ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.19, July 2011.

E-mail: eduf@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

连续状态-连续行动强化学习

夏丽丽

(巢湖市科技成果转化服务中心,安徽 巢湖 238000)

摘要:标准的强化学习通常用于解决离散状态空间和行动空间序列决策问题,而很多实际系统的状态和行动为连续变量甚至混合变量,连续状态-连续行动强化学习已经成为该领域研究热点。该文将重点讨论一些将强化学习从离散空间推广到连续空间上的技术或方法,主要从离散化和值函数逼近两方面分析了国内外的研究现状,并介绍了一些常用方法的具体实现。最后,对连续状态-连续行动强化学习未来可能发展方向进行展望。

关键词:强化学习;连续状态-连续行动;离散化;值函数逼近

中图分类号:TP202 **文献标识码**:A **文章编号**:1009-3044(2011)19-4669-04

Reinforcement Learning with Continuous State-continuous Action

XIA Li-li

(Chaohu Scientific and Technological Achievements Service Centre, Chaohu 238000, China)

Abstract: Standard reinforcement learning is commonly applied to sequential decision-problems with discrete states and actions. A wide classes of actual systems are those in which the states and actions of them must be described using continuous or hybrid variables, and reinforcement learning with continuous state-continuous action has become a hot issue of the field. In this paper, some technologies or approaches are mainly discussed to extend reinforcement learning from discrete space to continuous space, and current research situation in the world are analyzed from two sides which are discretization and value function approximation. The actualization of some common solutions is introduction. At last, the future tends of reinforcement learning with continuous state-continuous action are provided.

Key words: reinforcement learning; continuous state-continuous action; discretization; value function approximation

强化学习是机器学习的一个重要分支[1-4],决策主体通过试探与未知环境不断地进行信息交互,利用评价性的反馈信号实现决策的优化。它包括两方面的含义[1],一方面将其看作一类问题,一方面指解决这类问题的技术。近年来,随着强化学习理论的不断发展和完善,已成为一个多学科交叉的研究方向,成功应用于机器人、智能交通、生产调度等应用中[5-8]。

强化学习是解决一类序贯决策问题的有效方法,可以通过统计技术、在线学习、动态规划等估计状态值函数或状态-行动值函数,运用行动值函数确定各个状态的最优行动,形成最优控制策略。强化学习方法通常保存一种表格,每个方格用于存储状态-行动值函数(即 Q 因子),通过样本学习实现值函数更新。因此,该类方法适合解决系统状态空间、行动空间为有限离散的优化控制问题[1]。而生产生活中的许多实际问题的状态或行动为连续变量甚至混合变量,必然要求将该类方法进行推广。目前,解决连续状态-连续行动问题的方法主要包括离散化和值函数逼近两大类[9-12],前者将连续变量转化为离散的[9,11],后者则主要通过参数结构的泛化实现值函数表示[9-10,12]。本文将对连续状态-连续行动强化学习所面临的问题进行分析,对一些求解方法或技术进行讨论,并对未来可能发展方向进行展望。

1 连续状态-连续行动强化学习

1.1 强化学习理论

强化学习方法结合了动态规划、函数随机逼近等思想[1],包括策略,奖惩反馈、值函数和环境模型四个要素。智能主体基于环境模型的系统状态,按照一定的策略确定执行行动,环境模型根据智能主体行动给予相应的奖惩反馈,奖惩反馈通过长期积累构成值函数,其工作过程如下。在 t 时刻(t=1,2,3…表示离散的时间步骤),智能主体感知环境模型状态 $s_t \in S$(S 为系统状态集合),选择执行行动 $a_t \in A(s_t)$作用于环境模型,其中 A(st)为状态 st 的行动集合。环境模型将针对执行行动给予智能主体奖惩反馈 $r_{t+1} \in R$,并转至下一状态 s_{t+1}。所有的状态-行动映射就可以构成系统的控制策略 π,策略 π 下系统的状态转移规律可表示为 $P_{ss'}^{a} = \Pr_{\pi}\left\{s_{t+1} = s' \mid s_t = s, a_t = a\right\}$,其中 s, $s' \in S, a_t \in A(s_t)$。通过智能主体和环境模型的反复交互,获得系统长期的奖惩折扣累积公式:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \cdots \quad (1)$$

其中 γ(0<γ<1)为折扣因子。在策略 π 下,定义状态 s 性能值函数为:

$$V^{\pi}(s) = E\left\{R_t \mid s_t = s\right\} = E_{\pi}\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \mid s_t = s, \pi\right\} \quad (2)$$

定义状态-行动对 Q(s,a)性能值函数为:

收稿日期:2011-05-15
基金项目:国家技术创新试点省工程项目;全国中小企业科技创新基金项目;安徽高校省级自然科学研究重点项目资助
作者简介:夏丽丽(1979-),女,安徽天长人,助理研究员,主要研究方向为计算机应用,人工智能等。

$$Q^{\pi}(s,a)=E\left\{r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\cdots\middle|s_t=s,a_t=a,\pi\right\} \tag{3}$$

其中,状态值函数基于 TD 学习的更新公式为:

$$V(s_t)=V(s_t)+\rho(r_{t+1}+\gamma V(s_{t+1})-V(s_t)) \tag{4}$$

$V(s_t)$表示状态 s_t 下的状态值函数。状态-行动对值函数 $Q(s_t,v(s_t))$学习更新公式为:

$$Q(s_t,a_t)=Q(s_t,a_t)+\rho(r_{t+1}+\gamma\max_{a\in A(s_{t+1})}Q(s_{t+1},a)-Q(s_t,a_t)) \tag{5}$$

1.2 连续状态-连续行动泛化问题

标准强化学习算法利用数据样本实现状态值函数或状态-行动值函数的学习逼近,从而实现最优控制策略的求解。然而,在实际生活中一些系统变量则具有连续性的特点,如状态变量为热量、位置或控制变量为温度、距离等,无法直接利用标准强化学习求解。

为了解决强化学习中连续变量问题,改进算法必须满足鲁棒性、完整性及正确性等要求,需从以下几个方面考虑[13]。首先,模型无关问题。针对离散问题,大部分情况下可以较容易的建立系统数学模型,利用基于模型的数值算法求解。而连续变量问题,一些模型参数则难以获取或计算过于复杂,如系统转移矩阵等,因此改进的强化学习算法必然要不依赖于模型参数。其次,存储问题。在离散情况下采用表格形式存储状态值函数 v(s)或状态行动值函数 Q(s,v(s)),大规模问题则需要大量的空间,存在"维数灾"问题。因此,如何通过有限参数来实现连续状态-连续行动值函数的存储及学习更新问题,是值得探讨的问题。第三,连续性问题,最优控制策略的执行行动必须随着状态的变化而平滑变化,不能出现跳跃的情形。第四,状态和行动的泛化问题,即较好地实现近似状态或行动的泛化,减少在状态空间或行动空间的探索。第五,行动的探索和利用问题。对于连续的空间,如何高效的平衡行动的探索和利用,加快学习进程和提高算法效率是一个重要问题[9,14-16]。

2 离散化和值函数逼近方法

针对连续状态连续行动强化学习算法面临的主要问题,许多专家学者给出了一些有效地解决方法,可以分为两大类:离散化和值函数逼近[9]。离散化方法是利用一定的技术手段把连续空间分解为有限的离散空间[17-18],通过表格形式一一对应存储状态值函数 v(s)或状态-行动对值函数 Q(s,v(s))。而值函数方法则利用一些逼近结构(如神经网络、Tile Code、树等)逼近表示状态值函数 v(s)或状态-行动对值函数 Q(s,v(s)),逼近结构的输入为状态、行动等变量,输出为相应值函数。无论是离散化方法还是值函数逼近方法,都需要考虑的是值函数的学习和存储两大方面问题。

2.1 离散化

离散化方法是解决连续变量值函数问题最为直接的手段,目前存在许多不同的离散化技术,可以根据不同的分类标准进行分类,如监督和非监督、动态和静态、全局与局部、直接与递增等。因此,可以根据研究对象、结构特征等具体问题选择合适的离散化方法。

对于连续状态空间变量问题,离散化方法基本出发点是实现状态聚类,即以一定的标准或规则把若干连续的状态近似为一个状态,将该状态下的决策控制应用于其近似状态中。因此,离散化方法的选择取决于状态性质特征。等间距的均匀离散化方法或非监督离散方法只适应于连续变量状态性质均匀变化的问题,但存在离散粒度难以控制的缺点,粒度过粗造成信息丢失,粒度过细造成状态个数随粒度指数增长"维数灾"问题。因此,产生了一些自适应变间距的离散方法。文献[18]将连续 U 型树技术应用于连续空间离散,该方法就是一种变间距离散技术,采用回归树实现状态值的存储。文献[19]提出了一种基于节点生长 k-均值聚类算法的自适应离散划分方法,该方法使用聚类算法的聚类中心表示离散状态,随着学习的进行而调整聚类中心或添加新聚类中心。文献[20]给出了一种自适应向量量子化方法,不需要预先知道环境信息,通过主体与环境的交互实现连续状态空间的划分修正。文献[21]和[22]则分别提出了基于高斯过程分类器和基于核方法的连续空间的强化学习方法。

对于连续行动问题,离散方法会带来与连续状态空间问题面临的同样难题。另外,连续行动的离散化还存在行动探索和行动利用之间矛盾问题,严重影响着优化算法的收敛速度以及最终优化结果的精度,甚至导致优化算法收敛到局部最优解。因此,强化学习中的连续行动离散化面临更大难题。文献[23]利用"Adaptive Multistage Sampling"技术实现连续行动的离散化,即根据样本数量确定离散化粒度。而文献[24]在文献[11]基础上,将变分辨率的离散化思想应用于连续空间和连续行动 Markov 决策过程优化控制问题。针对联合状体-行动空间,给出了随优化过程进行而变化的分割准则。对于行动的探索和利用矛盾问题,大多数文献都采用的 Boltzamann 方法,文献[14]则提出了自适应行动修正方法,它利用当前状态-行动空间上学习得到的离散、二元决策策略确定增加或减少当前来连续行动值函数。

总之,一般的离散化方法存在泛化能力差、离散粒度难以控制等缺点;而自适应离散化方法固然一定程度上克服了离散粒度选择和变量泛化问题,但离散化带来的状态和行动存储空间"维数灾"、连续状态中任意状态的控制、连续行动中的行动探索和利用矛盾、每个状态下的精确控制等问题依然难以克服。而一些参数化方法基于非线性学习逼近能力强、泛化能力强、结构简单等特点,可有效克服离散化方法的一些缺点。

2.2 值函数逼近

值函数逼近是人工智能领域的一种重要技术[3,10],它运用神经网络、多项式逼近器、树、线性回归机等一些参数化结构逼近表示强化学习中的状态-行动 Q 值函数或状态值函数 ,替代传统方法中的表格形式。该类技术充分发挥神经网络、线性回归机等结构的非线性逼近能力强、泛化能力强、结构简单等特点,有效克服传统方法的不足,即大规模系统中表格存储表示带来的"维数灾"、无法表示连续状态或连续行动值函数等难题。

从值函数逼近的基本思想出发,解决连续状态或行动强化学习可以分为两类方法。第一类方法首先将连续状态或行动转化为

离散状态或行动,然后利用值函数逼近结构实现状态-行动 Q 值函数或状态值函数 的表示。该类方法利用一些逼近结构比较简单,逼近结构的相关参数远远小于状态-行动或状态的个数,从而克服大规模系统中"维数灾"难题。第二类方法利用逼近结构的输入可以为连续变量,直接实现状态-行动 Q 值函数或状态值函数 的逼近表示,从而可以实现任意状态下的精确控制。强化学习的参数化优化分为 Critic、Actor 和 Critic-Actor 三种模式,其中 Critic-Actor 模式是前两种模式的综合[1]。Actor-Critic 的研究最早源于文献[25]和文献[26]等,它是解决强化学习问题最为常见的一种值函数逼近方法,选择一种逼近结构作为 Actor 用于行动选择,而选择另一结构作为 Critic 用于值函数的逼近表示。另外,其与动态规划有效结合形成了一种解决该类问题的新方法-神经元动态规划[13]。本文将介绍一些值函数逼近的常用逼近结构及实现,如 BP 网络、CMAC 等。

2.2.1 BP 网络

BP(Back Propagation)网络是目前使用最为广泛的神经网络,它是由 Rumelhart、McCelland 等科学家在 20 世纪 80 年代提出的[27]。BP 网络在结构上是一种多层感知器或多层前向网络,包括输入层、隐含层(一层或多层)、输出层。它采用梯度下降方法试图最小化网络输出值和目标值之间的误差平方,从输出节点开始反向地向第一隐含层(即最接近输入层的隐含层)传播由总误差引起的权值修正。

Actor-Critic 模式分别采用两个网络进行行动选择和值函数的逼近[1,28-29]。假设 Actor 网络的权值参数为 $\theta=\{\theta_1,\theta_2,\cdots,\theta_n\}$,Critic 网络权值参数为 $w=(w_1,w_2,\cdots,w_m)$,n、m 分别为 Actor、Critic 网络权值个数。Critic 网络输出状态-行动值函数记为 Q_θ,其计算公式为:

$$Q_\theta^w(s,a)=\sum_{i=1}^{m} w_j \varphi_\theta^j(s,a) \tag{6}$$

$\varphi^j_\theta(s,a)$是依赖于 Actor 网络的 Critic 网络输入,$\Phi_\theta=\{\varphi_\theta^1,\varphi_\theta^2,\cdots\varphi_\theta^m\}$。Critic 网络权值更新公式为

$$w_{t+1}=w_t+\beta_t(r_{t+1}+\gamma Q_{\theta_t}^{w_t}(s_{t+1},a_{t+1})-Q_{\theta_t}^{w_t}(s_t,a_t))\Phi_{\theta_t}(s_t,a_t) \tag{7}$$

β_t 为学习步长。Actor 网络权值更新公式为

$$\theta_{t+1}=\theta_t+\rho_t\Gamma(w_t)Q_{\theta_t}^{w_t}(s_{t+1},a_{t+1})\Psi_{\theta_t}(s_{t+1},a_{t+1}) \tag{8}$$

ρ_t 为学习步长,$\Gamma(w_t)$标准化因子,Ψ_θ 与 Φ_θ 关系参见文献[32]。

2.2.2 小脑模型关节控制器

小脑模型关节控制器(Cerebelllar Model Articulation Controller,CMAC)是由 Albus 于 1975 年根据小脑的生物模型提出的一种神经网络[30],是一种基于表格查询式输入输出多维非线性局部逼近神经网络。它具有结构简单、泛化能力强、收敛速度快、容易实现等特点,在函数逼近、信号处理、模式识别及控制等领域有着广泛的应用[31-32]。

CMAC 由输入 X、虚拟层神经元(AC)、实际层神经元(AP)、输出 Y 等构成。它的输入 可以是多维连续的矢量,每个输入矢量被量化,映射到虚拟层神经元,并激活其中的 C 个单元(C 称为网络的泛化参数),被激活的单元输出为 1,未被激活的单元输出为 0。对于输入相近的两个点,则激活的单元将会出现部分重叠,输出也将相近。考虑到记忆单元随着输入空间增大而不断增大,因此一般将虚拟层神经元压缩变换再映射到实际层神经元, 而实际层神经元则存储着不同网络权值。对应于虚拟层神经元中被激活的 C 个单元,实际层神经元也将被激活 C 个单元,该 C 个单元的网络权值加权和即为网络输出 Y。CMAC 网络运用 Widrow-Hoff 规则更新网络权值,其公式为:

$$W_X=W_X+\zeta(y_d-y)/\mathrm{C} \tag{9}$$

其中 $W_X=\{w_{X1},\cdots,w_{Xc}\}$为输入样本 X 对应的 c 个记忆单元的权值向量,c 为 CMAC 的泛化参数,ζ 为学习步长,yd 为期望输出,y 为网络的实际输出。

本文采用 CMAC 网络实现离散状态-连续行动 Q 值函数逼近,网络输入为系统状态或行动,网络输出则为 Q 值函数。每一个系统状态对应一个 CMAC 网络,网络输入记为行动 d,输出为 $Q(i,d,w_i)$(w_i 为状态 i 对应的网络权值向量),其计算公式为

$$Q_\alpha(i,d,w_i)=\sum_{j=h}^{h+C-1} w_{i,j} \tag{10}$$

h 为输入变量 d 所激活存储单元的首地址。参照公式(9),网络权值更新公式为

$$w_{i,j}:=w_{i,j}+\zeta\frac{c_t}{\mathrm{C}} \tag{11}$$

ζ 为学习率,c_t 为 Q 值函数的即时差分值。根据 Sarsa 学习,可以获得 c_t 学习公式为

$$c_t=r_{t+1}+\gamma Q(s_{t+1},a_{t+1},w_{s_{t+1}})-Q(s_t,a_t,w_{s_t}). \tag{12}$$

w_{st} 为状态 s_t 下采用行动 a_t 激活的权值向量。

对于连续状态或连续行动强化学习中的参数化逼近,还包括一些其他不同的结构或方法,如 SVM、tile code、self organizing map 等,这里就不在一一叙述。

3 总结与展望

随着强化学习在许多领域的广泛应用,考虑到各类系统存在的连续或混合状态、行动的实际情况,连续状态-连续行动强化学习方法已成为该领域研究热点之一。在该类方法中,主要研究连续状态-行动值函数的参数化逼近和表示问题,以克服传统离散化方法存在的离散粒度有时难以控制或过细离散化带来的"维数灾"难题,同时解决连续行动空间探索和利用之间的矛盾问题,从而加快算法的学习收敛速度。目前,虽然已经存在了许多连续状态-连续行动强化学习方法,但在解决一些实际问题时还存在局部极值、过度拟合、泛化不够等缺点。因此,参数化逼近结构的选择问题、连续行动空间探索和利用之间的矛盾等问题依然是强化学习领域未来值得探讨和研究的热点之一。

参考文献:

[1] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. MIT Press: Cambridge, MA,1998.

[2] Xu Xin. Sequential anomaly detection based on temporal-difference learning: principles,models and case studies. Applied Soft Computing, 2010, 10(3): 859-867.

[3] Sutton R S. Learning to predict by the methods of temporal differences. Machine Learning,1998(3):9-44.

[4] 高阳,陈世福,陆鑫.强化学习研究综述[J].自动化学报,2004,30(1):86-100.

[5] Mao Z G, Yang L, Malec J. A New Q-learning Algorithm Based on the Metropolis Criterion. IEEE Trans. on Systems, Man, and cybernetics-Part B,2004,34(5): 2140-2143.

[6] Su S, Lee Z, Wang Y. Robust and fast learning for fuzzy cerebellar model articulation controllers. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics,2006,36(1):203-208.

[7] Ernst D, Geurts P, Wehenkel L. Tree-Based Batch Mode Reinforcement Learning. Journal of Machine Learning Research,2006,6(1): 503-556.

[8] 李琼,郭御风, 蒋艳凰.基于强化学习的智能 I/O 调度算法[J].计算机工程与科学,2010,32(7):58-61.

[9] Christopher Kenneth Monson. Reinforcement learning in the joint space: value iteration in worlds with continuous states and actions. Master of Science, Brigham Young University, 2003.

[10] Baird L. Residual algorithm: Reinforcement learning with function approximation. In Proceedings of the Twelfth International Conference on Machine Learning,Morgan Kaufmann, 1995:30-37.

[11] Remi Munos, Andrew Moore Variable Resolution Discretization in Optimal Control. Machine Learning, 2002, 49: 291-323.

[12] Shimon Whiteson, Peter Stone. Evolutionary function approximation for reinforcement learning. Journal of Machine Learning Research, 2006, 7: 877-917.

[13] Bertsekas D P, Tsitsiklis J N. Neuro-Dynamic Programming. Athena Scientific: Belmont, MA, 1996.

[14] Jason P, Lagoudakis Michail G. Learning continuous action control policies. 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2009:169-176.

[15] Millan J R, Posenato D, Dedieu E. Continuous action Q learning. Machine Learning,2002,49:247-265.

[16] Hasselt H V, Wiering M A. Reinforcement Learning in Continuous Action Spaces. IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007:272-279.

[17] Remi Munos, Andrew Moore Variable Resolution Discretization in Optimal Control. Machine Learning,2002,49:291-323.

[18] Uther W T B, Veloso M M. Tree based discretization for continuous state space reinforcement learning. Proceedings of the National Conference on Artificial Intelligence,1998:769-774.

[19] 陈宗海,文锋,聂建斌,等.基于节点生长 k_ 均值聚类算法的强化学习方法[J].计算机研究与发展,2006,43(4):661-666.

[20] Ivan S K Lee, Henry Y K Lau. Adaptive state space partitioning for reinforcement learning. Engineering Applications of Artificial Intelligence, 2004, 17(6): 577-588.

[21] 王雪松, 张依阳, 程玉虎. 基于高斯过程分类器的连续空间强化学习[J].电子学报,2009,37(6):1153-1158.

[22] 何源,张文生.基于核方法的强化学习算法[J].微计算机信息,2008,24(4):243 -245.

[23] Hyeong Soo Chang, Michael C. Fu, Jiaqiao Hu, Steven I. Marcus. An Adaptive Sampling Algorithm for Solving Markov Decision Processes. Operations Research, 2005, 53(1): 126-139.

[24] Monson Christopher K, Wingate David, Seppi Kevin D, Peterson Todd S. Variable resolution discretization in the joint space. Proceedings of the 2004 International Conference on Machine Learning and Applications, pp: 449-455, 2004.

[25] Witten I H. An adaptive optimal controller for discrete-time Markov environments. Information and Control, 1977, 34: 286-295.

[26] Barto A G, Sutton R S, Anderson C. Neuron-like adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics, 1983, 13(5): 834-846.

[27] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. Nature, 1986, 323: 533 - 536.

[28] Konda V R, Borkar V S. Actor-critic - type learning algorithms for Markov decision processes. SIAM Journal on Control and Optimization. 1999, 38 (1): 94-123.

[29] Konda V R, Tsitsiklis J N. Actor-critic algorithms. Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, pp: 1008-1014, 2000.

[30] Albus J S. A new approach to manipulator control: The Cerebellar model articulation controller (CMAC) . Journal of Dynamic Systems, Measurement, and Control Transactions of ASME, 1975,1(9):220-227.

[31] Li Xin, Chen Wei, Chen Mei. Reinforcement Learning Control Based on TWO-CMAC Structure. International Conference on Intelligent Human-Machine Systems and Cybernetics, 2009:116-121.

[32] 高阳,胡景凯,王本年,等.基于 CMAC 网络强化学习的电梯群控调度[J].电子学报,2007,35(2):362-365.