的移动方向向量  $dir$ ;
- 步骤 2 采用式(6)移动检测器  $d$ ;
- 步骤 3 如果移动成功并且识别相同最近自体,则转到步骤 2,否则取消移动并减半移动步长;
- 步骤 4 如果移动步长大于最小移动步长  $\eta_{min}$ ,则转到步骤 2,否则受体编辑成功并退出。

### 2.3 基于受体编辑的实值阴性选择算法

本文提出的 RERNS 算法同时采用了 DRE 和 DREIINS 两种受体编辑方法,其中 DRE 增加了检测器的多样性,而 DREIINS 则扩大了检测器的检测范围。下面给出 RERNS 的算法流程。

设  $S$  为训练用自体集合,  $r_{self}$  为自体半径。

- 步骤 1 随机生成实值表达的检测器集  $D_U$ ;
- 步骤 2 对匹配自体的检测器集  $D_S$  中的检测器采用 DRE 进行受体编辑;
- 步骤 3 删除  $D_S$  中受体编辑失败检测器集  $D_A$ ;
- 步骤 4 对余下的检测器集  $D_P = D_U - D_A$  中的检测器采用 DREIINS 进行受体编辑,编辑后加入成熟检测器集  $D_M$ ;
- 步骤 5 采用式(3)确定检测器集  $D_M$  中检测器的检测半径;
- 步骤 6 判断生成的成熟检测器集  $D_M$  是否满足要求,如果满足,则退出训练阶段,否则转到步骤 1。

RERNS 算法中,步骤 6 的退出条件“是否满足要求”可以有多种具体实现:生成指定的检测器数、用公式计算所需生成的检测器数<sup>[1]</sup>、采用 Monte carlo 统计估计计算所需检测器数<sup>[6]</sup>和结合统计估计及假设检验动态生成检测器<sup>[11]</sup>等。这些退出条件非本文讨论重点,不再详述。

## 3 算法性能分析

基于传统免疫理论的阴性选择算法是直接删除匹配自体的检测器,而 RERNS 算法则通过 DRE 使匹配自体的无效检测器通过定向移动成为不再匹配自体的有效检测器。

**定理 1** 通过 DRE 而存活的检测器分布在非自体空间中紧靠自体空间的区域。

证明:设通过 DRE 而存活的检测器为  $d'(c, r)$ 。由 DRE 退出条件“检测器不匹配自体”可得:

$$r_{self} \leq \|d'.c - Nearest(d', S)\| < r_{self} + \eta$$

即检测器  $d'(c, r)$  分布在其与最近自体相距  $r_{self}$  和  $r_{self} + \eta$  之间的区域,该区域正是非自体空间中紧靠自体空间的区域。

阴性选择算法中没被检测器覆盖的“漏洞”普遍存在于自体与非自体之间的边界区域<sup>[5]</sup>,因此,充分覆盖这部分区域对于提高阴性选择算法的检测性能十分关键。从定理 1 可知,采用 DRE 的 RERNS 算法可以使匹配自体的检测器在自体和非自体边界区域获得新生,因此,与直接删除匹配自体检测器的阴性选择算法相比, RERNS 算法一方面能减少随机生成的未成熟检测器数,另一方面其最终生成的成熟检测器也能以更大的机率覆盖存在于自体与非自体边界区域的“漏洞”。

**定理 2** 与固定检测半径和可变检测半径相比,对于同一检测器,采用 DREIINS 对非自体空间的覆盖最优。

证明:对于同一检测器,采用可变检测半径时,检测器的半径可以调整变大,显然其对非自体空间的覆盖优于采用固定检测半径,因此只需证明采用 DREIINS 优于采用可变检测半径即可。

设算法工作在  $n$  维状态空间  $U$ ,采用闵可夫斯基距离函数  $f$ ,非自体  $ag \in U$  可被可变检测半径的检测器  $d(c, r)$  检测,检测器  $d$  采用 DREIINS 成为检测器  $d'(c, r)$ 。

$$\text{由式(4)可得 } f(d.c, ag) < d.r;$$

由三角不等式定理,可得:

$$f(d'.c, ag) \leq f(d.c, ag) + f(d.c, d'.c) < d.r + f(d.c, d'.c)$$

由于  $dir = \frac{d.c - Nearest(d, S)}{\|d.c - Nearest(d, S)\|}$ ,因此检测器  $d$ 、检测器  $d'$  和自体  $Nearest(d, S)$  在同一直线上,有

$$d'.r = d.r + f(d.c, d'.c),$$

$$\text{可得 } f(d'.c, ag) < d'.r,$$

故非自体  $ag$  可被  $d'$  检测;

即受体编辑后的检测器  $d'$  包含了受体编辑前检测器  $d$  的覆

盖范围,且  $d'.r = d.r + f(d.c, d'.c) \geq d.r$ 。

可得  $d'$  对非自体空间的覆盖优于  $d$ 。

由于 RERNS 算法生成的成熟检测器集  $D_M$  中的每个检测器都经过 DREIINS, 因此相比于采用固定检测半径的 RNS 算法和采用可变检测半径的 V-detector 算法, 采用 DREIINS 的 RERNS 算法能获得更好的检测能力。

**定理 3** DRE 和 DREIINS 的时间复杂度都为  $O(|S|)$ 。

证明: 设算法运行在  $n$  维状态空间, 由于定向移动的性质, 采用 DRE 的检测器的定向移动次数最多为  $\lceil \sqrt{n}/\eta \rceil$  次, 而采用 DREIINS 的检测器的定向移动次数最多为  $\lceil \sqrt{n}/\eta \rceil + \lceil \ln(\eta/\eta_{\min}) \rceil$  次。在  $n, \eta$  和  $\eta_{\min}$  取值确定的条件下, DRE 的移动次数小于常数  $\lceil \sqrt{n}/\eta \rceil$ , 而 DREIINS 的移动次数则小于常数  $\lceil \sqrt{n}/\eta \rceil + \lceil \ln(\eta/\eta_{\min}) \rceil$ 。由于每次移动后都需要与自体集  $S$  中的每个自体进行比较, 因此 DRE 和 DREIINS 的时间复杂度都为  $O(|S|)$ 。

## 4 实验结果与分析

为验证 RERNS 算法的有效性, 实验选择了实值阴性选择算法中具有代表性的 RNS 算法<sup>[8]</sup> 和 V-detector 算法<sup>[9-12]</sup> 作为对比对象, 除了直接比较 3 种算法外, 实验还验证了 DRE 和 DREIINS 单独一种机制对 RERNS 算法检测性能的影响。实验数据选择了实值阴性选择算法研究中常用的两个数据集: 一个是 Memphis 数据集, 另一个是 Fisher's Iris 数据集。在具体对比时, 为了避免不同退出条件对算法的影响, 所有算法都统一采用相同的退出条件“生成指定的检测器数”。

### 4.1 Memphis 数据集

美国 Memphis 大学的 Dasgupta 教授是人工免疫系统研究领域的知名学者, 其研究团队提供和使用的数据集是目前实值阴性选择算法性能测试的权威数据集<sup>[8-12]</sup>。该数据集由多个子数据集组成, 这里选择其中具有代表性的 Ring 和 Stripe 子数据集作为实验数据。图 3 所示为 3 种算法在 Ring 训练数据集上的一次运行结果, 算法运行参数: 2-阶范数距离度量、点感知、50 个成熟检测器、自体半径为 0.03、 $\eta$  为 0.03、 $\eta_{\min}$  为 0.00001。图 4 所示为 3 种算法在 Stripe 训练数据集上的一次运行结果, 算法运行参数:  $\infty$ -阶范数距离度量、10 个成熟检测器、点感知、自体半径为 0.03、 $\eta$  为 0.03、 $\eta_{\min}$  为 0.00001。

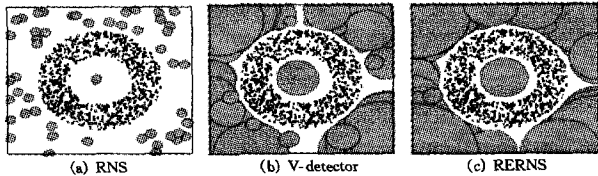


图 3 检测器对 Ring 非自体空间的覆盖情况

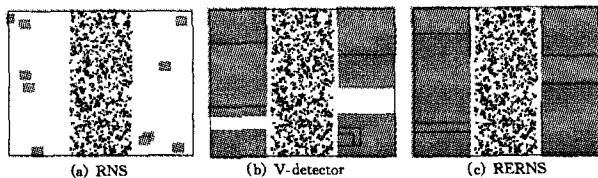


图 4 检测器对 Stripe 非自体空间的覆盖情况

图 3 和图 4 中黑点代表自体数据, 灰色填充部分代表检测器覆盖的区间(采用不同阶范数距离度量的检测器具有不同的几何形状, “2-阶”和“ $\infty$ -阶”分别为圆形和正方形), 空白

部分代表非自体区间。可以直观地看出, 以检测器对非自体空间的覆盖(灰色填充部分所覆盖的区域)作为评价标准, RERNS 算法在 Ring 和 Stripe 上都优于 RNS 和 V-detector 算法。

图 5 和图 6 分别为 3 种算法在 Ring 和 Stripe 数据集上检测器对非自体空间覆盖率的对比图。其中成熟检测器数量取值从 1 到 100, 3 种算法的其他运行参数和前面实验相一致。覆盖率的计算采用文献<sup>[11]</sup>提出的方法进行 Monte Carlo 统计估计。为了减少运行误差, 图中给出的结果都为 100 次运行的均值, 标准差用误差棒绘出。从图 5 和图 6 可以看出, 当产生相同数量的检测器时, 3 种算法中 RERNS 算法对非自体空间的覆盖率总是最优的, 这进一步证实了 RERNS 算法的有效性。

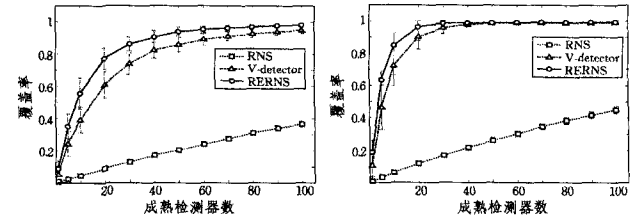


图 5 Ring 数据集上的覆盖率 图 6 Stripe 数据集上的覆盖率

表 1 所列 3 种算法在 Ring 和 Stripe 测试数据集上检测率、生成的未成熟检测器数、成熟检测器平均半径的对比。其中 RERNS1 表示只采用了 DRE 机制的 RERNS 算法, RERNS2 表示只采用了 DREIINS 机制的 RERNS 算法, 而 RERNS3 表示同时采用了 DRE 和 DREIINS 机制的 RERNS 算法。算法运行参数: 500 个成熟检测器、自体半径  $r_{self}$  分别取值为 0.03、0.05 和 0.1, 其它参数取值和前面实验一致。表中的数据都为 100 次运行的均值, 由于采用点感知机制, 此时几种算法的误报率都为 0, 因此检测率的高低可直接作为算法检测性能优劣的评价标准。

从表 1 可以发现, 在相同的运行参数取值下, 单独采用 DRE 的 RERNS1 算法和单独采用 DREIINS 的 RERNS2 算法的检测率都要优于 RNS 和 V-detector 算法, 而同时采用 DRE 和 DREIINS 的 RERNS3 算法, 则生成了最少的未成熟检测器并获得了最高的检测率。单独观察 DRE 和 DREIINS 对 RERNS 算法性能的影响, 可以发现, DRE 机制减少了生成的未成熟检测器数, 并且在检测器平均半径减小的情况下仍然提高了检测率。这是因为根据定理 1, 匹配自体的检测器获得新生后分布在自体和非自体的边界区域, 从而提高了算法对边界区域的覆盖率。同时可以发现, DREIINS 机制显著地扩大了检测器平均半径, 并且同时提高了算法的检测率, 这是由于其最大化了每个检测器检测范围的结果。

### 4.2 Fisher's Iris 数据集

在模式识别领域广泛采用的 Fisher's Iris 数据集也常用于实值阴性选择算法的性能测试<sup>[9-12]</sup>, 该数据集包括 3 种类别的数据: setosa、versicolor 和 virginica。每一类数据包括 50 个元素, 每个元素包括 4 个属性。为了使用该数据集, 首先将元素属性标准化到  $[0, 0, 1, 0]$  区间。实验中分别选择一类数据作为训练自体数据, 全部 Iris 数据都作为测试数据, 共进行 3 个对比实验, 比较结果如表 2 所列, 算法运行参数: 500 个成熟检测器、2-阶范数距离度量、边界感知、 $\eta$  为 0.1、 $\eta_{\min}$  为 0.00001、自体半径  $r_{self}$  分别取值为 0.03、0.1 和 0.2。由于自体数据都

是训练数据,因此几种算法的误报率都为 0。为了减少运行误差的影响,表 2 中的数据都为 100 次运行结果的均值。

表 1 3 种算法在 Memphis 数据集上的对比

数据集	算法	$r_{self}=0.03$			$r_{self}=0.05$			$r_{self}=0.1$		
		检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)	检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)	检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)
RNS	V-detector	72.09 (1.38)	879.58 (28.04)	0.03 (0.0000)	72.66 (0.93)	1131.21 (44.92)	0.05 (0.0000)	41.84 (0.58)	3359.69 (143.53)	0.1 (0.0000)
		87.94 (0.15)	743.72 (18.33)	0.14 (0.0036)	80.2 (0.21)	823.92 (23.7)	0.13 (0.0036)	54.89 (0.31)	1132.93 (35.12)	0.11 (0.0030)
		88.02 (0.18)	500.00 (0.00)	0.11 (0.0043)	80.29 (0.18)	500.00 (0.00)	0.10 (0.0042)	55.22 (0.15)	517.24 (4.49)	0.08 (0.0028)
Ring	RERNS1	88.17 (0.10)	747.44 (16.51)	0.20 (0.0038)	80.46 (0.12)	833.15 (23.55)	0.19 (0.0031)	55.26 (0.13)	1129.23 (37.4)	0.16 (0.0034)
		88.30 (0.08)	500.00 (0.00)	0.18 (0.0044)	80.59 (0.09)	500.00 (0.00)	0.17 (0.0039)	55.37 (0.11)	517.36 (3.91)	0.14 (0.0031)
		81.25 (1.24)	900.34 (27.18)	0.03 (0.0000)	80.27 (0.87)	1037.66 (28.71)	0.05 (0.0000)	67.88 (0.47)	1794.47 (73.78)	0.1 (0.0000)
Stripe	RERNS1	90.05 (0.14)	809.54 (23.68)	0.15 (0.0038)	84.08 (0.11)	854.21 (23.47)	0.14 (0.0033)	70.78 (0.06)	1041.22 (34.77)	0.14 (0.0029)
		90.43 (0.15)	551.84 (7.39)	0.11 (0.0042)	85.17 (0.09)	564.48 (7.69)	0.10 (0.0037)	70.80 (0.01)	584.74 (10.28)	0.11 (0.0031)
		90.23 (0.17)	806.70 (24.48)	0.24 (0.0043)	85.19 (0.09)	857.48 (25.7)	0.23 (0.0037)	70.80 (0.02)	1036.78 (32.94)	0.21 (0.0020)
V-detector	RERNS2	90.64 (0.13)	554.68 (6.85)	0.19 (0.0053)	85.30 (0.08)	564.11 (9.02)	0.19 (0.0046)	70.81 (0.00)	582.94 (9.82)	0.20 (0.0022)
		90.43 (0.15)	551.84 (7.39)	0.11 (0.0042)	85.17 (0.09)	564.48 (7.69)	0.10 (0.0037)	70.80 (0.01)	584.74 (10.28)	0.11 (0.0031)
		90.23 (0.17)	806.70 (24.48)	0.24 (0.0043)	85.19 (0.09)	857.48 (25.7)	0.23 (0.0037)	70.80 (0.02)	1036.78 (32.94)	0.21 (0.0020)

表 2 3 种算法在 Fisher's Iris 数据集上的对比

自体数据	算法	$r_{self}=0.03$			$r_{self}=0.1$			$r_{self}=0.2$		
		检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)	检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)	检测率/% (标准差/%)	未成熟检测器 数(标准差)	检测器平均半 径(标准差)
Setosa	V-detector	0.17 (0.51)	500.09 (0.29)	0.03 (0.0000)	23.17 (8.42)	504.14 (2.00)	0.10 (0.0000)	98.26 (2.48)	522.51 (4.33)	0.20 (0.0000)
		100 (0.00)	500.16 (0.37)	0.7 (0.0099)	100 (0.00)	503.87 (1.95)	0.71 (0.0105)	100 (0.00)	522.99 (5.37)	0.72 (0.0099)
		100 (0.00)	500 (0.00)	0.7 (0.0104)	100 (0.00)	501.28 (1.15)	0.70 (0.0107)	100 (0.00)	512.23 (3.26)	0.71 (0.0104)
Versicolor	RERNS1	100 (0.00)	500.1 (0.33)	0.86 (0.0092)	100 (0.00)	504.17 (2.05)	0.87 (0.0090)	100 (0.00)	521.25 (4.88)	0.88 (0.0087)
		100 (0.00)	500 (0.00)	0.86 (0.0096)	100 (0.00)	500.90 (0.94)	0.86 (0.0104)	100 (0.00)	511.52 (3.88)	0.87 (0.0093)
		0.13 (0.36)	500.14 (0.37)	0.03 (0.0000)	20.90 (8.09)	505.14 (2.15)	0.10 (0.0000)	88.04 (6.48)	535.41 (6.09)	0.20 (0.0000)
Verginica	RERNS2	99.86 (0.35)	500.15 (0.38)	0.46 (0.0065)	99.82 (0.38)	505.56 (2.44)	0.46 (0.0067)	99.78 (0.41)	535.97 (6.19)	0.48 (0.0055)
		99.89 (0.31)	500 (0.00)	0.46 (0.0066)	99.88 (0.32)	500.08 (0.31)	0.46 (0.0063)	99.87 (0.34)	504.6 (2.24)	0.47 (0.0067)
		99.94 (0.24)	500.12 (0.38)	0.57 (0.0063)	99.92 (0.27)	505.21 (2.39)	0.57 (0.0064)	99.89 (0.31)	536.57 (5.48)	0.58 (0.0057)
RNS	RERNS3	99.97 (0.16)	500 (0.00)	0.57 (0.0063)	99.97 (0.17)	500.05 (0.22)	0.57 (0.0056)	99.94 (0.24)	504.16 (2.22)	0.57 (0.0061)
		0.23 (0.49)	500.11 (0.31)	0.03 (0.0000)	21.20 (9.41)	506.96 (2.73)	0.10 (0.0000)	89.79 (5.58)	545.5 (6.44)	0.20 (0.0000)
		99.51 (0.49)	500.09 (0.35)	0.52 (0.0087)	99.50 (0.50)	506.76 (2.45)	0.52 (0.0091)	99.33 (0.47)	546.03 (6.94)	0.54 (0.0085)
V-detector	RERNS1	99.60 (0.50)	500 (0.00)	0.52 (0.0075)	99.57 (0.49)	500.46 (0.64)	0.52 (0.0082)	99.37 (0.48)	517.52 (3.89)	0.53 (0.0081)
		99.74 (0.44)	500.1 (0.33)	0.64 (0.0090)	99.72 (0.45)	507.07 (2.71)	0.64 (0.0087)	99.52 (0.5)	547.15 (6.92)	0.66 (0.0073)
		99.81 (0.39)	500 (0.00)	0.64 (0.0076)	99.79 (0.41)	500.53 (0.68)	0.64 (0.0076)	99.68 (0.47)	516.84 (3.89)	0.65 (0.0078)

从表 2 可以发现,采用固定检测半径的 RNS 算法的检测率对自体半径的取值十分敏感,当自体半径取值为 0.03 和 0.1 时,其检测率很低,这是由于在 4 维状态空间中,检测半径固定为 0.03 和 0.1 的检测器覆盖范围相对于整个状态空间非常小;采用可变检测半径的 V-detector 算法相比 RNS 算法有较大的检测性能提升;单独采用 DRE 和 DREINS 对

RERNS 算法的检测率都有提升作用,而同时采用 DRE 和 DREINS 的 RERNS 算法则获得了最好的检测性能,该实验结论与在 Memphis 数据集上获得的结论一致。

**结束语** 目前已有的阴性选择算法研究都源于传统免疫理论。传统免疫理论认为 T 淋巴细胞受体不可变,因此,基于该理论的阴性选择算法中的检测器也往往缺乏适应能力。

与传统免疫理论不同,近年兴起的受体编辑理论认为 T 淋巴细胞在一定条件下可以改变自身受体的特异性及其亲和力。受该理论启发,本文提出了一种基于受体编辑的实值阴性选择算法 RERNS,在 RERNS 算法中采用了两种受体编辑方法:DRE 和 DREIINS。其中 DRE 改善了算法对边界区域的覆盖;DREIINS 扩大了算法对非自体空间的覆盖。理论分析与实验验证表明,RERNS 算法的检测性能优于 RNS 和 V-detector 算法。

### 参考文献

[1] Forrest S, Perelson A S, Allen L, et al. Self-nonsel self discrimination in a computer[C]//Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Soc Press, 1994: 202-212

[2] Forrest S, Beauchemin C. Computer immunology[J]. Immunol Rev, 2007, 216(1): 176-197

[3] 李涛. 计算机免疫学[M]. 北京: 电子工业出版社, 2004: 147-159

[4] Timmis J. Artificial immune systems—today and tomorrow[J]. Natural Computing, 2007, 6(1): 1-18

[5] Balthrop J, Esponda F, Forrest S, et al. Coverage and generalization in an artificial immune system[C]// GECCO 2002. New York: Morgan Kaufmann Publishers Inc, 2002: 3-10

[6] 张衡, 吴礼发, 张毓森, 等. 一种  $r$  可变阴性选择算法及其仿真分析[J]. 计算机学报, 2005, 28(010): 1614-1619

[7] 何申, 罗文坚, 王煦法. 一种检测器长度可变的非选择算法[J]. 软件学报, 2007, 18(6): 1361-1368

[8] Gonzalez F, Dasgupta D, Gomez J. The effect of binary matching rules in negative selection[C]//GECCO 2003. Chicago: Springer Berlin, 2003: 196-206

[9] Ji Z, Dasgupta D. Real-valued negative selection algorithm with variable-sized detectors [C] // GECCO 2004. Washington: Springer Berlin, 2004: 287-298

[10] Ji Z. A boundary-aware negative selection algorithm[C]//The Ninth IASTED. Benidorm: Acta Press, 2005: 29-42

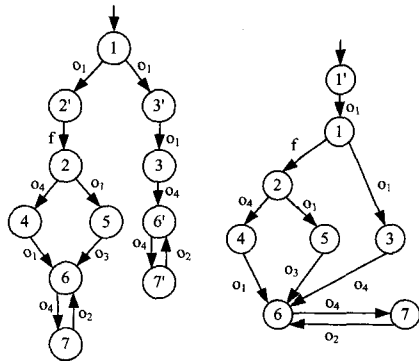
[11] Ji Z, Dasgupta D. Estimating the detector coverage in a negative selection algorithm [C] // GECCO 2005. Washington: ACM Press, 2005: 281-288

[12] Ji Z, Dasgupta D. V-detector: An efficient negative selection algorithm with “probably adequate” detector coverage[J]. Information Sciences, 2009, 179(10): 1390-1406

[13] Nemazee D. Receptor editing in lymphocyte development and central tolerance[J]. Nature Reviews Immunology, 2006, 6(10): 728-740

[14] 罗微, 马骊, 王小宁. T 细胞受体编辑与修正[J]. 中华微生物学和免疫学杂志, 2008, 28(3): 278-281

(上接第 214 页)



(a) 未合并状态之前的图 (b) 合并状态之后的图

图 6

第 3 节中的 3 个性质包括了通信事件和可观测事件之间时序信息的所有情况,所以单独抽取时序信息对可诊断性的判定和全局模型可诊断性判定的结果是一致的。本文侧重的是快速地对可诊断性做出判定。提取时序信息由于简化了很多其他信息,因此其带来的后果是不能得出具体的关键路径。

在全局模型分解成局部模型时,可以对一些状态进行合并以达到缩减模型规模的效果。与文献[10]的方法类似,从开始状态到无故障状态出现的路径  $p$ ,若有一不含故障的路径和  $p$  有相同的观测,则合并这两条路径,例如图 6(a)中的事件  $o_1$ ;出现故障状态的路径,若存在一条路径到达稳定状态的观测与其一致,则同样可以合并一致的观测;例如图 6(a)的  $o_2 o_4$  循环,图 6(b)是图 6(a)经过状态合并后得到的图,图 6(b)中从状态 1 开始考虑模型的划分,即是第 2 节中用到的全局模型。如果全局模型是可诊断的,在划分成局部模型时,域的选择对可诊断性的判断有很大的影响,有可能在划分出的某一个局部模型中能直接判断出该系统是可诊断的。所以,域的有效选择对局部模型可诊断性的判定有很大影响。以后要做的工作是,如何选取合适的域,把一个全局模型划分

成  $n$  个局部模型;如何运用时序信息,以最少的局部模型判定全局模型的可诊断性。

### 参考文献

[1] Lin F. Diagnosability of discrete-event systems and its applications[J]. J. Discrete Event Dyn. Syst.: Theory Appl., 1994, 4(2): 197-212

[2] Bavishi S. Automated fault diagnosis using a discrete event systems framework[C]//Proceedings of the 1994 IEEE International Symposium on. 1994: 213-218

[3] Sampath M, Sengupta R, Lafortune S, et al. Diagnosability of discrete-event systems in IEEE Trans[J]. Automat. Contr., 1995, 40: 1555-1575

[4] Jiang S, Huang Z, Chandra V, et al. A polynomial algorithm for testing diagnosability of discrete event systems[J]. IEEE Transactions on Automatic Control, 2001, 46(8): 1318-1321

[5] Yoo T S, Lafortune S. Polynomial-time verification of diagnosability of partially observed discrete-event systems [J]. IEEE Transactions on Automatic Control, 2002, 47(9): 1491-1495

[6] Debouk R, Lafortune S, Teneketzis D. Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems[J]. Discrete Event Dynamic Systems, 2000, 10(1/2): 33-86

[7] Pencolé Y. Diagnosability analysis of distributed discrete event systems[C]//Proceedings of European Conference on Artificial Intelligence(ECAI'04). 2004: 43-47

[8] Grastien A. Symbolic Testing of Diagnosability[C]//DX 09. 2009: 131-138

[9] Khoumsi A. Coordination of Components in a Distributed Discrete-Event System [C] // Processing of the 4th International Symposium on Parallel and Distributed Computing(ISPDC'05). 2005: 299-306

[10] Schumann A. Towards Efficiently Diagnosing Large Scale Discrete-Event Systems[D]. Australian National University, 2007