论 文

二次否定选择算法

郑旭飞^①, 方永慧^{②*}, 李涛^③^① 西南大学计算机与信息科学学院, 重庆 400715^② 西南大学电子信息工程学院, 重庆 400715^③ 四川大学计算机学院, 成都 610065

* 通信作者. E-mail: fyhui@swu.edu.cn

收稿日期: 2012-07-03; 接受日期: 2012-08-22

国家自然科学基金 (批准号: 61173159, 60873246) 和教育部重大培育基金 (批准号: 708075) 资助项目

摘要 否定选择算法 (NSA) 是人工免疫系统应用于异常检测生成检测器的重要算法, 传统 NSA 随机产生候选检测器与全部训练集进行耐受以消除免疫自反应, 该匹配过程是 NSA 的主要时间开销, 由于候选检测器在自体耐受过程中未考虑其与已有成熟检测器集的相互覆盖, 导致生成的成熟检测器与已有检测器重复覆盖, 经历不必要的自体耐受, 从而导致 NSA 生成检测器数量过多, 检测器的生成效率过低, 限制了人工免疫系统在异常检测中的应用. 为此, 本文提出了二次否定选择算法 (2-NSA), 算法包括两次否定选择过程, 分别耐受检测器集和训练集. 每个随机产生的候选检测器先与已有成熟检测器集耐受为第一次否定选择, 清除识别已有成熟检测器的候选检测器, 耐受成功的候选检测器成为半成熟检测器; 半成熟检测器在已有成熟检测器覆盖之外进行训练集的自体耐受为第二次否定选择, 清除识别自体的半成熟检测器, 耐受成功的半成熟检测器成为成熟检测器加入检测器集合. 2-NSA 算法有效避免了候选检测器在已有成熟检测器覆盖范围之内的自体耐受, 大大减少了成熟检测器的数量, 提高了成熟检测器集的生成效率, 降低了算法的时间复杂度. 此外, 2-NSA 算法按检测器半径从大到小优先产生覆盖范围更大的检测器, 进一步避免与已有成熟检测器的重复覆盖, 减少成熟检测器的数量. 理论分析表明 2-NSA 算法有效减小了成熟检测器数量、提高了检测器生成效率, 降低了系统的误报率. 对比实验结果表明: 在标准数据集 Iris 和期望覆盖率为 99% 的情况下, 与经典的 RNSA 和 V-Detector 等实值否定选择算法相比, 2-NSA 算法需要成熟检测器的数量分别减少了 99.84% 和 95.69%, 误报率分别降低了 60.13% 和 50.90%, 产生成熟检测器集的时间代价分别缩减了 99.79% 和 66.84%.

关键词 人工免疫系统 否定选择算法 检测器 变半径检测器 二次否定选择算法

1 引言

受生物免疫系统否定选择^[1]、克隆选择^[2]、免疫网络^[3]和危险理论^[4]等启发, 以 Forrest 所提出的否定选择算法^[5] (negative selection algorithm, NSA) 为代表的人工免疫系统对当前和未来大型 IT 系统的自动和自适应保护提出了新的解决方案, 因其克服了传统异常检测系统的缺陷, 被认为是非

引用格式: 郑旭飞, 方永慧, 李涛. 二次否定选择算法. 中国科学: 信息科学, 2013, 43: 529–544, doi: 10.1360/112011-1409

常重要且有意义的研究^[6~8]. NSA 通过模拟生物系统中 T -细胞成熟过程的免疫耐受, 清除自体耐受不成功的候选检测器, 达到对非自体的有效识别, 成功应用于模式识别和入侵检测等领域^[5,9].

早期的否定选择算法 (string representation negative selection algorithm, SNSA) 采用字符串或二进制串编码抗原 (样本) 和抗体 (检测器), 并使用 r -连续位匹配规则计算检测器与抗原的亲合程度^[5,10]. SNSA 算法存在着两个重大的不足: 检测器生成效率过低, 误报率过高. 文献 [10, 11] 指出 SNSA 存在检测器生成效率过低, 随机生成的候选检测器通过否定选择成为成熟检测器, 候选检测器数量 N 与训练集大小 N_s 呈指数关系 $N = \frac{-\ln(P_f)}{P_m \cdot (1-P_m)^{N_s}}$, 算法的时间复杂度为 $O(N \cdot N_s) = O\left(\frac{-\ln(P_f) \cdot N_s}{P_m \cdot (1-P_m)^{N_s}}\right)$ ^[5,12].

实值否定选择算法 (real-valued representation negative selection algorithm, RNSA) 采用 $[0, 1]^n$ 实值空间编码抗原和抗体, 并使用 Minkowski 距离计算抗原与抗体的亲和程度^[13]. 文献 [14, 15] 提出了可变半径的实值否定选择算法 (RNSA with variable-sized detector, V-Detector), 通过计算候选检测器中心与最近邻自体间的距离动态确定半径生成检测器, 取得了较好的检测效果. 文献 [16, 17] 分别提出了利用遗传算法和混沌理论优化检测器生成过程, 减少了候选检测器间的重叠覆盖. 文献 [18] 结合 K -Means 聚类方法优化否定选择过程, 并用于数据分析. 文献 [19] 则利用层次聚类方法对自体训练集进行聚类, 以聚类中心取代自体元素进行自体耐受, 提高了检测器的生成效率. 文献 [20] 在 V-Detector 算法基础上, 在自体耐受阶段根据部分自体训练集元素生成自体检测器进行再次训练, 加快检测器的生成效率. 此外, 文献 [21] 对否定选择进行了较为全面的综述. 但是, RNSA, V-Detector 算法及其相应改进算法均采用一次否定选择随机生成候选检测器耐受自体集, 只考虑了候选检测器与自体集的相互关系而未考虑其与已有成熟检测器集的重叠覆盖, 导致新生成的检测器与已有成熟检测器集对非自体空间的冗余覆盖, 达到所要求期望覆盖率所需成熟检测器数量显著增加, 检测器生成效率过低. 文献 [22] 指出, 由于成熟检测器的生成效率过低, 实值否定选择算法的执行时间严重限制了其实际应用. 文献 [23] 对使用进化策略的否定选择算法的复杂度进行了分析, 指出使用遗传算法等进化策略并不能根本上改进算法的时空复杂度.

因此, 本文提出二次否定选择算法 (dual negative selection algorithm, 2-NSA), 对候选检测器进行两次否定选择过程产生检测器集. 第一次否定选择过程: 随机产生的候选检测器先与成熟检测器集进行耐受, 清除识别已有成熟检测器的候选检测器, 耐受成功的候选检测器成为半成熟检测器, 保证了其中心点落在已有成熟检测器覆盖之外; 第二次否定选择过程: 半成熟检测器与训练集进行自体耐受, 清除识别自体的半成熟检测器, 耐受成功的半成熟检测器成为成熟检测器并加入检测器集合, 保证其对非自体空间的有效覆盖. 2-NSA 算法有效避免了当前实值否定选择算法中候选检测器在已有成熟检测器覆盖范围之内不必要的自体耐受, 大大减少了成熟检测器的数量, 提高了成熟检测器集的生成效率, 降低了算法的时间复杂度. 此外, 2-NSA 算法按检测器半径从大到小优先产生覆盖范围更大的成熟检测器, 进一步避免与已有成熟检测器的重复覆盖, 实现更少的检测器覆盖尽可能多的非自体区域, 减少成熟检测器的数量和提高检测器生成效率.

2 实值否定选择算法的基本定义

SNS(self non-self) 理论认为生物机体依靠抗体 (T -细胞和 B -细胞) 识别自体与非自体以排除异物达到机体的平衡和稳定^[1]. 受此启发, 在人工免疫理论中, 抗体被定义为检测器用于对非自体抗原的识别, 其质量决定了检测系统的准确性和有效性. 然而, 随机生成的候选检测器可能识别自体抗原而发生免疫自反应而产生误报. Forrest 通过模拟生物免疫中免疫细胞自体耐受而成熟的过程, 提

出了 NSA, 排除识别自体的检测器^[5]. 本文讨论基于实值表示的 NSA, 即 RNSA, 定义一些基本概念如下:

定义 1 抗原 $Ag = \{ag | ag = \langle x_1, x_2, \dots, x_n \rangle, x_i \in [0, 1]\}$, 表示问题空间中的所有样本, n 为数据维度.

定义 2 自体集 $Self \subset Ag$, 表示抗原集合中所有正常样本; 非自体集 $Nonself \subset Ag$, 表示抗原集合中所有异常样本, 有

$$Self \cup Nonself = Ag, \quad Self \cap Nonself = \emptyset. \quad (1)$$

定义 3 训练集 $Train \subset Self$, 表示检测的先验知识, 其中 $r_s \in [0, 1]$ 表示自体半径, N_s 表示训练集大小.

定义 4 检测器集 $D = \{d | d = \langle y_1, y_2, \dots, y_n, r_d \rangle, y_i \in [0, 1], r_d \in [0, 1]\}$, 是否定选择算法根据训练集而生成的成熟检测器集合, 其中 r_d 表示检测器半径, N_d 表示检测器集大小.

定义 5 估计覆盖率 C 表示一个采样周期中落在成熟检测器集 D 中的次数占采样点的比例, 其中 Num 为采样点, $Num_{covered}$ 为一个采样周期中落在成熟检测器集 D 中的次数.

$$C = \frac{Num_{covered}}{Num}. \quad (2)$$

定义 6 检测过程 $f(Train) \rightarrow Nonself$, 是根据训练集在抗原集合中找出非自体集的过程.

定义 7 检测率 TPR 表示检测器正确识别的非自体样本占总非自体样本的比例, 其中 TP 表示检测器正确识别的非自体的数量, FN 表示错误识别的非自体的数量.

$$TPR = \frac{TP}{TP + FN}. \quad (3)$$

定义 8 误报率 FPR 表示检测器错误识别的自体为非自体样本占总自体样本的比例, 其中 FP 表示检测器错误识别自体为非自体的数量, TN 表示正确识别自体的数量.

$$FPR = \frac{FP}{FP + TN}. \quad (4)$$

3 二次否定选择算法实现策略

3.1 RNSA 与 V-Detector 算法

实值否定选择算法通过亲和力和来表达候选检测器是否识别自体, 亲和力是根据 Minkowski 距离计算候选检测器中心与自体中心距离来表达的, 当该距离小于自体半径就表示该候选检测器识别了自体^[17]. 经典实值否定选择算法主要包括: 固定半径的 RNSA 算法和变半径的 V-Detector 算法.

RNSA 算法^[13] 采用固定的检测器半径 r_d , 以检测器数量作为算法的终止条件, 随机生成候选检测中心点 $X(x_1, x_2, \dots, x_n)$, 求得 X 与训练集中所有自体的最短距离 dis_{min} , 如果 $dis_{min} > r_s + r_d$, 则生成以 X 作为检测器中心点, r_d 为检测半径的成熟检测器, 其中 r_s 为自体半径, r_d 为检测器半径.

V-Detector 算法^[14,15] 采用变化的检测器半径, 以期望覆盖率作为算法的终止条件, 随机生成候选检测中心点 $X(x_1, x_2, \dots, x_n)$, 求得 X 与训练集中所有自体的最短距离 dis_{min} , 如果 $dis_{min} > r_s$, 则生成以 X 作为检测器中心点, $r_d = dis_{min} - r_s$ 为半径的成熟检测器.

不管是 RNSA 还是 V-Detector 算法, 均采用一次否定选择算法随机产生候选检测器经历训练集的自体耐受生成检测器, 检测器生成过程只考虑保证新生成检测器落在自体集之外覆盖非自体空间,

而未考虑与已有成熟检测器的重复覆盖. 随着估计覆盖率的上升, 新生成检测器与已有检测器重复覆盖机率随之加大, 大量重复覆盖的候选检测器都经历不必要的自体耐受过程, 增加了算法的时间开销, 从而导致成熟检测器数量过多, 检测器生成效率过低等问题.

3.2 二次否定选择算法: 2-NSA

针对当前实值否定选择算法的不足, 本文提出二次否定选择算法: 2-NSA. 算法采用变半径的检测器, 以期望覆盖率作为检测器生成的终止条件, 随机产生的候选检测器分别经历检测器集 D 的耐受产生半成熟检测器 (第一次否定选择), 保证半成熟检测器中心点落在检测器集 D 覆盖之外, 避免重复覆盖非自体空间; 半成熟检测器经历训练集的自体耐受生成成熟检测器 (第二次否定选择), 生成的成熟检测器加入检测器集 D 中.

第一次否定选择过程: 随机产生候选检测器 d_{new} 的中心点 $X(x_1, x_2, \dots, x_n)$, 求得 X 与检测器集 D 中每个成熟检测器 d_i 的欧几里德距离 $\text{dis}(d_{\text{new}}, d_i)$, 如果 d_{new} 与成熟检测器 D 中所有检测器 d_i 均满足公式 (5), 则通过检测器集 D 的耐受而成为半成熟检测器 d_{semi} ; 否则, 该候选检测器被任一成熟检测器识别 (不满足公式 (5)) 则被清除, 并重新随机产生新的候选检测器 d_{new} , 重新开始第一次否定选择过程.

$$\text{dis}(d_{\text{new}}, d_i) > r_{d_i} \quad i = 1, 2, \dots, N_d. \quad (5)$$

第二次否定选择过程: 通过第一次否定选择的半成熟检测器 d_{semi} 与训练集中每个自体进行自体耐受, 根据 Euclid 距离求得 d_{semi} 中心点 $Y(y_1, y_2, \dots, y_n)$ 与训练集中所有自体的最短距离 $\text{dis}_{\min}(d_{\text{semi}}, s_j)$, 如果公式 (6) 成立, 则该半成熟检测器 d_{semi} 通过训练集的自体耐受而成为成熟检测器 d_{mat} , 其检测器半径为 $r_d = \text{dis}_{\min}(d_{\text{semi}}, s_j) - r_s$, 并将该 d_{mat} 加入成熟检测器集合 D 中, $D \leftarrow D \cup \{d_{\text{mat}}\}$; 否则, 清除 d_{semi} 并随机产生新的候选检测器 d_{new} 重新开始第一次否定选择过程.

$$\text{dis}_{\min}(d_{\text{semi}}, s_j) > r_s, \quad s_j \in \text{Self}, \quad j = 1, 2, \dots, N_s. \quad (6)$$

此外, 2-NSA 按检测器半径从大到小优先产生覆盖范围更大的成熟检测器, 进一步避免与已有成熟检测器的重复覆盖, 实现更少的检测器覆盖尽可能多的非自体区域, 减少成熟检测器的数量, 提高检测器的生成效率. 为达到优先产生覆盖范围更大的成熟检测器 d_{mat} , 半成熟检测器 d_{semi} 被要求满足公式 (7) 才能成为成熟检测器. 其中 R_{Max} 表示每个训练集自体 S_y 与 n -维超立方体所有顶点 P_x 间距离的最大值 $\text{dis}_{\max}(p, s_y)$ 中的最小距离, 顶点与自体之间的距离使用 Euclid 距离计算, R_{Max} 按衰减因子 ε 逐渐减少, 直到 $R_{\text{Max}} \leq r_s$, 此时式 (7) 退化为式 (6). 二次否定选择算法过程如表 1 所示.

$$\begin{cases} \text{dis}_{\min}(d_{\text{semi}}, s_j) > R_{\text{Max}}, \\ \text{dis}_{\min}(d_{\text{semi}}, s_j) > r_s, \end{cases} \quad (7)$$

$$R_{\text{Max}} = \text{Min}(\text{dis}_{\max}(p, s_y)), \quad s_y \in \text{Self}, \quad y = 1, 2, \dots, N_s. \quad (8)$$

$$\text{dis}_{\max}(p, s_y) = \text{Max}(\text{dis}(p_x, s_y)), \quad x = 1, 2, \dots, 2^n. \quad (9)$$

$$\text{dis}(p, s) = \sqrt{\sum_{i=1}^n (p_i - s_i)^2}. \quad (10)$$

为了直观展示 2-NSA 算法与当前实值否定选择算法的区别, 采用 Iris 数据集进行了仿真实验. 实验选择 Iris 数据集中 Setosa 的 25 个实例作为自体训练集进行自体耐受生成检测器, 为了能够在二维

表 1 2-NSA 算法

Table 1 The 2-NSA algorithm

Train: self training set	D : detector set	$r(d_i)$: the radius of detector d_i
C : estimated coverage	C_{exp} : expected coverage	ε : attenuation factor
Step 1 Initialization of the training set Train.		
Step 2 According to training set Train and vertex of n -dimensional hyper-cube, calculate the value of R_{Max} by Eqs. (8) ~ (10), and initialize $D \leftarrow \phi$.		
Step 3 Randomly generate a new candidate detector d_{new} , and calculate the distance between d_{new} and every detector d_i in detector set, goto Step 6 if $\text{dis}(d_{\text{new}}, d_i) \leq r(d_i)$.		
Step 4 The candidate detector pass self-tolerance and becomes a semi-mature detector $d_{\text{new}} \rightarrow d_{\text{semi}}$, and calculate the shortest distance dis_{min} between the semi-mature detector d_{semi} and every self element in training set Train, goto Step 5 if $\text{dis}_{\text{min}} > R_{\text{Max}}$, else do $R_{\text{Max}} \leftarrow R_{\text{Max}} - \varepsilon \cdot r_s$.		
Step 5 If $\text{dis}_{\text{min}} > r_s$, then $d_{\text{mat}} \leftarrow d_{\text{semi}}$, $r(d_{\text{mat}}) = \text{dis}_{\text{min}} - r_s$, $D \leftarrow D \cup \{d_{\text{mat}}\}$.		
Step 6 Calculate current estimated coverage C , goto Step 3 if $C < C_{\text{exp}}$, else stop and return detector set D .

图 1 RNSA, V-Detector, 2-NSA 算法产生检测器过程在二维空间的比较 (生成 5 个成熟检测器, 其中灰色圆圈表示 25 个 Setosa 自体训练集元素, 白色圆圈表示生成的检测器)

Figure 1 The comparison of 5 detectors generation process of RNSA, V-Detector and 2-NSA in 2-dimension (where the gray circles are the 25 Setosa self elements in the training set and the white circles are the detectors)

图形上进行显示, 只选择花萼长度 (sepal length) 和花瓣长度 (petal length) 两个属性进行自体耐受, 所有数据归一化处理到实值 $[0, 1]^2$ 空间, 自体半径 $r_s = 0.05$, RNSA 算法中检测器采用固定半径 $r_d = 0.1$.

图 1 和 2 阐述了 2-NSA 算法的基本思想及其与 RNSA, V-Detector 算法的区别. 如图 1 所示, RNSA 算法以固定半径 $r_d = 0.1$ 生成检测器, V-Detector 算法根据检测器中心与最近自体距离动态生成变半径检测器, 两种算法均会随覆盖率的上升而导致检测器对非自体区域的冗余覆盖, 从而导致该检测器经历不必要的耗时的自体耐受; 而 2-NSA 算法将新生成的候选检测器先与成熟检测器集进行耐受产生半成熟检测器, 只对落在已有成熟检测器集覆盖之外的候选检测器 (半成熟检测器) 进行自体耐受, 从而避免了大量候选检测器不必要的自体耐受. 此外, 如图 2 所示 2-NSA 算法按检测器半径从大到小优先产生覆盖范围更大的检测器, 在达到相同的期望覆盖率 C_{exp} 所需成熟检测器数量显著减小.

图 2 达到期望覆盖率 $C_{exp}=90\%$, RSA, V-Detector 及 2-NSA 算法分别需要生成 58, 17, 4 个成熟检测器 (其中灰色圆圈表示 25 个 Setosa 自体训练集元素, 白色圆圈表示生成的检测器)

Figure 2 When the estimated coverage reaches $C_{exp} = 90\%$, the detector set size of RSA, V-Detector and 2-NSA are 58, 17, and 4 respectively (where the gray circles are the 25 Setosa self elements and the white circles are the detectors)

2-NSA 算法有效避免了当前实值否定选择算法中候选检测器在已有成熟检测器集覆盖范围之内不必要的自体耐受过程, 大量重复覆盖的候选检测器在第一次否定选择过程中被清除, 有效减少了检测器数量, 提高了检测器的生成效率.

3.3 算法性能分析

检测率、误报率和生成检测器集的效率是衡量检测器生成算法有效性的重要因素, 下面对 2-NSA 算法检测率、误报率和算法时间复杂度进行讨论. 设一台计算机中的计算机程序数量为 N_p , 其中一般情况下含有非自体的比例为 ρ , $0 < \rho < 1$, 训练集的大小为 N_s , 检测器集的大小为 N_d , 任意给定的检测器与任意抗原之间的匹配概率为 P_m , $P(X)$ 定义为事件 X 发生的概率^[24].

引理 1 对任一通过自体耐受的成熟检测器, 该检测器识别那些未被描述的自体的概率 P_n 表示为

$$P_n = (1 - P_m)^{N_s} \cdot [1 - (1 - P_m)^{N_p \cdot (1 - \rho) - N_s}]. \quad (11)$$

证明 设事件 A 为“给定的检测与自体集中所有自体都不匹配”, 事件 B 为“给定的检测器与未被描述的自体集中至少一个匹配”. 显然, 事件 A 中的检测为通过自体耐受的检测器, 而事件 B 中的检测为未通过自体耐受的检测器. 根据 P_n 的定义, 有 $P_n = P(A) \cdot P(B)$. 在事件 A 中, 检测器与自体匹配发生的次数 Y 满足二项式分布, 即 $Y \sim b(n, p)$ ^[25], 其中 $n = N_s, p = P_m$. 于是有 $P(A) = P(Y = 0) = (P_m)^0 \cdot (1 - P_m)^{N_s} = (1 - P_m)^{N_s}$, 同理可得 $P(B) = 1 - P(Y = 0) = 1 - (P_m)^0 \cdot (1 - P_m)^{N_p \cdot (1 - \rho) - N_s} = 1 - (1 - P_m)^{N_p \cdot (1 - \rho) - N_s}$, 于是有 $P_n = P(A) \cdot P(B) = (1 - P_m)^{N_s} \cdot [1 - (1 - P_m)^{N_p \cdot (1 - \rho) - N_s}]$, 证毕.

定理 1 对于任意给定的自体抗原 Ag, 检测系统对该抗原发生错误肯定的概率 P_{fp} 表示为:

$$P_{fp} = 1 - (1 - P_m)^{N_d \cdot P_n}. \quad (12)$$

证明 根据引理 1, 设事件 A 为“给定抗原与成熟检测器发生匹配”, 根据误报的定义有 $P_{fp} = P(A)$, 自体抗原与检测器匹配的次数 $Y \sim b(n, p)$ ^[25], 其中 $p = P_m, n = N_d \cdot P_n$ 是识别自体的成熟检测器的数目, 于是 $P_{fp} = P(A) = 1 - P(Y = 0) = 1 - (1 - P_m)^{N_d \cdot P_n}$, 证毕.

引理 2 对任意异常检测系统, 当训练集 Train 足够大时, 该系统的误报率接近于 0%.

证明 根据引理 1 有 $P_n = (1 - P_m)^{N_s} \cdot [1 - (1 - P_m)^{N_p \cdot (1 - \rho) - N_s}]$, 根据定理 1 有 $P_{fp} = 1 - (1 - P_m)^{N_d \cdot P_n}$, 由 Poisson 定理^[24], 当 P_m 很小, N_s 很大时, $P_n \approx [1 - (1 - P_m)^{N_p \cdot (1 - \rho) - N_s}] \cdot e^{P_m \cdot N_s}$, 由于在既定匹配规则下 P_m 是一常数^[5,19], 由极限定理^[26], 当 $N_s \rightarrow \infty, e^{P_m \cdot N_s} \rightarrow 0, P_n \rightarrow 0$, 相应地, 有 $P_{fp} \approx 1 - (1 - P_m)^0 \rightarrow 0$, 证毕.

定理 2 对任意给定的非自体抗原 Ag, 该抗原被正确识别的概率, 即检测率 P_{tp} 为

$$P_{tp} = 1 - (1 - P_m)^{N_d \cdot (1 - P_n)}. \quad (13)$$

证明 设 A 为事件“抗原 Ag 与某个检测器匹配”, 根据检测率的定义可得 $P_{tp} = P(A)$. 在事件 A 中, 抗原和检测器发生匹配的次数 Y 满足二项式分布 $Y \sim b(n, p)^{[26]}$, 其中 $p = P_m, n = N_d \cdot (1 - P_n)$, 于是有 $P_{tp} = P(A) = 1 - P(Y = 0) = 1 - (1 - P_m)^{N_d \cdot (1 - P_n)}$, 证毕.

引理 3 对任意异常检测系统, 当检测器集 D 足够大时, 该系统的检测率接近于 100%.

证明 根据定理 1 有 $P_{tp} = 1 - (1 - P_m)^{N_d \cdot (1 - P_n)}$, 又由 Poisson 定理^[24], 当 P_m 很小, $N_d \cdot (1 - P_n)$ 很大时, $P_{tp} \approx 1 - e^{-P_m \cdot N_d \cdot (1 - P_n)}$, 由极限定理^[26], 当 $N_d \rightarrow \infty, e^{-P_m \cdot N_d \cdot (1 - P_n)} \rightarrow 0$, 故有 $P_{tp} \rightarrow 1$, 证毕.

在既定匹配规则下, 任意给定检测器与任意抗原之间的匹配概率 P_m 是一常数^[5,19], 特别是采用 r -连续匹配位表示的 SNSA, 其中 c 为字符集中字符个数, l 为字符串总长度, r 为连续匹配位数, 有

$$P_m = c^{-r} \cdot \left[\frac{(l - r) \cdot (c - 1)}{c} + 1 \right]. \quad (14)$$

如果字符集采用二进制 $\{0, 1\}$ 表示, 即 $c = 2$, 则有

$$P_m = 2^{-r} \cdot \left[\frac{(l - r)}{2} + 1 \right]. \quad (15)$$

文献 [19] 指出在 RNSA 算法中, P_m 可以通过检测器容量占抗原空间总容量的计算而得, 有

$$P_m = \frac{N_d \cdot V_{\text{detector}}}{V_{\text{cube}}} = \frac{N_d \cdot (r_d)^n \cdot \pi^{n/2}}{\Gamma(n/2 + 1)}. \quad (16)$$

据此, 由于 V-Detector 算法和 2-NSA 算法采用变半径检测器, 可得变半径检测器生成算法的 P_m :

$$P_m = \frac{N_d \cdot V_{\text{detector}}}{V_{\text{cube}}} = \frac{N_d \cdot \prod_{i=1}^n r_{d_i} \cdot \pi^{n/2}}{\Gamma(n/2 + 1)}. \quad (17)$$

根据定理 1 可知, P_{fp} 取决于 N_p, N_s, ρ 和 N_d , 若假定 $N_p = 10^5, P_m = 0.01$, 可得 P_{fp} 与 N_s, ρ, N_d 的关系为 $P_{fp} = 1 - (1 - P_m)^{N_d \cdot (1 - P_m)^{N_s} \cdot [1 - (1 - P_m)^{10^5 \cdot (1 - \rho) - N_s}]}$. 图 3 和 4 是定理 1 和引理 2 的 MATLAB 仿真, 随着 N_d 值的增大, P_{fp} 值也相应增大; P_{fp} 受 ρ 的影响较小; 当 N_s 足够大时, $P_{fp} \rightarrow 0$, 例如, 当 $N_p = 10^5, N_s > 1000$ 时 $P_{fp} < 1\%$, 达到满意的理论值.

根据定理 2 可知, P_{tp} 取决于 N_p, N_s, ρ 和 N_d , 若假定 $N_p = 10^5, P_m = 0.01$, 可得 P_{tp} 与 N_s, ρ, N_d 的关系为 $P_{tp} = 1 - (1 - P_m)^{N_d \cdot [1 - (1 - P_m)^{N_s \cdot [1 - (1 - P_m)^{10^5 \cdot (1 - \rho) - N_s}]}]}$. 图 5 和 6 是定理 2 和引理 3 的 MATLAB 仿真, 随着 N_d 的增大, P_{tp} 值也相应增大; P_{tp} 受 ρ 的影响较小; 当 N_d 足够大时, $P_{tp} \rightarrow 1$, 例如, 当 $N_p = 10^5, N_d > 500$ 时 $P_{tp} > 99\%$, 达到满意的理论值.

图 3 N_S , N_d 对 P_{tp} 的影响, $N_p = 10^5$, $P_m = 0.01$, $\rho = 0.01$

Figure 3 The effect of N_S and N_d on P_{tp} , where $N_p = 10^5$, $P_m = 0.01$, $\rho = 0.01$

图 4 ρ , N_d 对 P_{tp} 的影响, $N_p = 10^5$, $N_s = 100$, $P_m = 0.01$

Figure 4 The effect of ρ and N_d on P_{tp} , where $N_p = 10^5$, $P_m = 0.01$, $N_s = 100$

图 5 N_S , N_d 对 P_{tp} 的影响, $N_p = 10^5$, $P_m = 0.01$, $\rho = 0.01$

Figure 5 The effect of N_S and N_d on P_{tp} , where $N_p = 10^5$, $P_m = 0.01$, $\rho = 0.01$

图 6 ρ , N_d 对 P_{tp} 的影响, $N_p = 10^5$, $N_s = 500$, $P_m = 0.01$

Figure 6 The effect of ρ and N_d on P_{tp} , where $N_p = 10^5$, $P_m = 0.01$, $N_s = 500$

由于 2-NSA 算法的改进, 达到相同覆盖率所需的检测器数量 N_d 明显减少, 基于定理 1 的分析, 可见 N_d 值的减小可以降低 P_{tp} 值, 因此相比较 RNSA 和 V-Detector 算法, 2-NSA 算法降低了系统的误报率。

根据定理 2, $P_{tp} = 1 - (1 - P_m)^{N_d \cdot (1 - P_n)}$, 根据 Poisson 定理 [24], 当 P_m 很小, $N_d \cdot (1 - P_n)$ 很大时, $P_{tp} = 1 - (1 - P_m)^{N_d \cdot (1 - P_n)} \approx 1 - e^{-P_m \cdot N_d \cdot (1 - P_n)}$, 又根据公式 (17) 可得

$$P_{tp} \approx 1 - e^{-\frac{N_d^2 \cdot \prod_{i=1}^n r_{d_i} \cdot (1 - P_n) \cdot \pi^{n/2}}{\Gamma(n/2 + 1)}}. \quad (18)$$

由于相同覆盖率情况下, 2-NSA 算法需要检测器数量 N_d 值更小, 但其相应 $\prod_{i=1}^n r_{d_i}$ 值更大 (因为 $0 < r_{d_i} \leq \sqrt{n} - r_s$, 而 2-NSA 算法中检测器是根据半径从大到小依次产生的, 因此随着检测器数量

增多, r_{d_i} 值不断减小), 所以 2-NSA 算法在 P_{tp} 上相对稳定. 另一方面, 依据检测率的定义, 检测率主要取决于覆盖率, 我们将在第 4 部分通过 UCI 标准数据集实验对比 3 种算法的检测率.

定理 3 RNSA、V-Detector 和 2-NSA 算法的时间复杂度如公式 (19) 所示, 其中 Num 为采样点, C_i 为估计覆盖率, N_s 为训练集大小, N_d 为检测器集大小, P_m 为任意给定的检测器与任意抗原之间的匹配概率.

$$\left\{ \begin{array}{l} \text{RNSA} : O\left(\frac{N_s \cdot N_d}{(1 - P_m)^{N_s}}\right), \\ \text{V-Detector} : O\left(\frac{\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i \cdot N_s \cdot [1 - (1 - P_m)^{N_d}]}{N_d \cdot P_m \cdot (1 - P_m)^{N_s} \cdot (1 - P_m)^{N_d-1}}\right), \\ \text{2-NSA} : O\left(\frac{(\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i + N_d \cdot N_s) \cdot [1 - (1 - P_m)^{N_d}]}{N_d \cdot P_m \cdot (1 - P_m)^{N_s} \cdot (1 - P_m)^{N_d-1}}\right). \end{array} \right. \quad (19)$$

证明 RNSA 算法的主要时间代价就是候选检测器通过自体耐受产生成熟检测器. 设共需 N 个候选检测器进行自体耐受产生大小为 N_d 的检测器集, 即 $N \cdot (1 - P_m)^{N_s} = N_d$, 可得 $N = \frac{N_d}{(1 - P_m)^{N_s}}$. RNSA 算法的主要时间代价就是这 N 个候选检测器的自体耐受过程, 即时间复杂度为 $O(N \cdot N_s) = O(\frac{N_s \cdot N_d}{(1 - P_m)^{N_s}})$. V-Detector 和 2-NSA 算法均采用 Monte Carlo 方法估计成熟检测器集的覆盖率, 两者的区别在于 2-NSA 算法中未通过检测器集耐受的候选检测器不进行自体耐受, 而 V-Detector 算法中所有候选检测器均需进行自体耐受. 在 2-NSA 算法中, 设共需 N_0 个候选检测器进行耐受以产生第 1 个成熟检测器, 由于此时检测器集为空则只需要进行自体耐受, 即有 $N_0 \cdot (1 - P_m)^{N_s} = 1$, 可得 $N_0 = \frac{1}{(1 - P_m)^{N_s}}$; 第 2 个成熟检测器的产生需要经历二次否定选择, 先与已有的第 1 个成熟检测器耐受, 然后进行自体耐受, 设共需 N_1 个候选检测器以产生第 2 个成熟检测器, 即有 $N_1 \cdot (1 - P_m) \cdot (1 - P_m)^{N_s} = 1$, 可得 $N_1 = \frac{1}{(1 - P_m)^{N_s}} \cdot \frac{1}{(1 - P_m)}$; 依次类推, 可知产生第 N_d 个成熟检测器需要的候选检测器数量为 $N_{N_d-1} = \frac{1}{(1 - P_m)^{N_s}} \cdot \frac{1}{(1 - P_m)^{N_d-1}}$. 总共需要的候选检测器数据 $N = \sum_{i=0}^{N_d-1} N_i = \frac{1}{(1 - P_m)^{N_s}} (1 + \frac{1}{(1 - P_m)} + \cdots + \frac{1}{(1 - P_m)^{N_d-1}}) = \frac{[1 - (1 - P_m)^{N_d}]}{P_m \cdot (1 - P_m)^{N_s} \cdot (1 - P_m)^{N_d-1}}$. 2-NSA 算法的时间代价由生成半成熟检测器和成熟检测器的代价构成, 其中生成半成熟检测器的代价与估计覆盖率相关. 根据估计覆盖率的定义, 可得生成半成熟检测器的代价为 $O(\text{Num} \cdot \frac{1}{N_d} \cdot \sum_{i=1}^{N_d} C_i \cdot N)$, 由半成熟检测器生成成熟检测器的代价为: $O(N \cdot N_s)$, 总代价为 $O(\text{Num} \cdot \frac{1}{N_d} \cdot \sum_{i=1}^{N_d} C_i \cdot N + N \cdot N_s)$, 因此可得 2-NSA 算法的时间复杂度为 $O(\frac{(\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i + N_d \cdot N_s) \cdot [1 - (1 - P_m)^{N_d}]}{N_d \cdot P_m \cdot (1 - P_m)^{N_s} \cdot (1 - P_m)^{N_d-1}})$. 相应地, 由于 V-Detector 算法需要所有的候选检测器都进行自体耐受, 故其时间复杂度为 $O(\frac{\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i \cdot N_s \cdot [1 - (1 - P_m)^{N_d}]}{N_d \cdot P_m \cdot (1 - P_m)^{N_s} \cdot (1 - P_m)^{N_d-1}})$, 证毕.

文献 [13, 17, 20] 指出 V-Detector 算法性能远高于 RNSA, 在此对比 V-Detector 和 2-NSA 算法的时间复杂度如公式 (20) 所示. 由于 $N_s \gg 1$, 而 $\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i$ 亦数倍于 N_d , 可知 2-NSA 产生检测器的效率优于 V-Detector 算法.

$$\frac{O(2\text{-NSA})}{O(\text{V-Detector})} = \frac{\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i + N_d \cdot N_s}{\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i \cdot N_s} = \frac{1}{N_s} + \frac{N_d}{\text{Num} \cdot \sum_{i=0}^{N_d-1} C_i}. \quad (20)$$

综合定理 3 的证明和分析, 由于在 2-NSA 算法中, 所增加的第一次否定选择过程 (成熟检测器集耐受) 是在使用 Monte Carlo 方法估计成熟检测器集覆盖率的过程中完成的 (判定覆盖率的同时即进行了成熟检测器的耐受, 2-NSA 算法直接根据此信息进行第二次否定选择过程), 因而本文所提出的 2-NSA 算法相对于 V-Detector 算法并不会增加时间代价, 定理 3 的时间复杂度理论分析也进一步证实了这一点. 此外, 文献 [23] 对结合进化策略的否定选择算法进行了时间复杂度分析, 指出使用进化

表 2 实验参数
Table 2 Experiment parameters

Dataset	Instances	Dimension	Type	Self set	Non-self set	Training set and size	Detection set and size
Iris	150	4	Real	Setosa: 50	Versicolour: 50 Virginica: 50	Setosa:25	Setosa: 25 Versicolour: 25 Virginica: 25
BCW1	683	9	Integer	benign: 444	malicious: 239	benign: 150	benign: 150 malicious: 150
BCW2	569	30	Real	benign: 357	malicious: 212	benign: 150	benign: 150 malicious: 150
Chess	28056	5	Categorical integer	draw: 2796	others: 25260	draw: 1000	draw: 1000 others: 2000

策略改进候选检测器的生成并不能根本上改进算法的时空复杂度. 所以文章中采用随机方式生成候选检测器, 以保持与 RNSA 和 V-Detector 算法的一致性, 更好的对比显示二次否定选择算法的意义所在.

4 实验

本节通过实验验证 2-NSA 算法的有效性. 选用了广泛应用于检测器性能测试和生成效率分析研究^[12,15,18~23]的 Iris、Breast Cancer Wisconsin Original(BCW1)、Breast Cancer Wisconsin Diagnostic(BCW2)¹⁾和 Chess²⁾等 4 个标准数据集进行实验.

实验采用成熟检测器数量(DN)、检测率(TPR)、误报率(FPR)及检测器生成时间(DT)等指标对算法的检测器生成的有效性进行衡量. 由于传统的 RNSA 算法采用预设的检测器数量作为检测器生成的终止条件, 为了保证相同的实验条件下对 3 种否定选择算法进行有效比较, 本文对 RNSA 算法进行了修改, 采用对非自体空间覆盖达到期望覆盖率作为检测器生成的终止条件, 检测器半径 $r_d = 0.1$.

为了验证 2-NSA 算法产生的成熟检测器覆盖能力, 本文进行了对比实验, 使用 Iris, BCW1, BCW2 和 Chess 等 4 个标准数据集进行实验, 并与两种经典的实值否定选择算法 RNSA 和 V-Detector 进行对比. 实验参数如表 2 所示, 数据预处理为实值并归约化在 $[0, 1]$ 空间, 实验重复 100 次取平均值.

实验结果如图 7~10 所示. 图 7 显示了 3 种算法在低维数据集上所需成熟检测器数量的比较, 在随着期望覆盖的上升, RNSA 和 V-Detector 算法需要成熟检测器数量急剧上升, 而 2-NSA 算法则是缓慢增长; 当估计覆盖率为 $C_{\text{exp}} = 99\%$, RNSA 算法需要成熟检测器数量 $DN_{\text{RNSA}} = 10395$, V-Detector 算法 $DN_{\text{V-Detector}} = 374.9$, 而 2-NSA 算法 $DN_{2\text{-NSA}} = 16.14$, 同比减少 99.84% 和 95.69%. 图 8 显示了 3 种算法需要成熟检测器数量随采样点的变化趋势, 随着采样点的增加, 估计覆盖率的估计更接近真实值, 成熟检测器数量略有增加, 但 2-NSA 算法变化极小. 图 9 显示了 V-Detector 和 2-NSA 算法所需检测器数量在更高维数据集 BCW1(9 维)和 BCW2(30 维)的比较 (RNSA 算法完全无法处理高维数据集, 以 BCW2 数据集为例, 由于 RNSA 算法需要的成熟检测器数量在完全没有冗余覆盖的理

1) <http://archive.ics.uci.edu/ml/datasets>

2) 为了适应 RNSA 算法的处理需要, 将原 6 维的 Chess 数据集处理为 5 维, 去除了相关性较小的第 1 维数据

图 7 RNSA, V-Detector 和 2-NSA 算法需要生成的成熟检测器数量在低维数据集上的比较 (采用 Iris 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点 Num=200)

Figure 7 The comparison of detector set size of RNSA, V-Detector and 2-NSA in low-dimensional dataset (where dataset is "Iris", $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200)

图 9 V-Detector 和 2-NSA 算法需要生成的成熟检测器数量在高维数据集上的比较 (采用 BCW1 和 BCW2 数据集, 自体半径 $r_s = 0.05$, 采样点 Num=200)

Figure 9 The comparison of detector set size of V-Detector and 2-NSA in high-dimensional dataset (where dataset are "BCW1" and "BCW2", $r_s = 0.05$, Num=200)

图 8 RNSA, V-Detector 和 2-NSA 算法需要生成的成熟检测器数量随采样点 Num 的变化 (采用 Iris 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 期望覆盖率 $C_{exp}=99\%$)

Figure 8 The comparison of detector set size of RNSA, V-Detector and 2-NSA with different sample points Num (where dataset is "Iris", $r_s = 0.05$, $r_d = 0.1$ in RNSA, $C_{exp} = 99\%$)

图 10 RNSA, V-Detector 和 2-NSA 算法需要生成的成熟检测器数量在大训练集上的比较 (采用 Chess 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点 Num=200)

Figure 10 The comparison of detector set size of RNSA, V-Detector and 2-NSA with big training set (where dataset is "Chess", $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200)

论情况下为: $\frac{1}{V_{\text{Detector}}} \cdot C_{\text{exp}} > \frac{1}{V_{\text{cube}}} \cdot C_{\text{exp}} \approx 5^{30} \cdot C_{\text{exp}}$, 这在当前的计算能力下是完全无法实现的), V-Detector 和 2-NSA 算法达到相应覆盖率所需成熟检测器数量并无明显增长, 2-NSA 相比 V-Detector 算法所需成熟检测器数量有明显减少. 图 10 显示了 3 种算法在更大的训练集下所需成熟检测器数量的比较, 当估计覆盖率为 $C_{\text{exp}} = 99\%$, RNSA 算法需要成熟检测器数量为 $DN_{\text{RNSA}} = 88356$, V-Detector

图 11 RNSA, V-Detector 和 2-NSA 算法检测率和误报率在低维数据集上的比较 (采用 Iris 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点 Num=200)

Figure 11 The comparison of TPR and FPR of RNSA, V-Detector and 2-NSA in low-dimensional dataset, where dataset is "Iris", $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200

图 12 RNSA, V-Detector 和 2-NSA 算法检测率和误报率在较大训练集上的比较 (采用 Chess 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点 Num=200)

Figure 12 The comparison of TPR and FPR of RNSA, V-Detector and 2-NSA with big training set, where dataset is "Chess", $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200

图 13 V-Detector 和 2-NSA 算法检测率和误报率在高维数据集上的比较 (采用 BCW1 数据集, 自体半径 $r_s=0.05$, 采样点 Num=200)

Figure 13 The comparison of TPR and FPR of V-Detector and 2-NSA in high-dimensional dataset (where dataset is "BCW1", $r_s = 0.05$, Num = 200)

图 14 V-Detector 和 2-NSA 算法检测率和误报率在高维数据集上的比较 (采用 BCW2 数据集, 自体半径 $r_s=0.05$, 采样点 Num=200)

Figure 14 The comparison of TPR and FPR of V-Detector and 2-NSA in high-dimensional dataset (where dataset is "BCW2", $r_s = 0.05$, Num = 200)

算法 $DN_{V-Detector} = 1860$, 而 2-NSA 算法 $DN_{2-NSA} = 567.5$, 2-NSA 算法所需成熟检测器数量明显减少. 由此可见, 2-NSA 算法相比 RNSA 和 V-Detector 算法在相同估计覆盖率, 不同维度和训练集规模下所需成熟检测器数量均显著提高, 并且算法并不因数据维度的增加而显著增加成熟检测器数量.

为了验证 2-NSA 算法的检测能力, 本文进行了对比实验, 采用 Iris, BCW1, BCW2 和 Chess 等 4 个标准数据集对比 RNSA, V-Detector 和 2-NSA 算法的检测率 TPR 和误报率 FPR. 实验参数如表 2 所示, 数据预处理为实值并归约化在 $[0, 1]$ 空间, 实验重复 100 次取平均值.

实验结果如图 11 ~ 14 所示. 图 11 显示了 3 种算法在低维数据集 Iris 下检测率和误报率的比较,

图 15 2-NSA 算法在不同维度数据集下生成检测器效率的比较 (自体半径 $r_s=0.05$, 采样点 Num=200)

Figure 15 The comparison of detector generation efficiency of 2-NSA in different dimension dataset (where $r_s = 0.05$, Num = 200)

图 16 RNSA, V-Detector 和 2-NSA 算法生成检测器效率在低维数据集下的比较 (采用 Iris 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点 Num=200)

Figure 16 The comparison of detector generation efficiency of RNSA, V-Detector and 2-NSA in low-dimensional dataset (where dataset is "Iris", $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200)

在估计覆盖率高于 90% 时, 3 种算法的检测率都达到了 99% 以上; 2-NSA 算法的误报率明显低于 RNSA 和 V-Detector 算法, 当估计覆盖率 $C_{\text{exp}} = 99\%$, RNSA 算法的误报率 $\text{FPR}_{\text{RNSA}} = 56.36\%$, V-Detector 算法的误报率 $\text{FPR}_{\text{V-Detector}} = 45.76\%$, 2-NSA 算法的误报率 $\text{FPR}_{2\text{-NSA}} = 22.47\%$, 同比下降 60.13% 和 50.90%。图 12 显示了 3 种算法在更大训练集的 Chess 数据集下检测率和误报率的比较, 检测率相当, 2-NSA 算法误报率有明显降低。图 13 和 14 显示了 V-Detector 和 2-NSA 算法在更高维数据集 BCW1 和 BCW2 下的检测率和误报率的比较, 实验结果表明两种算法检测率相当, 2-NSA 算法误报率有明显下降, 当估计覆盖率 $C_{\text{exp}} = 99\%$, V-Detector 算法的误报率分别为 15.20% 和 8.27%, 2-NSA 算法的误报率分别为 8.47% 和 4.23%。由此可见, 2-NSA 算法相比 RNSA 和 V-Detector 算法, 在保证了解测率的同时, 有效降低了误报率。

为了进一步验证 2-NSA 算法的有效性, 采用 Iris, BCW1, BCW2 和 Chess 等 4 个标准数据集对比 RNSA, V-Detector 和 2-NSA 算法生成检测器的效率。实验参数如表 2 所示, 数据预处理为实值并归约化在 $[0, 1]$ 空间, 实验重复 100 次取平均值。

实验结果如图 15 ~ 18 所示。图 15 显示了 2-NSA 算法产生成熟检测器所耗费时间随数据维度和训练集大小的增加而增长, 但并不会随数据维度的增加而指数增长。图 16 显示了 3 种算法在低维数据集 Iris 下检测器生成效率的比较, 当估计覆盖率 $C_{\text{exp}} = 99\%$ 时, RNSA 算法生成检测器耗时 $\text{DT}_{\text{RNSA}} = 8.8965$ 秒, V-Detector 算法 $\text{DT}_{\text{V-Detector}} = 0.0576$ 秒, 2-NSA 算法 $\text{DT}_{2\text{-NSA}} = 0.0191$ 秒, 同比耗时减少 99.79% 和 66.84%, 2-NSA 算法的检测器生成效率显著提升。图 17 显示了 3 种算法在更大训练集的 Chess 数据集下检测器生成效率的比较, 当估计覆盖率 $C_{\text{exp}} = 99\%$ 时, RNSA 算法生成检测器耗时 $\text{DT}_{\text{RNSA}} = 1126$ 秒, V-Detector 算法 $\text{DT}_{\text{V-Detector}} = 22.12$ 秒, 2-NSA 算法 $\text{DT}_{2\text{-NSA}} = 8.261$ 秒, 同比下降 99.27% 和 62.65%。图 18 显示了 V-Detector 和 2-NSA 算法在更高维的 BCW1 和 BCW2 数据集下检测器生成效率的比较, 2-NSA 算法在高维数据集下效率同样显著提升。由此可见, 2-NSA 算法相比 RNSA 和 V-Detector 算法在相同估计覆盖率, 不同维度和训练集规模下检测器生成效率均

图 17 RNSA, V-Detector 和 2-NSA 算法生成检测器效率在较大训练集的比较 (采用 Chess 数据集, 自体半径 $r_s=0.05$, RNSA 算法 $r_d=0.1$, 采样点Num=200)

Figure 17 The comparison of detector generation efficiency of RNSA, V-Detector and 2-NSA with big training set (where dataset is “Chess”, $r_s = 0.05$, $r_d = 0.1$ in RNSA, Num = 200)

图 18 V-Detector 和 2-NSA 算法生成检测器效率在高维训练集的比较 (采用 BCW1 和 BCW2 数据集, 自体半径 $r_s=0.05$, 采样点Num=200)

Figure 18 The comparison of detector generation efficiency of V-Detector and 2-NSA in high-dimensional dataset (where dataset are “BCW1” and “BCW2”, $r_s = 0.05$, Num = 200)

显著提高, 算法并不因数据维度的增加而显著降低检测器的生成效率。

5 结论

否定选择算法是人工免疫系统生成检测器的重要算法, RNSA 算法采用实值空间编码抗原和抗体生成检测器, V-Detector 算法采用可变半径的 RNSA 生成检测器, 取得了较好的检测效果. 但是, 传统的实值否定选择算法只对自体集进行耐受导致了生成的检测器的重复覆盖, 检测器实际覆盖效率不高导致检测器数量过多, 检测器生成效率过低, 以及误报率过高等问题。

针对上述问题, 本文提出二次否定选择算法, 通过第一次否定选择过程耐受成熟检测器集生成半成熟检测器, 在第二次否定选择过程中半成熟检测器只需要在未覆盖的非自体区域进行训练集的自体耐受, 从而避免了大量候选检测器对已覆盖的非自体区域的自体耐受. 2-NSA 算法通过增加代价较小的检测器集耐受 (第一次否定选择过程) 有效避免候选检测器不必要的耗时的自体耐受, 显著减少了成熟检测器的数量, 降低了算法的计算时间复杂度, 提高了检测器的生成效率并降低了误报率. 理论分析和实验结果表明 2-NSA 相比经典的 RNSA 和 V-Detector 算法具有更好的时间效率和检测器生成质量。

参考文献

- 1 Bretscher P, Cohn M. A theory of self-nonself discrimination, Science, 1970, 169: 1042–1049
- 2 Burnet F. The Clonal Selection Theory of Acquired Immunity. Nashville: Vanderbilt University Press, 1959
- 3 Jerne N. Towards a network theory of the immune system. Annals Immunol, 1974, 125C: 373–389
- 4 Matzinger P. The danger model: a renewed sense of self. Science, 2002, 296: 301–305

- 5 Forrest S, Perelson A, Allen L, et al. Self-nonsel self discrimination in a computer. In: Proceeding of the IEEE Symposium on Research in Security and Privacy, Oakland, 1994. 202–212
- 6 Medzhitov R, Janeway J. Decoding the patterns of self and nonself by the innate immune system. *Science*, 2002, 296: 298–300
- 7 Klarreich E. Inspired by Immunity. *Nature*, 2002, 415: 468–470
- 8 Glickman M, Balthrop J, Forrest S. A machine learning evaluation of an artificial immune system. *Evolut Comput*, 2005, 13: 179–212
- 9 Li T. *Computer Immunology*. Beijing: Publishing House of Electronics Industry, 2004 [李涛. 计算机免疫学. 北京: 电子工业出版社, 2004]
- 10 Li T. An immune based network security risk estimation. *Sci China Ser F-Inf Sci*, 2005, 48: 557–578
- 11 Stibor T, Timmis J, Eckert C. On the appropriateness of negative selection defined over Hamming shape-space as a network intrusion detection system. In: *Proceedings of IEEE Evolutionary Computation*, Edinburgh, 2005. 995–1002
- 12 Li T. Dynamic detection for computer virus based on immune system. *Sci China Ser F-Inf Sci*, 2008, 51: 1475–1486
- 13 Gonzalez F, Dasgupta D. Anomaly detection using real-valued negative selection. *Genet Program Evol Mach*, 2003, 4: 383–403
- 14 Zhou J. *Negative Selection Algorithms: from the Thymus to V-detector*. Ph. D dissertation. Memphis: University of Memphis, 2006
- 15 Zhou J, Dasgupta D. V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage. *Inf Sci*, 2009, 19: 1390–1406
- 16 Gao X, Ovaska S, Wang X. Genetic algorithms-based detector generation in negative selection algorithm. In: *2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, Logan, 2006. 133–137
- 17 Aydin I, Karakose M, Akin E. Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. *Expert Syst Appl*, 2010, 37: 5285–5294
- 18 Bereta M, Burczynski T. Immune K-means and negative selection algorithms for data analysis. *Inf Sci*, 2009, 179: 1407–1425
- 19 Chen W, Liu X, Li T, et al. A negative selection algorithm based on hierarchical clustering of self set and its application in anomaly detection. *Intl J Comput Intell Syst*, 2011, 4: 410–419
- 20 Gong M, Zhang J, Ma J, et al. An efficient negative selection algorithm with further training for anomaly detection. *Knowl Based Syst*, doi: 10.1016/j.knosys.2012.01.004, in press
- 21 Dasgupta D, Yu S, Nino F. Recent advances in artificial immune systems—models and applications. *Appl Soft Comput*, 2011, 11: 1574–1587
- 22 Stibor T, Philipp M, Jonathan T. Is negative selection appropriate for anomaly detection? In: *Proceedings of IEEE Evolutionary Computation*, Edinburgh, 2005. 569–576
- 23 Xu B, Luo W, Pei X, et al. On average time complexity of evolutionary negative selection algorithms for anomaly detection. In: *Proceeding of 2009 World Summit on Genetic and Evolutionary Computation*, Shanghai, 2009
- 24 Chen X. *Probability and Statistics*. Beijing: Science Press, 2000 [陈希孺. 概率论与数理统计. 北京: 科学出版社, 2000]
- 25 Perelson A, Weisbuch G. Immunology for physicists. *Rev Mod Phys*, 1997, 69: 1219–1263
- 26 Chang G Z, Shi J H. *A course in Mathematical Analysis*. Beijing: Higher Education Press, 2003 [常庚哲, 史济怀. 数学分析教程. 北京: 高等教育出版社, 2003]

Dual negative selection algorithm

ZHENG XuFei¹, FANG YongHui^{2*} & LI Tao³

1 *School of Computer and Information Science, Southwest University, Chongqing 400715, China;*

2 *School of Electronic and Information Engineering, Southwest University, Chongqing 400715, China;*

3 College of Computer Science, Sichuan University, Chengdu 610065, China

*E-mail: fyhui@swu.edu.cn

Abstract A negative selection algorithm (NSA) is an important method of generating artificial immune detectors for anomaly detection. However, traditional NSAs aim at eliminating self-recognized invalid detectors by matching randomly generated candidate detectors with the whole self-training set. The matching process of the training set (self-tolerance) contributes the main time cost of such NSAs. The self-tolerance of these NSAs only considers the relationship between candidate detectors and the self-training set, and does not consider the candidate detectors' repetitive coverage with the existing detector set, which leads to unnecessary self-tolerance of candidate detectors and thus an excessive count of detectors and much lower efficiency of detector generation. In this paper, we put forward the dual negative selection algorithm (2-NSA), which includes two negative selection processes. In the first negative selection process, each randomly generated candidate detector first tolerates with the existing detector set and becomes a semi-mature detector when it does not match with any existing mature detector in the existing detector set. In the second negative selection process, the semi-mature detector outside the mature detectors' coverage tolerates with the self-training set and becomes a mature detector when it does not match with any element in the self-training set. The 2-NSA avoids the time-consuming self-tolerance process of the candidate detector within the coverage of existing mature detectors, and thus greatly reduces the size of the detector set and improves detector generation efficiency. Theoretical analysis shows that 2-NSA effectively improves the efficiency of detector generation, reduces the time cost of the algorithm and reduces the false-positive rate of the detection system. The experimental results show that, for the Iris dataset and 99% expected coverage, 2-NSA has 99.84% and 95.69% fewer detectors, a 60.13% and 50.90% lower false-alarm rate, and a 99.79% and 66.84% lower time cost than the classical real-valued NSA and V-Detector algorithm respectively.

Keywords artificial immune system, negative selection algorithm, detector, variable-sized detector, dual negative selection algorithm

ZHENG XuFei was born in 1978. He received his Ph.D. degree in the College of Computer Science in Sichuan University, in 2012. Since 2004, he has been a Lecturer in the School of Computer and Information Science in Southwest University, Chongqing China. He has been actively involved in the research of natural computation and software testing at Southwest University. His current research interests include computer network security, artificial immune system and software testing theory.

FANG YongHui was born in 1978. She received her M.S. degree in computer science from SouthWest China Normal University in 2004. Currently, she is a Ph.D. student at the College of Electrical Engineering of Chongqing University. Since 2002, she has been a Lecturer at the School of Electronic and Information Engineering of Southwest University. Her current research interests include intelligent information processing and computer network security.

LI Tao was born in 1965. He received his Ph.D. degree in computer science from the University of Electronic Science and Technology of China in 1994. From 1994 to 1995, he was a visiting scholar at the University of California in Berkeley studying neural networks theory. He is presently a Professor at the Department of Computer Science of Sichuan University. His current research interests include network security, artificial immune theory and disaster recovery application.