负选择模型中初始检测器集的一个生成算法

吴作顺,窦文华,朱小俊
(国防科技大学计算机学院,湖南长沙 410073)

摘 要: 在介绍人工免疫系统基本概念的基础上,讨论了人工免疫系统中应用广泛的负选择模型.研究的重点是负选择模型中初始检测器集的生成算法,在穷举法的基础上提出了一个新的检测器生成算法,包括算法的设计、性能分析和试验.理论分析与试验结果表明,新生成算法的时间复杂度小于穷举法,随检测器规模成线性递增.在所生成检测器集的性能上新算法也优于穷举法.

关键词: 人工免疫系统; 负选择模型; 检测器生成算法
中图分类号: TP393.08 **文献标识码**: A **文章编号**: 0372-2112 (2003) 05-0687-03

An Algorithm to Generate Detectors in Negative Selection Model

WU Zuo-shun, DOU Wen-hua, ZHU Xiao-jun
(School of Computer Science, Nat'l Univ. of Defense Tech, Changsha, Hunan 410073, China)

Abstract: The Artificial Immune System (AIS) inspired by the natural immune system is introduced. Based on exhaustive algorithm, a new algorithm is presented as detectors generating method in the negative selection model, including its design, performance analysis and experiment. Both mathematical analysis and experiment show that the algorithm runs in time linearly with the size of the detector set. The new algorithm can possibly generate a more powerful detector set than exhaustive algorithm.

Key words: artificial immune system; negative selection model; detector generating algorithm

1 引言

生物体的免疫系统负责抵御外部病原体的入侵.作为一个信息处理系统,免疫系统具有增强学习、免疫记忆、分布式结构等特征[1].研究人员将免疫系统的这些特性应用于解决实际问题,形成了人工免疫系统这个新的研究领域.

迄今为止,已有多个免疫模型被应用来解释免疫现象,模拟免疫系统的体系结构与行为能力.美国 New Mexico 大学的 Forrest 教授等人在自然免疫学中 Self/Nonself 划分的基础上,提出了负选择模型[3],应用于解决异常变化的检测问题,如故障检测与恢复等[4].负选择模型有其生理依据,即 T 细胞的产生过程. T 细胞是免疫细胞的一种.生物体在产生 T 细胞时,首先随机地进行基因重组.所有生成的 T 细胞会在胸腺 (thymus) 中进行负选择,能够对生物体本身基因起反应的 T 细胞都被清除,只有那些对 Self 不敏感的 T 细胞才能够离开胸腺,循环到生物体各个部分,行使免疫功能. Forrest 提出的负选择模型采用类似的工作原理[5]:首先随机地产生候选检测器集,然后在负选择过程中清除那些对自身敏感的检测器,得到的检测器就能够而且只能够对 Nonself 进行免疫反应.

在负选择模型中,初始检测器生成是一个重要的步骤.如何快速有效地生成尽可能多地覆盖 Nonself 空间的候选初始检测器集,从而减少负选择过场中被清除候选检测器数量,对于提高系统性能至关重要.

本文就负选择模型中初始检测器集的生成算法进行研究,在穷举法的基础上提出了一个新的生成算法.

2 问题定义

在负选择模型中,通常采用位串来表述各种细胞类型,包括 Self 细胞、检测器细胞和 Nonself 细胞[5].本文定义 Self 和检测器都为字母表 m 上长度为 l 的位串.为了简化起见,取 $m=2$,即字母表是二进制的. Self 集是未经排序的,可能包含重复的细胞模式.图 1 给出了集合间的相互关系.

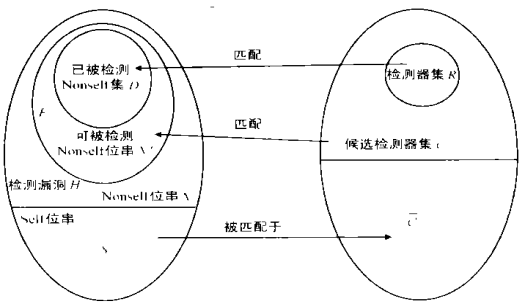

图 1 AIS 空间定义

收稿日期:2002-05-31;修回日期:2002-12-30
基金项目:国家"863"计划专题项目(No. 863-306-ZD07-02-3)

图 1 中有关符号定义为: U:位串空间; U_d:检测器空间. 本文定义 $U_d = U$; F:未被检测到的 Nonself 集($F = N - D$); Q 匹配 P:当且仅当 Q 中任意检测器匹配的位串都在 P 中; P 被匹配于 Q:当且仅当匹配 P 中任意一个位串的检测器都在 Q 中.

在图 1 基础上进行如下定义: P_M:匹配概率. 即随机抽取的位串和检测器能够匹配的概率; P_f:匹配失败概率. 随机选取一个 Nonself 细胞模式,未被 R 中所有检测器匹配的概率; N_x:集合 X 的大小$|N|$. 如 N_S 表示 Self 中位串个数, N_R 表示检测器集的大小;检测规则:采用局部匹配规则. 两个位串只要有 r 个连续位相同,就认为两个位串互相匹配,称作 r 连续位检测规则. 生成算法的目标就是快速有效地产生检测器集 R,匹配 N 中尽可能多的 Nonself 位串,而不匹配 S 中所有的 Self 位串.

3 检测器集生成算法

根据负选择模型的定义,首先随机地产生候选检测器集,然后在负选择过程中清除那些对自身敏感的检测器. 因而检测器生成最直接的方法就是穷举法[3],穷举法非常接近地模拟了自然免疫系统中 T 细胞的发育过程. 穷举法随机地从 U_d 中抽取一个候选检测器,将其与 S 中所有的细胞模式进行匹配. 如果所有的匹配都失败了,该候选检测器成为一个有效的检测器. 穷举法重复这个过程,一直到规定数目的检测器全部产生为止. 由于每个候选检测器都要与 Self 中所有位串进行匹配,穷举法的时间复杂度与初始候选检测器集大小 N_{R0} 的大小以及 Self 空间的大小有关. 穷举法的时间复杂度与空间复杂度分别为 $O(-\ln(P_f)/P_M(1-P_M)^{N_S}\cdot N_S)$ 与 $O(l\cdot N_S)$.

实际上,穷举法生成的检测器在进行负选择过程中大部分都被丢弃,因此其效率非常低,所以有必要针对特定的匹配规则寻求更加有效的生成算法.

当负选择模型采用 r 连续位匹配规则时,可以采用一个二阶段的生成算法. 算法首先求解递归方程,计算出未被 S 中位串匹配的所有位串数,即候选检测器集 C 的规模;第二步,根据递归求解的序号随机生成检测器.

我们先进行如下定义: s 表示一位串; $\hat{s}$ 表示 s 去掉最左边的位后剩余的位串; $s.b$ 表示 s 右边附加位 b,其中 $b \in \{0, 1\}$. $\hat{s}.b$ 表示 s 去掉左边第一位并附加位 b.

定义 r 阶模板(template)为包含 $l-r$ 个未确定位和连续 r 个确定位的长度为 l 的位串. 未确定位用 * 表示. 特别地,定义模板 $t_{i,s}$ 表示模板 t 的 r 连续位从第 i 位开始,且 r 连续位为 s. 定义模板(位串)匹配一个位串必须至少有 r 连续位相同.

模板 t 的右(左)完成(completion)指 t 的右边(左边)未确定位全部确定取值后的位串. $(p, q]$ 表示 $p+1$ 到 q 之间的所有整数.

检测器集合可以按照以下步骤生成:

第 1 步 求解递归方程

对于长度为 r 的位串 s,当 $1 \le i \le (l-r+1)$ 时,定义 $C_i[s]$ 为模板 $t_{i,s}$ 右完成中所有没有匹配 S 中位串的总数.

这些模板实际上列举出了两个位串能够连续 r 位匹配的所有情形. 当 $i = l-r+1$ 时,模板 $t_{i,s}$ 包含 $l-r$ 个未确定位,紧接着是连续 r 个确定位. 此时模板右边没有未确定位,因此模板的右完成就是模板本身. 模板 t 与 S 匹配记作 $match(t, S)$,可以得到 C_{l-r+1} 如下取值:

$$C_{l-r+1}[s] = \begin{cases} 0, & match(t_{l-r+1}, S) \\ 1, & \neg\, match(t_{l-r+1}, S) \end{cases}$$

对于 $1 \le i \le (l-r+1)$,模板的未被匹配右完成数可以根据位置 $i+1$ 的未被匹配右完成数计算. 如果模板 $t_{i,s}$ 匹配 S 中的位串,则 $C_i[s]$ 取 0;否则,可以将模板的右完成分为两类:一类在 r 连续位后附加 0,一类在 r 连续位后附加 1. 也就分别是 $\hat{s}.0$ 和 $\hat{s}.1$ 的右完成数. 得到递归方程如下:

$$C_i[s] = \begin{cases} 0, & match(t_{i,s}, S) \\ C_{i+1}[\hat{s}.0] + C_{i+1}[\hat{s}.1], & \neg\, match(t_{i,s}, S) \end{cases}$$

在算法的第一步,需要进行 $(l-r+1)$ 次循环,每个循环要计算 2^r 个位串 s 的右完成 $C_i[s]$. 计算 $C_i[s]$ 主要取决于 s 与 S 中位串比较需要花费的时间. 算法实现时,将所有匹配 S 中位串的 s 的 $C_i[s]$ 置 0,其他项取 -1.

完成上述预处理后,要进行右完成的递归计算,只需简单地判断 $C_i[s]$ 就够了. 如果 $C_i[s]$ 等于 0,说明 s 匹配 S 中的位串,继续保持为 0;如果 $C_i[s]$ 等于 -1,说明 s 未匹配 S,可以根据递归方程计算其右完成数.

预处理循环需要进行 $\max\{2^r, |S|\}$ 次. 一般情况下,$|S| > 2^r$,所以算法第一步时间代价为 $O(l*|S|)$. 由于算法空间代价较大,实现时可以将 $C[\,.\,]$ 存放在外部存储器中.

第 2 步 生成未被 S 匹配的检测器

随着递归求解过程从右完成数组 C 的 $C_{l-r+1}[\,.\,]$ 列到 $C_1[\,.\,]$ 列,右完成中的未确定位逐渐被填充而确定下来. 对于 $C_1[\,.\,]$,右完成已经是一个所有位都确定的、长为 l 的位串. 因此, $C_1[s]$ 表示由 r 位位串 s 开始的未被 S 匹配的长度为 l 的位串的总数. 据此可以得到未被 S 匹配的所有位串共有:

$$T = \sum_s C_1[s]$$

$C_1[\,.\,]$ 可以看作是根据 r 位位串 s 对未被匹配空间的分割,每个分割空间的大小为 $C_1[s]$. 对于所有由 s 开始的未被匹配位串, $C[\hat{s}.0]$ 在 s 后附加 0, $C[\hat{s}.1]$ 在 s 后附加 1. 因此, $C_2[\,.\,]$ 可以看作是对未被匹配空间的进一步分割. 类似地,从 $C_3[\,.\,]$ 到 $C_{l-r+1}[\,.\,]$ 进行更加详细地分割. 通过 $C_{l-r+1}[\,.\,]$ 分割后,每个分割只是包括一个长度为 l 的位串. 根据这个分割过程,可以给每个未被匹配位串一个序号. 序号范围是从 1 到 T,根据位串本身的大小排序分配.

反过来,根据分配的序号可以查找相应的位串. 对于 $k \in \{1\,..\,T\}$,可以按如下方法查找到第 k 个未被匹配位串 u_k.

首先在 $C_1[\,.\,]$ 进行二叉搜索,查找得到 s_1 满足:

$$P_1 = \sum_{s < s_1} C_1[s] < k < Q_1 = \sum_{s \le s_1} C_1[s]$$

$(P_1, Q_1]$ 里所有位串都由 r 位位串 s_1 开始. 要查找的位串 u_k 在由 $(P_1+1)\cdots Q_1$ 标识的未被匹配空间分割中,因此 u_k 的开始 r 位位串为 s_1.

接着可以逐步判断 u_k 的第 $(r+i-1)$ 位,其中 $i=2\cdots(l-r+1)$.判断的依据是 k 所在的区间.例如,要确定 u_k 的第 $r+1$ 位,将区间 $(P_1, Q_1]$ 进一步划分为 $(P_1, P_1+C_2[s.0]]$ 和 $(P_1+C_2[s.0], Q_1]$,分别对应于第 $r+1$ 位取 0 和取 1.如果 k 落在第一个区间,则 u_k 的第 $r+1$ 位取 0,否则取 1.可以根据下面递归方式确定 P_2 和 Q_2:

$$Q_i=\begin{cases}P_{i-1}+C_i[s_{i-1},0], & b_{i-1}=0\\ Q_{i-1}, b_{i-1}=1, & b_{i-1}=1\end{cases}$$

$$P_i=\begin{cases}P_{i-1}, & b_{i-1}=0\\ P_{i-1}+C_i[s_{i-1}.0], & b_{i-1}=1\end{cases}$$

取 $s_2=s_1.b_1$,此时 k 落在 $(P_2, Q_2]$ 中.继续将 $(P_2, Q_2]$ 划分为 $(P_2, P_2+C_3[s.0]]$ 和 $(P_2+C_3[s.0], Q_2]$,根据 k 所在的区间可以确定 b_2 的取值.位串 u_k 所有剩下的未确定位采用类似方法都可以得到求解.

生成算法对空间的需求较大,需一个 $(l-r)\times 2^r$ 维数组 C 来存储两个位串可能 r 连续位匹配的所有情形.算法的时间和空间代价分别为:时间复杂度:$O((l-r)\cdot N_s)+O((l-r)\cdot 2^r)+O(l\cdot N_R)$;空间复杂度:$O((l-r)^2\cdot 2^r)$

当 l 和 r 取值确定时,新算法的时间开销与 Self 集和检测器集的大小成正比.而穷举法的时间开销则是随 Self 集指数递增,但穷举法的空间代价相对固定.新算法的时间和空间代价与匹配规则 r 呈指数关系,当位串长度 l 和匹配长度 r 增大时,算法开销指数增加.

4　试验分析

我们在二进制位串的基础上进行试验,分别采用穷举法与新算法生成初始检测器集,比较算法的性能以及所生成的检测器集的效率.

试验过程如下(括号中取值为理论计算值,根据文献[2]计算得出):(1)取 $P_f=0.1$;(2)取 $m=2$, $l=32$; $r=8$;计算 P_M $(=0.05)$;(3)根据 P_f 和 P_M,计算得到 N_R $(=46)$;(4)重复下面过程 1000 次:(a)随机生成 Self 集 S,包含 N_S 个随机二进制位串;(b)采用相应的生成算法,生成初始检测器集 R,包含 N_R 个有效检测器;记录时间开销;(c)测试检测器集的效率:(i)随机替换 S 中某个位串;(ii)将改变后的位串与 R 进行匹配;(iii)如果匹配成功,则认为检测器集有效;(5)计算 1000 次试验平均的 P_f.

表 1　穷举法与新生成算法性能比较

N_s	新生成算法		穷举法	
	算法时间(ms)	P_f	算法时间(ms)	P_f
8	13	0.099	2	0.100
16	19	0.107	4	0.088
32	31	0.105	11	0.098
40	37	0.101	18	0.102
56	48	0.105	45	0.111
72	58	0.114	106	0.136
96	73	0.136	377	0.141
128	90	0.144	1989	0.151

根据上述试验步骤,我们在赛扬 300A、64 兆 RAM、Win98SE 上进行了验证试验,编程环境为 VC++6.0.试验结果与理论分析结果基本一致.穷举法与新的生成算法的性能参数如表 1 所示.

试验结果表明,随着 Self 集合规模增大,穷举法所耗费的时间成指数递增,并且检测器集合的性能下降(P_f 增大).新生成算法时间复杂度随着 Self 集规模的增加成线性递加,生成的检测器集合性能则同样有所下降.

比较穷举法和新生成算法,当 N_s 较大时,新算法时间代价明显优于穷举法,生成的初始检测器集性能较穷举法也有一定提高,P_f 值小于穷举法下的对应值.

5　结论

在人工免疫系统的负选择模型中,初始检测器集的生成速度与效率对模型的应用至关重要.穷举法最直接地模拟了生物体内 T 细胞的发育过程,但生成初始检测器集的时间代价随着检测器规模与 Self 集规模成指数增长,因此很难应用于大规模数据集的处理.本文对负选择模型中初始检测器集的生成算法进行研究,在穷举法的基础上提出了一个新的生成算法.与穷举法相比,新生成算法的时间代价比穷举法大为降低,只需要线性的时间开销,新算法生成的检测器集在性能上也比穷举法高.新算法的空间开销较大,但算法所占用空间不被重复利用,只是用于生成检测器集合,完成后就可以释放,因而不会导致整个系统的资源需求增加.

参考文献:

[1] Dasgupta D. An Overview of Artificial Immune Systems and Their Applications [M]. Berlin: Springer-Verlag, 1999.

[2] J K Percus, O E Percus, A S Perelson. Probability of Sel-Nonself Discrimination [C]. A S Perelson and G Weisbbuch, ed. Theoritical and Experimental Insights into Immunology, NY: Springer-Verlag, 1992. 183 - 197.

[3] P D'haeseleer, S Forrest, P Helman. An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications [C]. Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy, 1996.

[4] R J De Boer, A S Pererson. How Diverse Should the Immune System Be [C]. Proceedings of the Royal Society London B, v. 252. London: Biol. Sci, 1993. 171 - 175.

[5] S Forrest, A S Perelson, L Allen R, et al. Self-Nonself Discrimination in a Computer [C]. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Computer Society Press, 1994.

作者简介:

吴作顺　男,1974 年生于湖北省通山县,长沙国防科技大学计算机学院 99 级博士生,主要研究方向为分布式系统与免疫计算,现从事人工免疫算法、计算机网络安全的研究工作,曾参与 863 的课题研究和多项国防预研任务. Email: Wuzuoshun@lycos.com