一种合作 Markov 决策系统

雷 莹,许道云
(贵州大学 计算机科学与技术学院,贵州 贵阳 550025)

摘 要:在机器学习中,强化学习是一个重要的研究领域。Markov 决策过程(MDP)是强化学习的重要基础,在一般的 Markov 决策系统中,只考虑一个智能体的学习演化。但目前诸多问题中只考虑单个智能体的学习演化有一定的局限性,越来越多的应用中都涉及到多个智能体。进而引入一种带有两个智能体的联合 Markov 决策系统(CMDP),该系统适用于两个智能体之间合作决策的学习演化。智能体之间存在合作或博弈两种类型,文中重点研究合作类型的 CMDP,在此类学习模型中,智能体交替执行行为,以社会价值作为求优准则,寻找最优策略对 (π_0^*,π_1^*),共同完成目标任务。进一步给出了在联合 Markov 系统中寻找最优策略对的算法,其根本任务是寻找一个最优策略对 (π_0^*,π_1^*),形成一个合作系统 $CMDP^{(\pi_0^*,\pi_1^*)}$,且系统模型可以进一步扩充到多个智能体的联合决策系统。

关键词:强化学习;智能体;联合 Markov 决策过程;最优策略对;算法
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2020)12-0008-07
doi:10.3969/j.issn.1673-629X.2020.12.002

A Cooperation Markov Decision Process System

LEI Ying, XU Dao-yun
(School of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract: Reinforcement learning is an important research area in machine learning. Then Markov decision process (MDP) is the important basics in reinforcement learning. In the usual Markov decision system, only one agent's learning evolution is considered. Among many problems, only the learning evolution of a single agent is considered, which has certain limitations, but the actual application involves multiple agents. For the reason, a cooperation Markov decision process (CMDP) with two agents is introduced, which is suitable for the learning evolution of cooperation decision between two agents. The focuses of the research is the cooperative CMDP. In this kind of learning model, the agent alternately performs behaviors, seeks the optimal pair of strategies (π_0^*,π_1^*) with social value optimization criteria and accomplishes the target tasks together. Researching the algorithm for finding the optimal strategy pair (π_0^*,π_1^*), which is to find an optimal strategy pair and form an evolutionary system $CMDP^{(\pi_0^*,\pi_1^*)}$. In addition, this system model can also be extended to the joint decision system of multiple agents.

Key words: reinforcement learning; agent; cooperation Markov decision process; optimal pair of strategies; algorithm

1 概 述

在机器学习中,依据学习方式不同可以分为三大类[1]:监督学习、无监督学习和强化学习。其中,强化学习[2-4](又称为增强学习)是一个重要的研究领域,其目的是学习环境状态到行动的映射,本质是自主学习并解决负责的决策问题。通过强化学习的方式,智能体[5]能够在探索和开发之间进行折中处理,进而寻找最优策略[6-7]并获得最大回报。Asmuth 等人[8]将寻找最优策略的方法归纳为三类,第一类是基于贝叶斯的方法,第二类是间接选择的方法,第三类是近似效用的方法,其中第一类方法的设计理论较为完善,可用来解决随机决策问题。

在强化学习中,Markov 决策过程系统[9-10](Markov decision process system, MDP)极其重要,从计算模型的演变过程看:Markov 过程(Markov process)(也称为 Markov 系统,Markov process system, MP)是基础模型,引入奖励函数和执行行为,并以状态集到行为集的一个映射作为策略[11],求解 Markov 决策问题。在给定策略 π 下,Markov 决策系统退化到带奖励函数的 Markov 系统 MDP^{π}。一旦策略 π 给定,带

收稿日期:2020-01-09 **修回日期**:2020-05-11
基金项目:国家自然科学基金(61762019,61862051)
作者简介:雷 莹(1992-),女,硕士研究生,CCF 会员(92284G),研究方向为算法设计与分析;通讯作者:许道云(1959-),男,教授,博导,CCF 高级会员(06612S),研究方向为可计算性与计算复杂性。

奖励函数的 Markov 系统的主要任务是演化生成在每一个状态下的期望收集到的奖励值。Markov 决策系统 MDP^{π} 的主要目标是求一个最优策略 π^*,使得在此策略下,在 Markov 链上收集到的期望奖励值最大。

在通常的 Markov 决策系统中,形式上是一个智能体在学习演化的过程。然而,在实际中往往会遇到这样的两类问题:

第一类是多个智能体在同一个环境中共同完成一个目标任务。这类系统称为协同[12]或合作系统[13]。在这类系统中,智能个体单独执行策略行为,智能体之间相互配合、共同完成一个目标任务。其最优策略组合表现为智能体策略向量(或组),奖励值体现为组合执行的社会综合价值。

第二类是多个智能体在同一个环境中相互制约完成各自的目标任务。这类系统称为对抗或博弈系统[14-15]。在这类系统中,智能个体单独执行策略行为,智能体之间相互制约或博弈,各自完成自己的目标任务。其最优策略表现为智能体策略向量(或组),奖励值体现为自身执行的奖励价值。文献[16]提出了多智能体非零和 MDPs(Markov decision processes)对策的增强学习模型和算法,并通过预测对手的策略模型来修正自己的策略。

为方便起见,文中仅考虑两个智能体的联合 Markov 决策系统(CMDP),这样的系统适用于两个智能体之间合作(或对抗)决策的演化学习系统。文中引入了联合 Markov 决策系统的形式定义,在此形式系统中,两个智能体(Alice 和 Bob)交替依据自己的策略执行行为动作,在给定策略对的 Markov 系统(带奖励函数)中演化学习,系统求优是针对策略求优。

文中重点研究合作类型的 CMDP,在此类学习模型中,智能体交替执行行为,以社会价值求优准则,寻找最优策略 (π_0^*,π_1^*),共同完成目标任务。文中给出了寻找最优策略对的算法,其根本任务是寻找一对最优策略 (π_0^*,π_1^*),形成一个演化系统 $CMDP^{(\pi_0^*,\pi_1^*)}$。文章引入的联合 Markov 决策模型适用于两个智能体之间合作执行,系统模型也可以扩充到多个智能体的联合决策系统。

2　Markov 决策系统的进化过程

设 $S=\{s_1,s_2,\cdots,s_n\}$ 为一个有穷状态集(含有 n 个状态),S 上的一个分布用一个函数 $\mu:S\to[0,1]$ 表示,对应一个随机变量 X,其中 $\mu[s]=\Pr_{\mu}[X=s]$。一个非负矩阵 $\boldsymbol{P}=(p_{i,j})_{n\times n}$ 称为一个概率矩阵,如果对每个 $i=1,2,\cdots,n$,$\sum_{j=1}^{n}p_{i,j}=1$。视矩阵 $\boldsymbol{P}$ 中第 i 行 $p_i=(p_{i,1},p_{i,2},\cdots,p_{i,n})$ 为 S 的一个分布,则 $p_{i,j}$ 被认为是从状态 s_i 转移到状态 s_j 的概率。

本节介绍从 Markov 系统模型到 Markov 决策系统的演化过程。

2.1　Markov 系统

设 S 为一个有穷状态集(含有 n 个状态),一个序对 $<S,T>$ 构成一个有限 Markov 系统,其中对每一个 $s\in S$,定义一个概率分布 $T_s:S\to[0,1]$,$T_s(s')$ 表示由状态 s 转移到状态 s' 的概率。

通常,将概率转移函数 T 写成如下形式:

$$T:S\times S\to[0,1],\forall s\in S:\sum_{s'\in S}T(s,s')=1 \tag{1}$$

由一个 Markov 系统 $<S,T>$,可以规定一个 Markov 链:$X_0,X_1,\cdots,X_t,X_{t+1},\cdots$,它是状态集 S 上的一个随机变量序列,满足如下条件:

$$\Pr[X_{t+1}=s_{t+1}\mid X_0=s_0,X_1=s_1,\cdots,X_t=s_t]=\Pr[X_{t+1}=s_{t+1}\mid X_t=s_t]=T(s_t,s_{t+1}) \tag{2}$$

可遍历性[17](ergodicity)是指:如果存在一个稳定分布 π,使得对于 $\forall(x,y)\in S\times S,\lim_{t\to\infty}\Pr[X_t=y\mid X_0=x]=\pi(y)$。即从任意的状态 x 出发,最终稳定在任一状态 y 上的概率由概率分布 π 决定。稳定分布 π 有如下的一个基本性质:

$$\pi P=\pi$$

其中:

$$\pi=(\pi(s_1),\pi(s_2),\cdots,\pi(s_n))$$

$$P=(p_{i,j}),p_{i,j}=T(s_i,s_j)$$

2.2　带状态转移奖励的 Markov 系统

设 $S=\{s_1,s_2,\cdots,s_n\}$,函数 T 可由概率转移矩阵 $\boldsymbol{P}=(p_{i,j})_{n\times n}$ 决定。进一步,引入一个奖励函数 $R:S\times S\to\mathbb{R}$(实数集),$R(s,s')$ 表示由状态 s 转移到状态 s' 所获得的奖励值。类似地,函数 R 写成一个 n 阶方阵 $\boldsymbol{R}=(r_{i,j})_{n\times n}$,并将最大值记为 $R_{\max}$。

递归定义一个(向后)期望收集到的奖励值:

$$V(s)=\sum_{s\in S}T(s,s')[R(s,s')+\gamma\cdot V(s')] \tag{3}$$

其中,$0<\gamma<1$ 称为折扣因子,以保证一个无穷级数收敛。式(3)表明,状态 s 为初始状态,在 Markov 链 $X_0,X_1,\cdots,X_t,X_{t+1},\cdots$ 上得到的奖励值。在式(3)中,项 $R(s,s')+\gamma\cdot V(s')$ 表明:从 s 开始,一步转移到 s' 获得的奖励值加上从 s' 开始收集的期望值打折。

为便于计算,可以将式(3)改写成如下方程:

$$V(s_i)=\sum_{j=1}^{n}T(s_i,s_j)[R(s_i,s_j)+\gamma\cdot V(s_j)]\quad i=1,2,\cdots,n \tag{4}$$

其向量形式写成:

$$\begin{pmatrix}\boldsymbol{V}(s_1)\\ \boldsymbol{V}(s_2)\\ \boldsymbol{M}\\ \boldsymbol{V}(s_n)\end{pmatrix}=\begin{pmatrix}\boldsymbol{E}[\boldsymbol{R}(s_1)]\\ \boldsymbol{E}[\boldsymbol{R}(s_2)]\\ \boldsymbol{M}\\ \boldsymbol{E}[\boldsymbol{R}(s_n)]\end{pmatrix}+\gamma\times\begin{pmatrix}\boldsymbol{V}(s_1)\\ \boldsymbol{V}(s_2)\\ \boldsymbol{M}\\ \boldsymbol{V}(s_n)\end{pmatrix} \tag{5}$$

其中，$E[R(s_i)] = \sum_{j=1}^{n} T(s_i,s_j)R(s_i,s_j), i = 1,2,\cdots,n$ 表示在状态 s_i 下，一步转移所获得的期望奖励值。

在迭代计算过程中，式(4)可以写成如下迭代公式：

$$V_{k+1}(\vec{s}) = \mathrm{diag}(P\boldsymbol{R}^{\mathrm{T}}) + \gamma \cdot PV_k(\vec{s}) \tag{6}$$

其中，$\boldsymbol{R}^{\mathrm{T}}$ 表示矩阵 $\boldsymbol{R}$ 的转置，diag($\boldsymbol{A}$)表示矩阵 $\boldsymbol{A}$ 对角线上元素构成的向量。

在收敛性条件下，$V(\vec{s}) = \lim_{t\to\infty} V_t(\vec{s})$。收敛性保证：在实际计算中，给定一个误差 $\varepsilon > 0$，当演化到 $t+1$ 步时，如果 $\| V_{t+1}(\vec{s}) - V_t(\vec{s}) \| \leqslant \varepsilon$，退出计算，以 $V_{t+1}(\vec{s})$ 近似 $V(\vec{s})$。

例 1：概率矩阵 $\boldsymbol{P}$、奖励函数 $\boldsymbol{R}$、折扣因子取值如下。通过算例观察函数 V 的迭代计算收敛性。

$$\boldsymbol{P} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 2/3 & 0 \\ 0 & 0 & 2/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{R} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$\gamma = 0.9$

初始 V_0 取为零向量(全取 0)，则收敛状况如图 1 所示，其中横坐标表示迭代步数，纵坐标表示 $V(s)$ 收敛值。

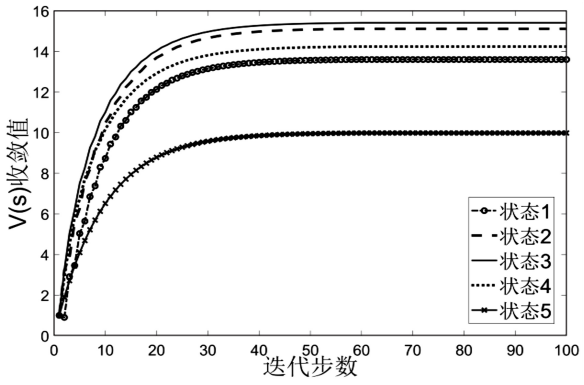

图 1 V -函数收敛示意图

2.3 Markov 决策系统

Markov 决策系统是在带奖励函数的 Markov 系统中引入行为，并将奖励函数修改到行为执行步上。

引入一个有限集 $A = \{a_1,a_2,\cdots,a_m\}$ 作为行为集，对于每一“状态-行为”(s,a) 对，引入 S 上的一个概率分布 $T_{(s,a)}: S \to [0,1]$。同样，将所有分布集中于一个函数 T 中，表示为：$T: S \times A \times S \to [0,1]$，其中，对每一对 (s,a)，$\sum_{s' \in S} T(s,a,s') = 1$。

奖励函数修改为：$R: S \times A \times S \to \mathbb{R}$（实数集）。

引入策略概念，形如 $\pi: S \to A$ 的函数称为一个策略。$\pi(s) = a$ 代表在状态 s 下执行行为 a。对于一个给定的策略 π，可以将式(4)改写为：

$$V^{\pi}(s_i) = \sum_{j=1}^{n} T(s_i,\pi(s_i),s_j)[R(s_i,\pi(s_i),s_j) + \gamma \cdot V^{\pi}(s_j)], i = 1,2,\cdots,n$$

或

$$V^{\pi}(s) = \sum_{s' \in S} T(s,\pi(s),s')[R(s,\pi(s),s') + \gamma \cdot V^{\pi}(s')] \tag{7}$$

式(7)称为 Bellman 递归方程。它是一个不动点方程，V-函数作为一个不动点。可以看出：对于给定的策略，Markov 决策系统是一个带奖励函数的 Markov 系统。

根据函数 V^{π}，将行为作为一个变量，可以诱导出另一种类型的奖励函数：Q-函数，如式(8)所示：

$$Q^{\pi}(s,a) = \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma \cdot V^{\pi}(s')] \tag{8}$$

其中，$Q^{\pi}(s,a)$ 理解为：在指定策略 π 下，在 s 状态下执行行为 a 后期望收集到的奖励值。

Markov 决策系统的主要目标是寻找一个最优策略 π，使函数 V^{π} 最大化。

3 Markov 决策系统中求解最优策略方法

由上一节介绍，一个有限 Markov 决策系统是一个四元组$< S,A,T,R >$，其中 S 是一个有穷的状态集，A 是一个有穷的行为集，$T: S \times A \times S \to [0,1]$ 是概率转移函数，$R: S \times A \times S \to R^{+}$ 是奖励函数。

对于策略 $\pi: S \to A$，由于表示期望收集奖励值的 V -函数和 Q -函数依赖 π，并且由 V^{π} 函数可以决定 Q^{π}。因此，目标是寻找一个策略 π，使函数 V^{π} 最大化。

一个 V -函数 V^{*} 是最优的，如果对于任意策略 π，对于任意状态 $s \in S$，有 $V^{*}(s) \geqslant V^{\pi}(s)$。

一个策略 π^{*} 是最优的，如果对于任意策略 π，对于任意状态 $s \in S$，有 $V^{\pi^{*}}(s) \geqslant V^{\pi}(s)$。

显然，函数 $V^{\pi^{*}}$ 是一个最优 V -函数，其中，对于任意的状态 $s \in S$，取 $V^{*}(s) = V^{\pi^{*}}(s)$。

一个自然的问题是：最优策略 π^{*} 的存在性。其次，如果存在，如何计算得到 π^{*}。理论结果表明[18]：一个有限 Markov 决策系统$< S,A,T,R >$的最优策略存在。

最优 V -函数 V^{*} 与最优策略 π^{*} 有如下关系：

(1)如果已经得到最优函数 V^*,利用如下方法可得到最优策略 π^*:

$$\pi^*(s) = \arg\max_a \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma \cdot V^*(s')] \tag{9}$$

(2)如果已经得到最优策略 π^*,利用如下方法可得到最优函数 V^*:

$$V^*(s) = V^{\pi^*}(s) = \sum_{s' \in S} T(s,\pi^*(s),s')[R(s,\pi^*(s),s') + \gamma \cdot V^{\pi^*}(s')] \tag{10}$$

式(9)和式(10)提供了一个交叉迭代,并求解得到最优策略 π^* 和最优函数 V^*。

在具体计算过程中,用一个给定误差($\varepsilon > 0$)控制给定策略下 Markov 系统演化计算的停机条件,其计算方法及步骤如下:

第 0 步:初始化 V-函数 V_0(可以随机取值,如取 $V_0(s) = 0 (s \in S)$)

贪心计算一个策略 π_0:

$$\pi_0(s) = \arg\max_a \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma \cdot V_0(s')]$$

记 $V_0^{\pi_0} = V_0$;

第 $k+1$ 步:假定已经得到 $\pi_k, V_k^{\pi_k}$,在此步上,做两段计算:

(1)Markov 系统演化计算。

令 $V_0 = V_k^{\pi_k}$ 作为初始 V-函数,在给定误差 $\varepsilon > 0$ 下,作如下迭代计算:

$$V_{m+1}(s) := \sum_{s' \in S} T(s,\pi_k(s),s')[R(s,\pi_k(s),s') + \gamma \cdot V_m(s')]$$

即当 π_k 给定后,用式(7)迭代计算,当 $\| V_{m+1} - V_m \| < \varepsilon$ 时,定义 $(s) = V_{m+1}(s) (s \in S)$。

(2)贪心计算。

$$\pi_{k+1}(s) = \arg\max_a \sum_{s' \in S} T(s,a,s')[R(s,a,s') + \gamma \cdot V_{k+1}^{\pi_k}(s')]$$

算法终止条件:当 $\pi_{k+1} = \pi_k$,终止计算,并定义最优策略 $\pi^* = \pi_k$。

最终,$V_{k+1}^{\pi_k}$ 为最优 V-函数,π_{k+1} 为最优策略。

注:算法终止条件 $\| V_{m+1} - V_m \| < \varepsilon$ 也可以用单点比较达到误差条件:$| V_{m+1}(s) - V_m(s) | < \varepsilon$。上述计算过程可以用图 2 表示。

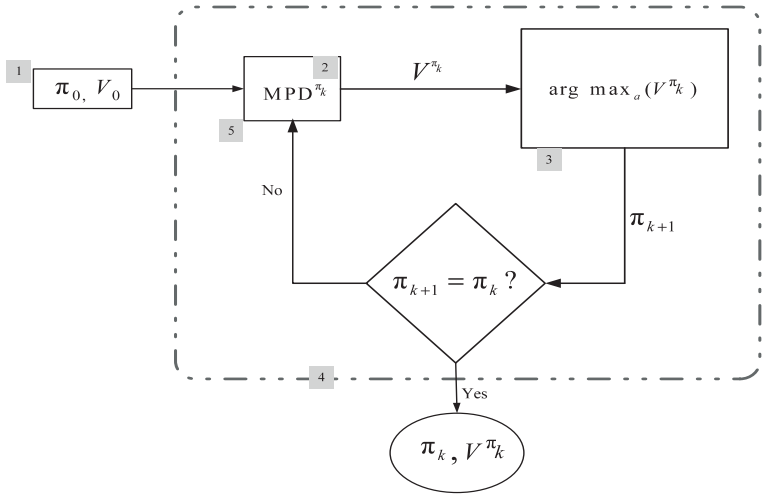

图 2　最优策略求解算法框架

例 2:考虑一个 Markov 决策系统 $< S, A, T, R >$,其中 $S = \{s_0, s_1, s_2\}$,$A = \{a_0, a_1\}$,概率转移函数 T 和奖励函数 R 分别如下:

$$T(S,a_0,S): \begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 0.5 & 0 & 0.5 \\ s_1 & 0.7 & 0.1 & 0.2 \\ s_2 & 0.4 & 0 & 0.6 \end{array},$$

$$R(S,a_0,S): \begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 1 & 0 & 1 \\ s_1 & 5 & 1 & 1 \\ s_2 & 1 & 1 & 1 \end{array}$$

$$T(S,a_1,S): \begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 0 & 0 & 1 \\ s_1 & 0 & 0.95 & 0.05 \\ s_2 & 0.3 & 0.3 & 0.4 \end{array},$$

$$R(S,a_1,S): \begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 0 & 0 & 1 \\ s_1 & 0 & 1 & 1 \\ s_2 & -1 & 1 & 1 \end{array}$$

折扣因子 $\gamma = 0.95$,误差控制取 $\varepsilon = 0.000\,1$。

求得最优策略为:$\pi^* = \begin{pmatrix} s_0 & s_1 & s_2 \\ a_1 & a_0 & a_1 \end{pmatrix}$。

如图 3 所示,描述了所有策略下状态 s_i 对应的 $V(s_i)$ 的收敛值。其中纵坐标表示 $V(s)$ 收敛值。横坐标表示 8($8=2\times2\times2$)种策略,考虑有 s_0、s_1、s_2 三个状态,且每个状态有 a_0、a_1 两种行为,因此横坐标上的每个数值对应一定的二进制码,进而表示相应的策略。表 1 为不同策略不同状态下的具体值。

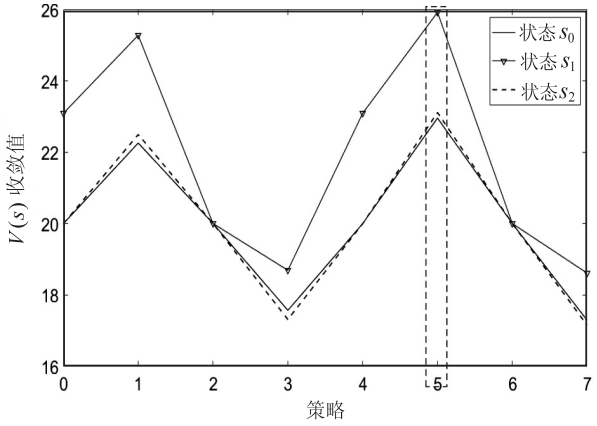

图 3 不同策略下的 V -函数比较

表 1 不同策略的具体值

策略	s_0	s_1	s_2
0	19.998 620 83	23.092 599 44	19.998 660 48
1	22.258 183 99	25.277 260 27	22.496 075 58
2	20.001 056 49	20.001 450 71	20.001 026 11
3	17.570 476 87	18.692 653 42	17.314 558 19
4	19.998 695 62	23.092 666 14	19.998 721 7
5	22.957 876 41	25.921 054 51	23.113 645 88
6	20.000 794 71	20.001 302 99	20.000 778 81
7	17.292 511 69	18.612 460 93	17.149 921 87

例如,图中标注出的 5 的二进制码(101)对应策略 $\begin{pmatrix} s_0 & s_1 & s_2 \\ a_1 & a_0 & a_1 \end{pmatrix}$。

实际计算中,最优策略可以由“迭代+演化”过程得到。此例中得到的最优策略下的概率转移矩阵 $\boldsymbol{P}$ 和奖励函数 $\boldsymbol{R}$ 分别为:

$$\boldsymbol{P}:\begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 0 & 0 & 1 \\ s_1 & 0.7 & 0.1 & 0.2 \\ s_2 & 0.3 & 0.3 & 0.4 \end{array}$$

$$\boldsymbol{R}:\begin{array}{c|ccc} & s_0 & s_1 & s_2 \\ s_0 & 0 & 0 & 1 \\ s_1 & 5 & 1 & 1 \\ s_2 & -1 & 1 & 1 \end{array}$$

4 联合 Markov 决策过程系统(CMDPs)

本节引入两个智能体在同一个 Markov 决策过程中运行的联合形式系统。

4.1 两个 Markov 系统的乘积系统

设 $\boldsymbol{A}=(a_{i,j})_{n\times n}$,$\boldsymbol{B}=(b_{i,j})_{m\times m}$ 为两个矩阵,矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 的张积定义为:$\boldsymbol{A}\otimes\boldsymbol{B}=(a_{i,j}B)$。

例如:设 $\boldsymbol{A}=\begin{pmatrix}1 & -1\\ 0 & 1\end{pmatrix}$,$\boldsymbol{B}=\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}$,则

$$\boldsymbol{A}\otimes\boldsymbol{B}=\begin{pmatrix}B & -B\\ 0 & B\end{pmatrix}=\begin{pmatrix}1 & -1 & -1 & 1\\ -1 & 1 & 1 & -1\\ 0 & 0 & 1 & -1\\ 0 & 0 & -1 & 1\end{pmatrix}$$

容易验证:如果 $\boldsymbol{P}_1$,$\boldsymbol{P}_2$ 均为概率矩阵(不一定同阶),则 $\boldsymbol{P}_1\otimes\boldsymbol{P}_2$ 也为概率矩阵。

设 $S_1=\{s_1^1,s_2^1,\cdots,s_n^1\}$,$S_2=\{s_1^2,s_2^2,\cdots,s_m^2\}$ 为两个状态集,(S_1,P_1),(S_2,P_2) 为两个 Markov 系统,取状态集 $S=S_1\times S_2$,S 的状态概率转移矩阵为 $\boldsymbol{P}=\boldsymbol{P}_1\otimes\boldsymbol{P}_2$,则称 Markov 系统 (S,P) 为 Markov (S_1,P_1) 系统和的 Markov (S_2,P_2) 积系统。

可分离的 Markov 系统:设 (S,P) 为一个 Markov 系统,如果它可以表示成两个 Markov 系统的积系统,则称 (S,P) 是可分离的。

4.2 两个智能体参与的联合 Markov 决策系统

现在考虑:在同一个系统中,具有两个智能体 Agent_0,Agent_1(Alice 和 Bob)分别执行各自的行为动作,以合作方式完成目标任务。对于“合作”类型:两者协同完成同一个目标;对于“对抗”类型:两者之间相互制约(如:博弈)各自完成自己的目标任务。

形式上,这样的系统为如下元组:

$$<S,A,\{\text{Agent}_0,\text{Agent}_1\},\{T_0,T_1\},\{R_0,R_1\}>$$

其中,

S 为一个有穷状态集;

A 为一个行为动作集;

T_i 为 Agent_i 的概率转移函数($i=0,1$),定义为:

$T_i:S\times S\times A\times S\times A\to[0,1]$。

其满足条件:对 $\forall(s_1,s_2,a)\in S\times S\times A$,$\sum_{(s',a')\in S\times A}T_i(s_1,s_2,a,s',a')=1$。换言之,对任意固定的 (s_1,s_2,a),$T_i(s_1,s_2,a,\cdot,\cdot)$ 规定了 $S\times A$ 上的一个概率分布。

直观上,$T_0(s_1,s_2,a,s',a')$ 表示:“当 Agent_0,Agent_1 处于状态对 (s_1,s_2) 时,执行行为动作 a 后,Agent_0 转移到状态 s',合作者(或对抗方) Agent_1 响应执行行为 a'”的概率。

R_i 为 Agent_i 的奖励函数($i=0,1$),定义为:

$R_i:S\times S\times A\times S\times A\to\mathbb{R}$ (实数集)

直观上,$R_0(s_1,s_2,a,s',a')$ 表示:“当 Agent_0,Agent_1 处于状态对 (s_1,s_2) 时,Agent_0 执行行为动作 a

后, $Agent_0$ 转移到状态 s',合作者(或对抗方) $Agent_1$ 响应执行行为 a'"时, $Agent_0$ 获取的奖励值。

5　联合 Markov 决策系统策略迭代方法

文中仅考虑两个智能体合作执行的联合 Markov 决策系统。

形式上,两个智能体 $Agent_0$,$Agent_1$ 对应两个 V -函数: $V_i: S \times S \to \mathbb{R}$ $(i=0,1)$ 。在联合形式下,用社会价值描述联合 V -函数 $V: S \times S \to \mathbb{R}$ 。形式上定义为:

$$\begin{cases} V_0^{k+1}(s,t) = \sum\limits_{s \in S} \sum\limits_{a \in A} T_0(s,t,\pi_0(s,t),s',a)[R_0(s,t,\pi_0(s,t),s',a) + \gamma \cdot V^k(s',t)] \\ V_1^{k+1}(s,t) = \sum\limits_{t \in S} \sum\limits_{a \in A} T_1(s,t,\pi_1(s,t),t',a)[R_1(s,t,\pi_1(s,t),t',a) + \gamma \cdot V^k(s,t')] \\ V^{k+1}(s,t) = \alpha V_0^{k+1}(s,t) + (1-\alpha) V_1^{k+1}(s,t) \end{cases} \tag{11}$$

请注意:在式(11)中,带打折因子的倍乘项是基于 S 上的边缘分布求期望值:

$$\sum_{s \in S}[\sum_{a \in A} T_0(s,t,\pi_0(s,t),s',a)]V^k(s',t)$$

$$\sum_{t \in S}[\sum_{a \in A} T_1(s,t,\pi_1(s,t),t',a)]V^k(s,t')$$

$$\begin{cases} \pi_0'(s,t) = \arg\max_a \sum\limits_{(s',a) \in S \times A} T_0(s,t,a,s',a')[R_0(s,t,a,s',a') + \gamma \cdot V^{(\pi_0,\pi_1)}(s',t)] \\ \pi_1'(s,t) = \arg\max_a \sum\limits_{(t',a) \in S \times A} T_1(s,t,a,t',a')[R_1(s,t,a,t',a') + \gamma \cdot V^{(\pi_0,\pi_1)}(s,t')] \end{cases} \tag{12}$$

为得到合作类型联合 Markov 决策系统中求解最优策略对 (π_0^*,π_1^*) 算法,仿照单个智能体的最优策略求解方法。将式(11)作为算法中"演化模块"部分的迭代计算公式,式(12)作为算法中"策略改进模块"的策略改进方式。

6　合作类型最优策略对的求解算法

仿照单智能体求解 Markov 决策系统最优策略 π^* 的求解方法,本节给出合作类型联合 Markov 决策系统最优策略对 (π_0^*,π_1^*) 的求解算法。

第 0 步:给定误差参数 ε 以及平衡参数 α ;

初始化函数 V_{00}, V_{10} ;令 $V_0 = \alpha V_{00} + (1-\alpha) V_{10}$;贪心计算策略对 (π_{00},π_{10}) ,记 $V_0 = V_0^{(\pi_{00},\pi_{10})}$。

$$\pi_{01}(s,t) = \text{argmax}_a \sum_{(s',a) \in S \times A} T_0(s,t,a,s',a')[R_0(s,t,a,s',a') + \gamma \cdot V_0^{(\pi_{00},\pi_{10})}(s',t)]$$

$$\pi_{11}(s,t) = \text{argmax}_a \sum_{(t',a) \in S \times A} T_1(s,t,a,t',a')[R_1(s,t,a,t',a') + \gamma \cdot V_0^{(\pi_{00},\pi_{10})}(s,t')]$$

第 $k+1$ 步:假定已经得到 $(\pi_{0k},\pi_{1k}), V_k^{(\pi_{0k},\pi_{1k})}$ 。在此步上,做两段计算:

(1) Markov 系统演化计算。

令 $V^0 = V_k^{(\pi_{0k},\pi_{1k})}$ 作为初始 V -函数,在给定误差 ε 下,作如下迭代计算:

$$V_0^{m+1}(s,t) = \sum_{s \in S} \sum_{a \in A} T_0(s,t,\pi_0(s,t),s',a)[R_0(s,t,\pi_0(s,t),s',a) + \gamma \cdot V^m(s',t)]$$

$$V_1^{m+1}(s,t) = \sum_{t \in S} \sum_{a \in A} T_1(s,t,\pi_1(s,t),t',a)[R_1(s,t,\pi_1(s,t),t',a) + \gamma \cdot V^m(s,t')]$$

$$V^{m+1}(s,t) = \alpha V_0^{m+1}(s,t) + (1-\alpha) V_1^{m+1}(s,t)$$

当 $\| V^{m+1} - V^m \| < \varepsilon$ 时,做如下定义:

$$V_{k+1}^{(\pi_{0k},\pi_{1k})}(s,t) = V^{m+1}(s,t)((s,t) \in S \times S)$$

(2)贪心计算 $(\pi_{0,k+1},\pi_{1,k+1})$ 。

$$\pi_{0,k+1}(s,t) = \text{argmax}_a \sum_{(s',a) \in S \times A} T_0(s,t,a,s',a')[R_0(s,t,a,s',a') + \gamma \cdot V_0^{(\pi_{0k},\pi_{1k})}(s',t)]$$

$$\pi_{1,k+1}(s,t) = \text{argmax}_a \sum_{(t',a) \in S \times A} T_1(s,t,a,t',a')[R_1(s,t,a,t',a') + \gamma \cdot V_0^{(\pi_{0k},\pi_{1k})}(s,t')]$$

算法终止条件:当 $\pi_{0,k+1} = \pi_{0,k}$, $\pi_{1,k+1} = \pi_{1,k}$ 时,终止计算,同时,定义最优策略对 $\pi_0^* = \pi_0^k, \pi_1^* = \pi_1^k$。

最终, V^{m+1} 为最优 V -函数, $(\pi_{0,k+1},\pi_{1,k+1})$ 为最优策略对。

7　结束语

引入一种两个智能体联合 Markov 决策系统(CMDP),系统适用于两个智能体之间合作(或对抗)决策的演化学习系统。对于合作还是对抗,主要区别在于价值函数的定义方式与组合(或制约)方式。文

中仅考虑合作类型的 CMDP,在此类学习模型中,智能体交替执行行为,以社会价值作为求优准则,寻找最优策略 (π_0^*,π_1^*),共同完成目标任务。文中给出了寻找最优策略对的算法,其根本任务是寻找一对最优策略 (π_0^*,π_1^*),形成一个演化系统 $\mathrm{CMDP}^{(\pi_0^*,\pi_1^*)}$。下一步工作,将通过理论分析两个智能体联合 Markov 决策系统(CMDP)中价值函数的变化情况,从而寻找最优策略。文中引入的联合 Markov 决策模型适用于两个智能体之间对抗(博弈)执行,仅修改价值函数定义方式和求优过程,进一步工作也可以将该系统模型扩充到多个智能体的对抗决策系统。

参考文献:

[1] SHALEV-SHWARTZ S, BEN-DAVID S. Understanding machine learning: from theory to algorithms[M]. England: Cambridge University Press, 2014.

[2] VAN HASSELT H. Reinforcement learning in continuous state and action spaces[M]//Reinforcement learning. Massachusetts: MIT Press, 2012: 207-251.

[3] KOBER J, BAGNELL J A, PETERS J. Reinforcement learningin robotics: a survey[J]. International Journal of Robotics Research, 2013, 32(11): 1238-1274.

[4] MNIH V, KAVUKCUOGLU K, SLIVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[5] DICHEV C, DICHEVA D, AROYO L. Using topic maps for web-based education[J]. Advanced Technology for Learning, 2004, 1(1): 696-700.

[6] 周亚平,奚宏生,殷保群,等. Markov 控制过程基于性能势的平均代价最优策略[J]. 自动化学报, 2002(6): 904-910.

[7] 张继红,郭世贞,章 芸. 非时齐部分可观察 Markov 决策规划的最优策略问题[J]. 运筹学学报, 2004, 8(2): 81-87.

[8] ASMUTH J, LI L, LITTMAN M L, et al. A Bayesian sampling approach to exploration in reinforcement learning[J]. Eprint Arxiv, 2012, 58(7): 1805-1810.

[9] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: a survey[J]. Journal of Artificial Intelligence Research, 1996, 4: 237-285.

[10] PUTERMAN M L. Markov decision processes: discrete stochastic dynamic programming[M]. New Jersey: John Wiley & Sons, Inc, 1994.

[11] JOHNSON J D, LI Jinghong, CHEN Zengshi. Reinforcement learning: an introduction: R. S. Sutton, A. G. Barto[J]. Neurocomputing, 2000, 35(1-4): 205-206.

[12] PUPPALA N, SEN S, GORDIN M. Shared memory based cooperative coevolution[C]//1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). State of Alaska: IEEE, 1998: 570-574.

[13] 郭 锐,吴 敏,彭 军,等. 一种新的多智能体 Q 学习算法[J]. 自动化学报, 2007, 33(4): 367-372.

[14] LITTMAN M L. Markov games as a framework for multi-agent reinforcement learning[C]//Proceedings of the eleventh international conference on machine learning. San Francisco: Morgan Kaufmann, 1994: 157-163.

[15] LITTMAN M L. Friend-or-foe Q-learning in general-sum games[C]//Proceedings of the eleventh international conference on machine learning. San Francisco: Morgan Kaufmann, 2001: 322-328.

[16] 高 阳,周志华,何佳洲,等. 基于 Markov 对策的多 Agent 强化学习模型及算法研究[J]. 计算机研究与发展, 2000, 37(3): 257-263.

[17] JERRUM M. Mathematical foundations of the Markov chain Monte Carlo method[M]//Probabilistic methods for algorithmic discrete mathematics. Berlin: Springer, 1998: 116-165.

[18] WIERING M, VAN OTTERLO M. Reinforcement learning[M]//Adaptation, learning, and optimization. Berlin: Springer, 2012.