ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 30,No. 6,Nov. 2015,pp. 1310-1317
DOI:10.16337/j.1004-9037.2015.06.021

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:+86-025-84892742

一种新的连续动作集学习自动机

刘 晓 毛 宁

(中航工业西安航空计算技术研究所,西安,710065)

摘 要:学习自动机(Learning automaton,LA)是一种自适应决策器。其通过与一个随机环境不断交互学习从一个允许的动作集里选择最优的动作。在大多数传统的LA模型中,动作集总是被取作有限的。因此,对于连续参数学习问题,需要将动作空间离散化,并且学习的精度取决于离散化的粒度。本文提出一种新的连续动作集学习自动机(Continuous action-set learning automaton,CALA),其动作集为一个可变区间,同时按照均匀分布方式选择输出动作。学习算法利用来自环境的二值反馈信号对动作区间的端点进行自适应更新。通过一个多模态学习问题的仿真实验,演示了新算法相对于3种现有CALA算法的优越性。

关键词:机器学习;强化学习;在线学习;学习自动机;连续动作集学习自动机

中图分类号:TP181;TP202.7;O234 **文献标志码**:A

New Continuous Action-set Learning Automaton

Liu Xiao, Mao Ning

(Xi'an Aeronautics Computing Technique Research Institute,AVIC,Xi'an,710065,China)

Abstract: Learning automaton (LA) is an adaptive decision maker that learns to choose the optimal action from a set of allowable actions through repeated interactions with a random environment. In most of the traditional LA, the action set is always taken to be finite. Hence, for continuous parameter learning problems, the action space needs to be discretized, and the accuracy of the solutions depends on the level of the discretization. A new continuous action-set learning automaton (CALA) is proposed. The action set of the automaton is a variable interval, and actions are selected according to a uniform distribution over this interval. The end-points of the interval are updated using the binary feedback signal from the environment. Simulation results with a multi-modal learning problem experiment demonstrate the superiority of the new algorithm over three existing CALA algorithms.

Key words: machine learning; reinforcement learning; online learning; learning automata; continuous action-set learning automata

引 言

学习自动机(Learning automaton, LA)是一种可应用于未知的、随机环境的自适应决策单元[1-2]。在任一时刻,LA根据某种概率分布从其动作集里选择一个动作,并输出给环境;环境则反馈回一个强

收稿日期:2014-06-05;**修订日期**:2014-06-30

化信号,作为对所选动作的评价。根据环境的评价,LA 对其概率分布进行调整,以使表现好的动作的被选概率逐渐增大。与有教师指导的监督学习不同,LA 采用的是一种强化学习。其中环境并不直接告诉自动机应该选择哪个动作,而只是对自动机选出的每个动作给出一个好或不好的评价,并且这种评价通常都带有一定的不确定性或随机噪声。作为一种有效的机器学习方法,LA 已经在许多领域得到了应用,如汽车悬挂控制[3]、数字滤波器设计[4]、噪声容忍模式分类[5]、自适应网页爬取[6]、智能电网中的电源管理[7]、无线传感器网络中的覆盖算法[8-9]以及糖尿病病人最佳胰岛素剂量的确定[10]等。

根据动作集的性质,LA 可分为两大类[2]:有限动作集学习自动机(Finite action-set learning automata,FALA)和连续动作集学习自动机(Continuous action-set learning automata,CALA)。FALA 的动作数是有限的,CALA 则可以从一个连续区间甚至整个实数轴上选取动作。对于实值参数学习问题,若采用 FALA,首先要对动作空间进行离散化。离散化的颗粒度太粗,不能保证结果的精度;太细又会导致动作数过多,学习速度减慢。LA 也可以根据环境反馈的强化信号来分类[1]:若强化信号只有两种取值(如用 0 表示成功,1 表示失败),就称为 P 型的;若强化信号取值多于两种但又是有限的,则称为 Q 型的;如果强化信号可取连续的实数值,则称作 S 型的。上述 LA 从环境接收的信息只有强化信号。还有一类 LA,除强化信号外还可以接收来自环境的状态信息(称为情境输入)。这时,LA 的任务是为每一种情境输入选择最适合的输出动作。这种 LA 被称作联想型 LA 或者广义学习自动机(Generalized learning automata,GLA)[2]。本文将要研究的是一种 P 型的、非联想型 CALA。

在 FALA 中,动作概率的表示只需一个维数与动作数相同的矢量即可。但对于 CALA,由于有无穷多个动作,动作概率如何表示就成为一个很棘手的问题。故相对于 FALA,CALA 的研究要困难得多。Gullapalli[11]和 Vasilakos 等[12]分别提出了可产生实值动作的联想型 CALA。这类带有情境输入的 GLA,主要用于随机神经网络的学习。目前已知的非联想型的 CALA 只有以下几种:由 Santharam,Sastry 和 Thathachar 提出的模型[13](为区分起见,以下简记为 CALA-SST,其中的后缀代表 3 位提出者名字的首字母),Beigy 和 Meybodi 提出的模型[14](以下简记为 CALA-BM),Vlachogiannis 提出的模型[15](原文作者称其为 R-CALA),以及由 Frost,Howell,Gordon 和吴青华提出的连续动作强化学习自动机(Continuous action-set reinforcement learning automata,CARLA)[3]。这些 CALA 都是 S 型的,其中的环境反馈都可以取连续的实数值。

CALA-SST,CALA-BM 和 R-CALA 都采用高斯分布作为动作选择的概率模型。R-CALA 实际上是一种截断高斯分布,因为其动作被限制在一个有限区间上。CALA-SST 每次要输出两个动作,一个根据高斯分布随机产生,另一个则直接取高斯分布的均值。算法根据这两个动作以及环境对它们的评价信号,对高斯分布的均值 μ 和标准差 σ 进行更新。为防止 σ 减小到 0 甚至出现负值,算法引入一个参数 σ_L,使 σ 不能小于 σ_L。与 CALA-SST 不同,CALA-BM 和 R-CALA 每次只输出一个动作(按照高斯分布随机产生)。CALA-BM 在对高斯分布的均值进行更新时,要求强化信号必须处于 0 到 1 之间(如果不在此区间,需先做归一化处理);R-CALA 采用简单的贪婪策略对分布均值进行更新:当所选动作好于迄今最好的动作时,将该动作作为新的均值,否则原均值保持不变。在 CALA-BM 和 R-CALA 中,高斯分布的标准差直接根据学习步数来计算,其 σ 单调减小并最终趋于 0。与前面几种 CALA 不同,CARLA 采用非参数化的概率模型,其初始概率分布为一个有限区间上的均匀分布。在学习过程中,算法通过一个对称的高斯型邻近函数,将表现好的动作的奖赏"传播"给其相邻的动作。该算法的实现非常复杂,需要大量的数值积分计算。另外,该算法也需要归一化的强化信号,为此需开辟一个参考存储器以保存环境反馈的历史值。

1　本文提出的 CALA

在本文提出的 CALA 模型中,自动机的动作集对应一个连续区间 $[x_L, x_R]$,该区间的位置和长度可

以动态变化。在任一时刻 n,自动机以均匀分布方式从当前区间上随机选择一个动作 x_n 并输出给环境,环境则给出一个二值的评价信号 $\beta_n \in \{0,1\}$,0 表示成功,1 表示失败。根据该评价信号,自动机对其动作区间的两个端点进行调整更新。

算法参数:λ_1 为大于 0 小于 1 的常数,控制区间外扩的幅度;λ_2 为大于 0 小于 1 的常数(应小于 λ_1),控制区间内缩的幅度;θ 为大于 0 小于 1 的常数,控制在强化信号为失败的情况下区间端点调整的幅度;ε 为足够小且大于 0 的常数,控制解的精度,并防止区间长度无限缩小。

初始化:给动作区间的左右端点 x_L 和 x_R 分别赋初值,并置 $n=0$。

学习过程:

(1) 按照下式产生一个动作 x_n,并输出给环境:

$x_n = x_L + r_n(x_R - x_L)$,其中 r_n 为 0 到 1 之间均匀分布的随机数(每次都重新产生)。

(2) 接收环境反馈的强化信号 β_n。

(3) 更新动作区间(等价于更新概率分布):

令 $c_L = x_L + \Delta$,$c_R = x_R - \Delta$,其中 $\Delta = (x_R - x_L)/3$;

当 $\beta_n = 0$ 时:

若 $x_n < c_L$ 则令 $x_L = x_L - \lambda_1(c_L - x_n)$,

否则令 $x_L = x_L + \lambda_2(1 - \varepsilon/\Delta)(x_n - c_L)$;

若 $x_n > c_R$ 则令 $x_R = x_R + \lambda_1(x_n - c_R)$,

否则令 $x_R = x_R - \lambda_2(1 - \varepsilon/\Delta)(c_R - x_n)$;

当 $\beta_n = 1$ 时:

若 $x_n < c_L$ 则令 $x_L = x_L + \theta\lambda_2(1 - \varepsilon/\Delta)(c_L - x_n)$,

否则,若 $x_n > c_R$ 则令 $x_L = x_L - \theta\lambda_1(x_n - c_L)$;

若 $x_n > c_R$ 则令 $x_R = x_R - \theta\lambda_2(1 - \varepsilon/\Delta)(x_n - c_R)$,

否则,若 $x_n < c_L$ 则令 $x_R = x_R + \theta\lambda_1(c_R - x_n)$;

(4) 令 $n = n+1$,转(1)。

上述动作区间更新的基本原理是:先确定区间的左右 1/3 分界点 c_L 和 c_R。然后,根据 β_n 的取值情况和 x_n 落于 3 等分区间的哪一段,对区间的端点进行调整。当 β_n 为成功时,将两个端点均朝 x_n 所在位置的方向移动(奖励 x_n);当 β_n 为失败时,若 x_n 落于中间的 1/3 段,两个端点均保持不变,否则均朝 x_n 所在位置相反一侧的方向移动(惩罚 x_n)。区间端点移动的幅度与 x_n 跟相应分界点之间的距离成正比,具体的比例系数由参数 λ_1,λ_2,θ 和 ε 控制。其中 λ_2 通常可取 $\lambda_1/3$,以使左端点向右、右端点向左的移动(收缩)比左端点向左、右端点向右的移动(扩张)更谨慎一些。θ 的作用是让对失败动作的惩罚比对成功动作的奖励轻一些。

在区间左右移动时,由于两个端点移动的幅度不同,整个区间实际上是扩张的;而当两个端点均向内移动时,区间会收缩。自动机正是通过对其动作区间的自适应调整(可形象地称之为调焦和变焦),以发现和跟踪最好的动作,将其包围在一个长度逐渐缩小的区间的中心。故将该模型称作“聚焦区间学习自动机(Focused interval learning automaton,FILA)。为体现其对成功的动作进行奖励、对失败的动作进行惩罚的“奖罚”式学习的特点,再在 FILA 的后面缀以 RP,记作 FILA/RP。

2 本文算法的分析

(1) FILA/RP 选择动作的方法非常简单,只需产生一个均匀分布随机数,并将其映射到当前的动作区间上即可。CALA-SST,CALA-BM 和 R-CALA 则需要产生高斯分布随机数。对于没有现成的高斯随机数函数的场合,则需要通过均匀分布随机数来生成高斯分布随机数,计算量将明显增加。至于

CARLA,其动作选择需要计算数值积分,时间开销更大。

(2) FILA/RP 对动作区间(对应概率分布)的更新也相当简单,像 CALA-SST 那样只涉及四则运算,再加一些条件判断。而 CALA-BM 和 R-CALA 在计算高斯分布的标准差时,则需要进行开 3 次方根的运算。CARLA 就更加复杂,其对概率密度函数的更新、对概率密度和环境反馈的归一化处理,都需要大量的数值计算。

(3) CARLA 需要以离散化的方式存储其概率密度函数,为计算归一化的强化信号还要保存一个滑动窗口内的历史环境反馈值(典型设置为 500 个)。而 FILA/RP 则只需保存动作区间的两个端点,存储空间花费非常少。

(4) 在 FILA/RP 中,当区间左端点向右、或右端点向左移动时,通过因子$(1-\varepsilon/\Delta)$来调节移动的幅度。在学习开始阶段,区间范围宽,ε/Δ 很小,$(1-\varepsilon/\Delta)$基本等于 1,移动量几乎不受影响;而当区间变得足够窄时,ε/Δ 接近于 1,$(1-\varepsilon/\Delta)$几乎为 0,相应端点的移动量将非常小。这样可防止区间收缩为一个长度为 0 的"点",以保持对环境变化进行跟踪的能力(CALA-SST 通过引入 σ_L 来达到这一点)。而在 CALA-BM 和 R-CALA 中,由于高斯分布的标准差单调减小并最终趋于 0,自动机将无法发现和跟踪环境的动态变化。

(5) FILA 的核心思想是:以一个可变区间作为动作集,按照均匀分布方式产生动作,再根据环境反馈对动作区间的两个端点进行更新。在该框架下,根据环境反馈的性质、是否使用历史信息以及如何使用,可以形成不同的学习算法。本文的 FILA/RP 是一种 P 型 LA,直接利用环境的二值反馈信号对区间端点进行奖罚式更新。对于环境反馈(即待优化的目标函数)取连续值的情况,则可以记忆历史目标函数值,利用其中"最新"且"最好"的动作对区间进行"奖励"式更新,这样可以得到 S 型的学习算法。

3　仿真实验和结果分析

现在考虑这样一个随机学习问题,其中动作参数 x 可在$(-\infty,\infty)$上连续取值。对于每个 x,环境都会给出一个二值的强化信号 $\beta(x)$,其中 $\beta(x)$以 $d(x)$的概率取 0(表示成功),以 $1-d(x)$的概率取 1(表示失败)。不同动作参数的成功概率由下式确定

$$d(x)=0.4\{\exp[-0.15(x+4)^2]+\exp[-0.15(x-4)^2]\}+0.4 \tag{1}$$

这是一个多模态的非线性函数,在点-4 和 4 处各有一个高度约为 0.8 的峰值,在 x 等于 0 处则有一个高度约为 0.47 的"波谷"。在±4 之外、随着 x 绝对值的增大,$d(x)$逐渐减小并趋近其下限 0.4。分别利用 FILA/RP 和 3 种基于参数化概率模型的 CALA,即 CALA-SST,CALA-BM 和 R-CALA 对上述问题进行仿真实验。CALA-SST 的内部参数[5]经反复尝试选取如下的效果相对好的数值:$\lambda=0.008$,$K=0.7$,$\sigma_L=0.02$;高斯分布的初始参数取 $\mu_0=0$,$\sigma_0=5$。关于 CALA-BM,文献[14]指出可根据$\sigma_n=1/[\text{floor}(n/10)]^{1/3}$来计算 n 时刻高斯分布的标准差,其中 floor 为下取整函数。显然,当 $n<10$ 时该式会出现除法溢出。为此,floor 替换为向上取整函数 ceil。但即使这样,算法仍不能较好地工作,原因是 σ_n 衰减过快。因此对该算法再次进行了"改造",即像 CALA-SST 那样,引入一个新的参数 σ_0,并按照$\sigma_n=\sigma_0/[\text{ceil}(n/10)]^{1/3}$计算 n 时刻高斯分布的标准差,使得 σ_n 的衰减速度可以被控制。CALA-BM 的内部参数,学习步长 a 取 0.015,新引入的 σ_0 取 5;高斯分布的初始均值 μ_0 取 0。原始的 R-CALA 算法[15],保存 β 的全程最小值 $\beta_{\min}$,当 $\beta_n<\beta_{\min}$时将对应的 x_n 赋给 μ_n 同时更新 $\beta_{\min}$。由于本文的 β 只有 0 和 1 两种取值,若采用该规则,μ_n 和 $\beta_{\min}$最多只能被更新一次,无法实现学习。故在本研究中,以"≤"取代"<"。R-CALA 只有两个内部参数:动作参数的上、下限 $x_{\max}$和 $x_{\min}$,仿真中分别取 7 和-7。对于 FILA/RP,相关参数设置如下:$\theta=0.1$,$\varepsilon=0.02$,$\lambda_1=3\lambda_2$ 且 $\lambda_2=0.01$;初始动作区间取为$[-5,5]$。每次仿真均运行 20 000 步。图 1 给出的是 4 种算法各 7 次仿真过程中,动作空间的中心(对 FILA/RP 来说就是动作区间的中点,对另外 3 种算法来说则是高斯分布的均值)随仿真时间 n 的演化轨迹。

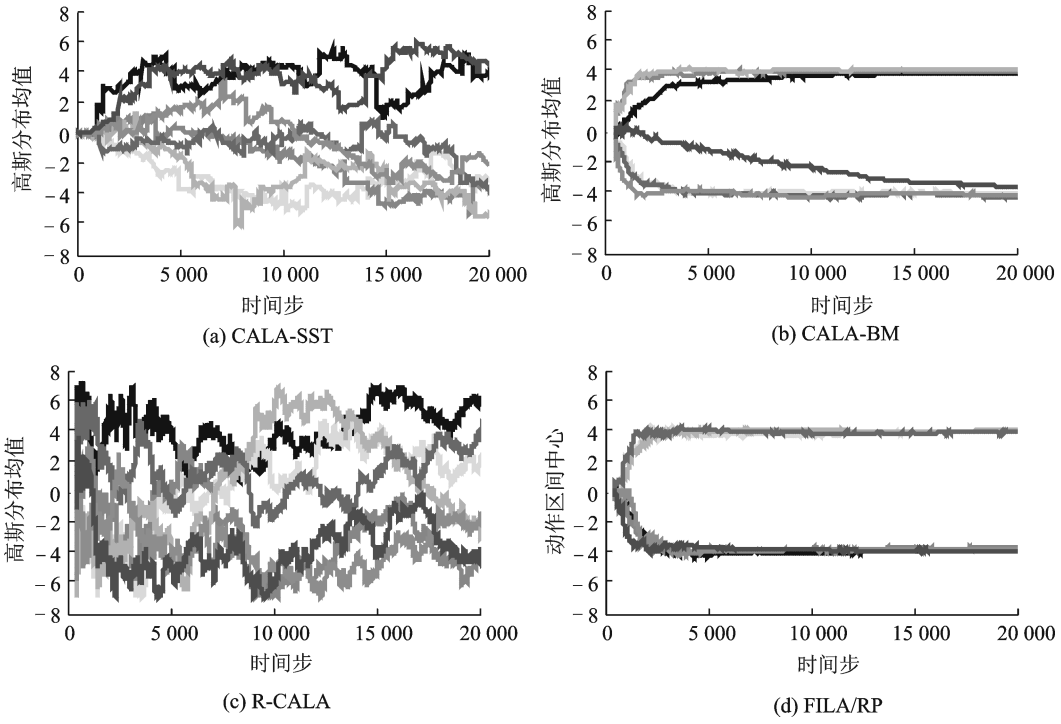

图 1 动作空间的中心随仿真时间的变化轨迹(每种算法各仿真 7 次)

Fig. 1 Center of action space vs. time step(7 simulations per algorithm)

由图 1 可以看出,CALA-SST 和 R-CALA 的高斯分布均值的演化轨迹非常杂乱,到仿真时间结束时($n=20\ 000$),仍有部分 μ_n 未能稳定于最优解 -4 或 4。相比之下,CALA-BM 则好很多,7 次仿真全都能收敛到两个最优解之一。不过,CALA-BM 有一条 μ_n 曲线向目标靠近的速度相当缓慢。这是由于其高斯分布的标准差 σ_n 衰减相对较快,限制了 μ_n 的移动速度。增大新引入的参数 σ_0,可以加快 μ_n 移动的速度,但算法选择动作时的目标将不再集中,自动机选出的 x_n 的波动将会增大。4 种算法中,表现最好的还是 FILA/RP,其动作区间的中心全都能很快移动到两个最优解之一。

图 2 给出的是每种算法一次典型的仿真中,自动机选出的动作序列 x_n 的演化轨迹。在图 2 所示的仿真中,4 种算法基本上都发现了最优动作 -4,但不同算法的运行轨迹却有很大的差异。FILA/RP 和 CALA-BM 均能较快地定位目标,然后坚守,它们找到目标后的动作轨迹都很笔直。不过,CALA-BM 的轨迹要臃肿和粗糙得多,这表明其目标不够集中,选择的动作有较大的波动。减小新引入的参数 σ_0,可以使轨迹变细,但这又会导致 μ_n 的移动变缓(正如图 1(b)中的那条移动缓慢的曲线)。再看 CALA-SST 和 R-CALA,二者的动作轨迹都相当曲折,很难稳定在最优解上。另外,CALA-SST 的动作轨迹上有一些明显的、几乎垂直分布的突跳点,这是由于其高斯分布的标准差变化剧烈,突然间增大,随即又快速变小。相比之下,FILA/RP 的表现最好,其先通过调整动作区间的端点,将最优解包围在区间的中心,然后再使区间迅速收缩,此后则基本上只在该最优解的一个很小的邻域内选取动作。

为评估每种算法的在线学习性能,在仿真过程中的每一步,都计算截止当前自动机实际获得的成功率 $R_n=s_n/n$,其中 s_n 为到 n 时刻的累计成功次数。这样,每做一次仿真,就会得到一条 R_n 曲线。由 $d(x)$ 的定义不难知道,R_n 的理想的变化轨迹是逐渐逼近 0.8(理论最大成功概率)。由于 LA 本质上是一种由随机数序列驱动的随机搜索算法,每次运行不一定能得到相同的结果,故为全面、客观地评估各算法的性能,对每一种算法各做了 200 次仿真。图 3 给出的是每种算法 200 次仿真中 R_n 曲线叠加在一起的效果。

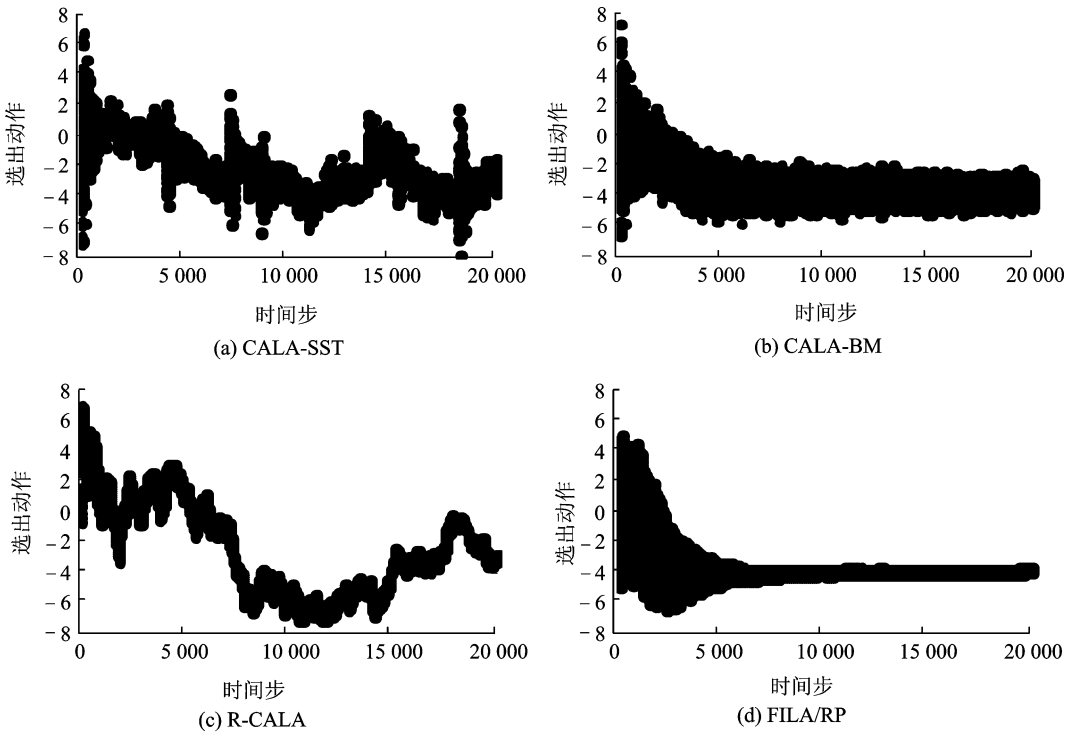

图 2　一次典型仿真中自动机选出的动作随时间的变化轨迹

Fig. 2　Action selected by LA vs. time step in typical simulation

由图 3 可以看出，3 种现有算法的学习轨迹都很散乱（尤其是 CALA-SST 和 R-CALA），这表明它们对随机数序列很敏感，算法每次运行结果的一致性很差。仔细观察不难发现，各算法的行为表现还存在一些细节上的差异。在学习初期，CALA-SST 反应较为迟钝，R_n 上升缓慢。R-CALA 的某些 R_n，从一开始就处于相当高的水平（不过后面并没有进一步上升）；但又有一些 R_n 却几乎一直呆在很低的水平上（CALA-SST 也一样）。到仿真时间结束时，CALA-SST 和 R-CALA 的 R_n 都非常分散，分布范围很宽。至于改造后的 CALA-BM，有一些仿真相当不错，最好的结果显著好于前两种算法；但另一些则不好，R_n 上升缓慢，其中最差的一个在仿真结束时只到 0.56。相比之下，FILA/RP 的学习轨迹则显得相当紧凑，上升趋势也很清晰，到仿真结束时其成功率全都集中于一个很窄的范围之内。表 1 给出了 4 种算法各 200 次仿真（每次仿真 20 000 步）的相关统计结果。

由表 1 不难看出，FILA/RP 的最高、最低和平均成功率在 4 种算法中都是最好的，其最差结果不仅远远好于 3 种现有算法的最差结果，甚至比 CALA-SST 和 R-CALA 的最好结果还要好，快赶上经改造的 CALA-BM 的平均结果了。FILA/RP 性能优异的根本原因，在于其独特的概率模型和模型参数的更新机制。在现有的采用参数化概率模型的 CALA-SST，CALA-BM 和 R-CALA 中，高斯分布的均值和标准差分别进行更新或计算，容易造成分布位置和宽度的脱节，使得均值尚未移动到理想位置、方差就已经很小，或者方差衰减过慢、算法迟迟不能收敛。FILA/RP 对区间端点的更新，则同时实现了区间位置和宽度的调整，其目标的准确度和集中度得到很好的统一。CALA-SST 和 R-CALA 表现不好还有一个原因，二者都基于对环境反馈（β）的比较来更新高斯分布的均值。对于环境反馈只有 0 和 1 两种取值的 P 型环境，这种比较不能给算法提供足够的信息。

在算法的运行时间方面，CALA-SST 耗时最少，其次是 FILA/RP。CALA-BM 和 R-CALA 在确定

高斯分布的标准差时都要计算三次方根,时间花费较长。在本文的仿真中,高斯随机数均通过调用 Matlab 的 randn 函数产生。如果通过自编程序产生,则 3 种现有算法的耗时都将显著加长。

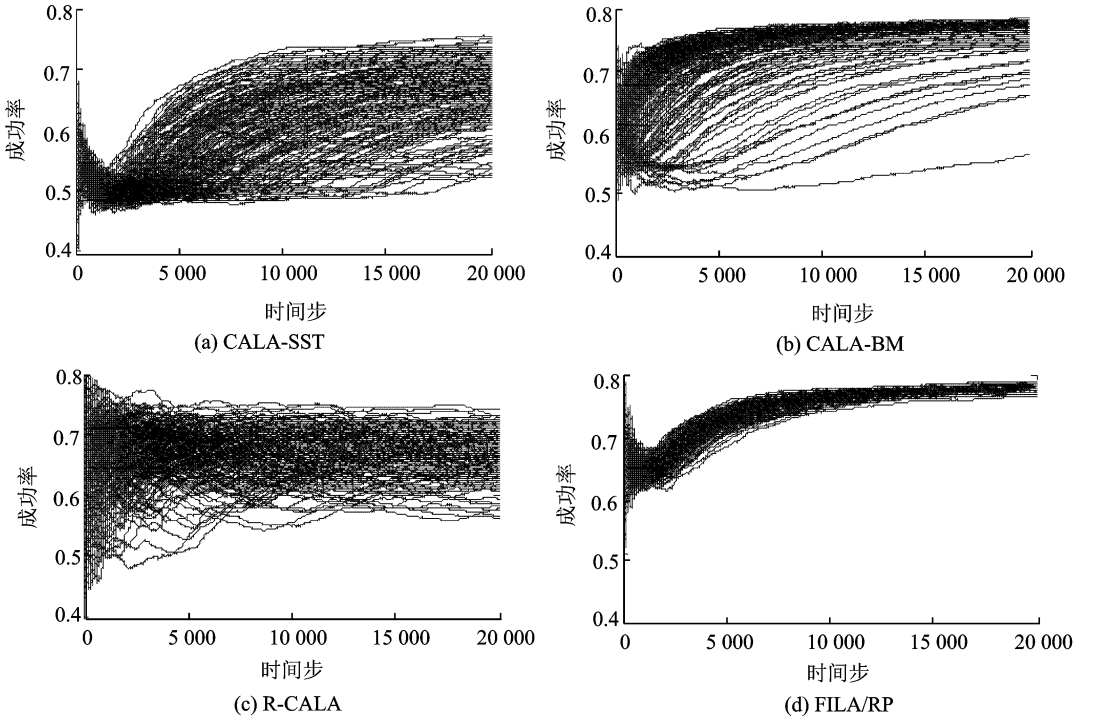

图 3 在线学习性能(每种算法各仿真 200 次)

Fig. 3 Online learning performance(200 simulations per algorithm)

表 1 最终的学习结果

Table 1 Final learning results

学习算法	最高成功率	最低成功率	平均成功率	花费时间/s
CALA-SST	0.756	0.521	0.663	24.67
CALA-BM	0.785	0.560	0.766	28.64
R-CALA	0.742	0.561	0.668	28.76
FILA/RP	0.788	0.763	0.778	27.42

4 结束语

为解决备选动作取实数值的强化学习问题,本文提出一种新的连续动作集学习自动机,即基于奖罚式学习的聚焦区间学习自动机(FILA/RP)。该自动机的动作集为一个实数区间,通过均匀分布方式选择输出动作,并根据环境反馈对区间的端点进行自适应调整。跟现有的 CALA 相比,本文的 FILA/RP 有一些显著的特点和优势。现有的 CALA 都是 S 型的,即环境反馈可以连续取值,FILA/RP 则是 P 型的,因而更适合具有二值反馈的随机环境。FILA/RP 采用一种特殊的参数化概率模型,学习过程中需要存储和更新的参数只有两个(即区间的两个端点),这使得与采用非参数化概率模型的 CARLA 相比,该算法的实现非常简单,时间和空间开销都很小。现有的参数化 CALA 在选择动作时,都需要高斯分

布随机数,新算法则只需要均匀分布随机数,其产生更为容易。由于动作区间不会无限收缩,FILA/RP可以跟踪环境的动态变化;CALA-BM和R-CALA则不能,因为它们的高斯分布的标准差只能单调减小并最终变为0。通过一个多模态的实值参数强化学习问题的仿真实验,演示了新算法的优异性能。与3种现有的参数化CALA相比,本文提出的FILA/RP具有学习速度快、精度高和运行结果的一致性好等优点。目前,已将该算法应用于联想强化学习问题,其效果仍然好于现有的CALA(相关内容将另文发表)。进一步研究的方向,包括用更多、更复杂的问题对该算法的学习性能进行测试和评估,以及尝试将其应用于模式识别、智能控制等实际工程领域。

参考文献:

[1] Narendra K S, Thathachar M A L. Learning automata: An introduction[M]. Englewood Cliffs, NJ: Prentice Hall, 1989: 35-58.

[2] Thathachar M A L, Sastry P S. Varieties of learning automata: An overview[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2002, 32(6): 711-722.

[3] Howell M N, Frost G P, Gordon T J, et al. Continuous action reinforcement learning applied to vehicle suspension control[J]. Mechatronics, 1997, 7(3): 263-276.

[4] Howell M N, Gordon T J. Continuous action reinforcement learning automata and their application to adaptive digital filter design[J]. Engineering Applications of Artificial Intelligence, 2001, 14(5): 549-561.

[5] Sastry P S, Nagendra G D, Mamwani N. A team of continuous-action learning automata for noise-tolerant learning of half-spaces[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2010, 40(1): 19-28.

[6] Torkestani J A. An adaptive focused web crawling algorithm based on learning automata[J]. Applied Intelligence, 2012, 37(4): 586-601.

[7] Misra S, Krishna P V, Saritha V, et al. Learning automata as a utility for power management in smart grids[J]. IEEE Communications Magazine, 2013, 51(1): 98-104.

[8] Mohamadi H, Ismail A S, Salleh S. Solving target coverage problem using cover sets in wireless sensor networks based on learning automata[J]. Wireless Personal Communications, 2014, 75(1): 447-463.

[9] 王建平,陈改霞,孔德川,等. 一种基于学习自动机的WSN区域覆盖算法[J]. 数据采集与处理, 2014, 29(6): 1016-1022.
Wang Jianping, Chen Gaixia, Kong Dechuan, et al. Learning automata-based area coverage algorithm for wireless sensor networks[J]. Journal of Data Acquisition and Processing, 2014, 29(6): 1016-1022.

[10] Torkestani J A, Pisheh E G. A learning automata-based blood glucose regulation mechanism in type 2 diabetes[J]. Control Engineering Practice, 2014, 26: 151-159.

[11] Gullapalli V. A stochastic reinforcement learning algorithm for learning real-valued functions[J]. Neural Networks, 1990, 3: 671-692.

[12] Vasilakos A, Loukas N H. ANASA—A stochastic reinforcement algorithm for real-valued neural computation[J]. IEEE Transactions on Neural Networks, 1996, 7: 830-842.

[13] Santharam G, Sastry P S, Thathachar M A L. Continuous action set learning automata for stochastic optimization[J]. Journal of the Franklin Institute, 1994, 331B(5): 607-628.

[14] Beigy H, Meybodi M R. A new continuous action-set learning automaton for function optimization[J]. Journal of the Franklin Institute, 2006, 343(1): 27-47.

[15] Vlachogiannis J G. Probabilistic constrained load flow considering integration of wind power generation and electric vehicles [J]. IEEE Transactions on Power Systems, 2009, 24(4): 1808-1817.

作者简介:

刘晓(1965-),男,高级工程师,研究方向:进化计算、学习自动机,E-mail: xiao. liu@163. com。

毛宁(1983-),男,高级工程师,研究方向:计算机应用,E-mail: vikermao@163. com。