doi:10.11959/j.issn.1000-436x.2016117

基于可中断 Option 的在线分层强化学习方法

朱斐[1,2]，许志鹏[1]，刘全[1,2]，伏玉琛[1]，王辉[1]
（1. 苏州大学计算机科学与技术学院，江苏 苏州 215006；2. 吉林大学符号计算与知识工程教育部重点实验室，吉林 长春 130012）

摘 要：针对大数据体量大的问题，在 Macro-Q 算法的基础上提出了一种在线更新的 Macro-Q 算法(MQIU)，同时更新抽象动作的值函数和元动作的值函数，提高了数据样本的利用率。针对传统的马尔可夫过程模型和抽象动作均难于应对可变性，引入中断机制，提出了一种可中断抽象动作的 Macro-Q 无模型学习算法(IMQ)，能在动态环境下学习并改进控制策略。仿真结果验证了 MQIU 算法能加快算法收敛速度，进而能解决更大规模的问题，同时也验证了 IMQ 算法能够加快任务的求解，并保持学习性能的稳定性。
关键词：大数据；强化学习；分层强化学习；Option；在线学习
中图分类号：TP181 **文献标识码**：A

Online hierarchical reinforcement learning based on interrupting Option

ZHU Fei[1,2], XU Zhi-peng[1], LIU Quan[1,2], FU Yu-chen[1], WANG Hui[1]
(1. School of Computer Science and Technology, Soochow University, Suzhou 215006, China;
2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract: Aiming at dealing with volume of big data, an on-line updating algorithm, named by Macro-Q with in-place updating (MQIU), which was based on Macro-Q algorithm and takes advantage of in-place updating approach, was proposed. The MQIU algorithm updates both the value function of abstract action and the value function of primitive action, and hence speeds up the convergence rate. By introducing the interruption mechanism, a model-free interrupting Macro-Q Option learning algorithm(IMQ), which was based on hierarchical reinforcement learning, was also introduced to order to handle the variability which was hard to process by the conventional Markov decision process model and abstract action so that IMQ was able to learn and improve control strategies in a dynamic environment. Simulations verify the MQIU algorithm speeds up the convergence rate so that it is able to do with the larger scale of data, and the IMQ algorithm solves the task faster with a stable learning performance.
Key words: big data, reinforcement learning, hierarchical reinforcement learning, Option, online learning

1 引言

在强化学习（RL，reinforcement learning）框架中，用户给出问题的目标，agent 选择某一个动作，实现与环境的交互，获得环境给出的奖赏作为强化信号，agent 根据强化信号和环境当前状态再选择下

收稿日期：2015-04-03；**修回日期**：2016-04-12
通信作者：伏玉琛，yuchenfu@suda.edu.cn
基金项目：国家自然科学基金资助项目（No.61303108, No.61373094, No.61272005, No.61472262）；江苏省高校自然科学研究基金资助项目（No.13KJB520020）；吉林大学符号计算与知识工程教育部重点实验室基金资助项目（No.93K172014K04）；苏州市应用基础研究计划基金资助项目（No.SYG201422）；苏州大学高校省级重点实验室基金资助项目（No.KJS1524）；中国国家留学基金资助项目（No.201606920013）

Foundation Items: The National Natural Science Foundation of China (No.61303108, No.61373094, No.61272005, No.61472262), The High School Natural Foundation of Jiangsu Province (No.13KJB520020), The Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education of Jilin University(No.93K172014K04), Suzhou Industrial Application of Basic Research Program (No.SYG201422), Provincial Key Laboratory for Computer Information Processing Technology of Soochow University (No.KJS1524), The China Scholarship Council Project (No.201606920013)

一个动作。Agent的目标是在每个离散状态发现最优策略以使期望的折扣奖赏和最大[1]。作为一种具有较高通用性的机器学习框架，强化学习得到了较为广泛的研究和应用[2]。然而，由于强化学习的算法需要通过不断地与环境交互来进行学习，同时还要保存经验数据，因此当问题规模扩大时，算法的复杂度往往会以指数级上升，导致算法的性能急剧下降，所以强化学习的经典算法很难直接用于解决数据规模比较大的问题。研究人员提出了多种改进的强化学习算法来解决大规模空间的“维数灾”问题，如分层强化学习[3,4]、核方法[5]、函数逼近方法[6]等。在这些方法中，分层强化学习被用于解决一些大数据环境的任务[7]。

在分层强化学习的算法中，通过分层处理，agent关注当前局部空间的环境以及子任务目标状态的变化，策略更新的过程限定于局部空间或者高层空间上，相应地，所需解决的问题规模被限定在agent当前所处的较小规模的空间或抽象程度较高、维数较低的空间。这样不仅可以加快学习的速度，而且可以降低对环境的依赖性。在动态变化的环境中，这种特性有助于解决问题，因此显得尤为重要。时间抽象的方法是分层强化学习的一类重要方法。利用时间抽象，agent可以关注更高层策略的选择，从而降低算法的复杂度，使算法能解决一些大规模的问题。抽象动作为时间抽象提供了广泛的框架，其代表性工作是由Sutton等[8]提出的使用“宏动作”作为抽象动作的Option框架。很多方法使用子任务来表达抽象动作，子任务构成了整个任务的一部分[9]。也有很多工作寻找与子任务对应的子目标点[10~12]，以及直接从值函数中得到抽象动作[13,14]。

一般而言，大数据是指不能在可以容忍的时间内用传统信息科学的技术、软件和硬件完成感知、获取、管理、处理和服务的数据集合[15]。大数据具有体量大(volume)、多变(variability)、价值高(value)、高速(velocity)等特点。由于大数据体量大，因此很多机器学习的算法无法直接用来解决大数据问题。大数据的多变性也要求机器学习的算法在考虑数据体量的同时，考虑数据的动态变化性。在大数据问题中，当无法直接从整个问题空间上求解最优解时，如何充分利用已有抽象动作来求解是一个需要解决的重要任务。虽然，Sutton等[16]对此有过初步的研究，但是，由于其工作是基于模型已知的前提下进行规划，故而在模型未知或环境动态变化的情况下，算法性能和效果会很差，导致算法很难应用于模型无关的任务和在线学习的任务中，更无法在大数据和动态的环境中很好地学习到最优策略。本文的主要工作就是解决动态环境下如何利用时间抽象学习的问题，针对大数据体量大的特点，在Macro-Q算法的基础上提出了在线式更新的算法，加快了算法的收敛速度，提高了数据样本的利用率，同时针对大数据可变化的特点，提出了中断式动作抽象的概念，使之能很好地适应环境的变化，并在此基础上提出了一种基于中断动作抽象的无模型学习算法。

2 相关工作

2.1 强化学习

大多数的强化学习方法都是基于马尔可夫决策过程(MDP, Markov decision process)。一个MDP可以用一个5元组表示$<S,A,P,R,\gamma>$，其中，S和A分别表示有限的状态集和动作集，$P\in[0,1]$表示迁移概率，$R:S\times A\to R$表示agent得到的立即奖赏，$\gamma\in[0,1]$表示折扣因子。在每个时间步，agent观察到系统的状态$s\in S$后采取某个动作$a\in A$，然后以概率$P(s'|s,a)$迁移到下一个状态$s'\in S$，此时agent会得到一个立即奖赏$R(s,a)$。Agent的目标是通过最大化期望奖赏来找到最优策略$\pi:S\times A\to[0,1]$。

在线学习是一种在学习的过程中需要及时处理收集的数据，进行预测并更新模型的学习方式[17]。在线式强化学习通过与环境实时的交互来获取样本，然后再通过这些样本更新策略。在线强化学习能够在保证学习效果的前提下，同时给出次优的学习结果，而且在线采样比离线采样更容易。相比之下，离线的算法要求样本已知，只有在样本学完后才能应用学习好的策略。在大数据环境下，由于数据体量大，无法完全装载到内存中处理，因此，大数据环境的很多任务都采用在线学习的方式完成。

2.2 抽象动作

本文使用马尔可夫抽象动作[1,18]来描述时间抽象的动作序列。马尔可夫抽象动作和元动作同样是由agent选择的，不同的是抽象动作的执行是一个时间段，是多步完成的，而元动作则是单步完成，所以元动作被视为一种基本动作。在抽象动作执行的过程中，遵循抽象动作的内部策略π，直到满足抽象动作的终止条件。

一般的抽象动作框架由一个3元组$<I,\pi,\beta>$，

其中，$I \subseteq S$ 表示抽象动作的输入集，$\pi : S \times A \to [0,1]$ 表示策略，$\beta : S \to [0,1]$ 表示终止条件。若 $s \in I$，那么抽象动作 $< I, \pi, \beta >$ 在 s 处即为可用的。一个马尔可夫抽象动作执行过程如下：如果 agent 在状态 s_t 处选择了抽象动作 o，那么 agent 将会根据 o 的策略 π 来选择下一个动作，即 $a_t \leftarrow \pi(s_t)$。环境的状态将会迁移到 s_{t+1}，即 $(s_t, \pi) \xrightarrow{a_t} s_{t+1}$，在 s_{t+1} 处，agent 会根据终止条件 β 来判断是否终止 o 的执行，如果 $\beta(s_{t+1}) \to 0$，将会一直执行抽象动作 o 直到满足终止条件 $\beta(s_{t+k}) \to 1$。这样，对所有的元动作，都有 $\beta(s) \to 0$。当一个抽象动作终止时，agent 可以选择另外的一个抽象动作继续执行，或元动作执行。

接下来，定义在抽象动作概念下的策略。设在状态 s_t 处可用的抽象动作集合定义为 O_s，当 agent 从状态 s_t 处出发，马尔可夫策略 v 会以 $v(s_t, o)$ 的概率选择抽象动作 o，其中，$o \in O_s$。随后，将会根据 o 的策略 π 来选择动作，直到 o 在 s_{t+k} 终止。然后再根据 $v(s_{t+k})$ 选择下一个抽象动作 o'，继续执行这一过程。实际上定义在抽象动作 o 上的策略 v 决定了一个定义在动作上的策略 u，即 $u = f(v)$。可得

$$V^u(s_t) = \mathrm{E}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \mathrm{L} \mid \varepsilon(u, s_t, t)\} \quad (1)$$

其中，$\varepsilon(u, s_t, t)$ 表示策略 u 在 t 时刻从状态 s_t 处开始的这个过程。由于式(1)是建立在元动作上的，而策略 u 是由 v 决定的，所以有 $V^u(s_t) = V^{f(v)}(s_t)$。这样，类似得到

$$Q^v(s_t, o) = \mathrm{E}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \mathrm{L} \mid \varepsilon(vo, s_t, t)\} \quad (2)$$

其中，$\varepsilon(vo, s_t, t)$ 表示策略 v 首先选择了抽象动作 o，直到 o 终止，然后再根据 v 选择其他抽象动作的过程。

2.3　半马尔可夫决策过程

在强化学习中，满足马尔可夫性的强化学习任务就被称为 MDP，而一个半马尔可夫决策过程 (SMDP, semi-Markov decision process) 可以由一个 MDP 和一个抽象动作集合组成。经典的 SMDP 理论是与动作相关的，其中，相关方法可以扩展到抽象动作中来。这样，对任意的抽象动作 o，若 $\varepsilon(o, s_t, t)$ 表示 o 在 t 时刻状态 s_t 处开始的过程，那么对应奖赏的模型为

$$r_{s_t}^{o} = \mathrm{E}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \mathrm{L} + \gamma^{k-1} r_{t+k} \mid \varepsilon(o, s_t, t)\} \quad (3)$$

其中，$t+k$ 表示 o 的终止时刻。类似地，转移概率的模型为

$$p_{s_t s'}^{o} = \sum_{k=1}^{\infty} p(s', k) \gamma^k \quad (4)$$

其中，$p(s', k)$ 表示 o 在 k 个时间步后在状态 s' 终止的概率。然后根据贝尔曼等式，对任意的马尔可夫策略 u，状态值函数为

$$V^u(s_t) = \sum\nolimits_{o \in O_{s_t}} u(s_t, o)[r_{s_t}^{o} + \sum\nolimits_{s'} p_{s_t s'}^{o} V^u(s')] \quad (5)$$

对应的动作值函数为

$$Q^u(s_t, o) = r_{s_t}^{o} + \sum\nolimits_{s'} p_{s_t s'}^{o} V^u(s') \quad (6)$$

在值函数的基础上，可以得到最优值函数。在 MDP 中，选择的是最优的动作，而这里选择的是最优的抽象动作。使用 O 来定义抽象动作集合，根据贝尔曼最优等式，可以得到最优值状态函数和最优动作值函数，分别如式(7)和式(8)所示。

$$V^*(s_t) = \max\nolimits_{o \in O_{s_t}} r_{s_t}^{o} + \sum\nolimits_{s'} p_{s_t s'}^{o} V^*(s') \quad (7)$$

$$\begin{aligned} Q^*(s, o) &= r_{s_t}^{o} + \sum\nolimits_{s'} p_{s_t s'}^{o} V^*(s') \\ &= r_{s_t}^{o} + \sum\nolimits_{s'} p_{s_t s'}^{o} \max\nolimits_{o' \in O_{s'}} Q^*(s', o') \end{aligned} \quad (8)$$

根据最优值函数，得到 Q 值的更新公式为

$$\begin{aligned} &Q^u(s_t, o) \leftarrow Q^u(s_t, o) + \\ &\alpha[r_{s_t}^{o} + \gamma^k \max\nolimits_{o'} Q^u(s_{t+k}, o') - Q^u(s_t, o)] \end{aligned} \quad (9)$$

若抽象动作集合已经得到，那么就可以求出最优的状态值函数和动作值函数，最后得出最优策略。而且，标准的 SMDP 理论能够保证这样的过程能够收敛。

3　算法描述

3.1　可中断 Option

抽象动作提高了 agent 探索的效率，从而使算法收敛速度更快[2]。利用抽象动作在解决相同领域的多任务时效果很好[10]。

传统应用抽象动作的 SMDP 方法通常是把抽象动作看作一个不透明、不可分割的整体。然而，要充分地发挥抽象动作的作用，需要改变抽象动作本身的结构。这里考虑使用中断抽象动作，即抽象动作在根据它的终止条件之前，如果有需要就中断抽象动作的执行。如在房间内导航的任务中，把 agent 从房间门入口进入到房间里这个动作序列建模成一个抽象动作，当 agent 执行这个抽象动作

到刚刚准备踏入房间的那一瞬间，门突然关闭了，根据传统的 SMDP 中抽象动作的定义，此时抽象动作不应该终止，而应该继续执行，因为抽象动作的终止条件还不满足，而这就与门已经处于关闭状态形成了矛盾，导致 agent 的执行效率降低甚至失效。如果采用可中断 Option，就可以解决这一问题。

3.2 可中断 Macro-Q 算法

传统的强化算法 agent 通过与环境反复交互的方式来学习值函数和策略，但是随着问题规模的扩大，agent 就需要大量的时间和经验来与环境进行交互以获得好的策略。使用分层强化学习方法，应用抽象动作能在一定程度上减少对环境的探索，从而加快算法收敛和保证算法学习前期性能的稳定性。经典的 SMDP 方法把抽象动作看作一个不可拆分的整体，一旦抽象动作开始执行，就必须执行到抽象动作终止，不能中途结束。事实上，这种方式会面临以下 2 个主要问题：首先，在动态的环境下，往往在抽象动作还没结束时，抽象动作就执行不下去，导致算法效果很差；其次，在抽象动作执行的过程中，在某些状态选择其他的抽象动作会获得更好的性能。针对这 2 种可能出现的情况，本文提出了一种可中断 Macro-Q(IMQ, interrupting Macro-Q)算法。

假设已经得到了策略 u 的抽象动作值函数 $Q^u(s,o)$，其中，u 是全局策略，s,o 是状态-Option 对。$Q^u(s,o)$ 这个状态-Option 的值函数不仅可以评估当前采用的策略 u 好坏，而且可以评估当前每一步动作实施的好坏。假设在 t 时刻，根据策略 u 的选择，agent 当前正在跟随抽象动作 o，这时可以比较按照 o 执行得到的 $Q^u(s_t,o)$ 和中断 o 选择新的抽象动作得到的值函数为

$$V^u(s)=\sum_{o'}u(s,o')Q^u(s,o')$$

如果 $V^u(s_t)>Q^u(s_t,o)$，说明此时选择其他抽象动作得到的回报会更高，这时就中断 o，然后再根据策略 u 选择其他抽象动作是完全可行的，如算法 1 所示。

算法 1 可中断 Macro-Q 算法

输入：折扣因子 γ, 学习率 α, Option 集合 O_g

输出：Q 值函数

1) 初始化 Q 值函数和队列 QE
2) for 每个情节 do
3) 以 s_t 作为起始状态，初始化 s_t
4) repeat
5) 根据策略 u 从 O_g 中选择一个 Option $o=<I,\pi,\beta>$
6) 执行 o
7) 根据 $\pi(s_t)$ 选择动作 a
8) 观察 s', r
9) $\delta\leftarrow r+\gamma\max_{a'}Q(s',a')-Q(s,a)$
10) $Q(s,a)\leftarrow Q(s,a)+\alpha\delta$
11) 将 s_t, s', r 保存到队列 QE 中
12) if $\beta(s')$=1 or s=s' then
13) for s in QE do
14) 以批量方式更新 $Q^u(s,o)$
15) 选择一个新的 Option o'
16) end if
17) else if $V^u(s)>Q^u(s,o)$
18) for s in QE do
19) 以批量方式更新 $Q^u(s,o)$
20) 选择一个新的 Option o'
21) $s\leftarrow s'$
22) 终止执行 o
23) until s' 是终止状态
24) return Q

算法 1 是一种基于中断思想的无模型学习算法，能够很好地解决环境变化情况下，抽象动作无法整体使用的问题。

3.3 在线更新的 Macro-Q 算法

在线学习方法延伸模型的学习过程。在使用过程中，新数据的到来会引发模型的更新。而这种学习方法的一个直接负面影响是采样代价较高[19]。作为一种在线式的学习算法，经典的 Macro-Q 就需要花费完成采样。本文改进了 Macro-Q 算法，采用在线式 in-place 更新方法，在 agent 对抽象动作值更新的同时，对执行过的元动作也进行更新，如在线更新的 Macro-Q(MQIU, macro-Q with in-place updating)算法所示。Macro-Q 算法加快了 Q 值更新速率，从而加快算法的收敛速度。

算法 2 在线更新的 Macro-Q 算法

输入：折扣因子 γ, 学习率 α, Options 集合 O_g

输出：Q 值函数

1) 初始化 Q 值函数和队列 QE
2) for 每个情节 do
3) 以 s_t 作为起始状态，初始化 s_t
4) repeat
5) 根据策略 u 从 O_g 中选择一个 Option o =

<I, π, β>

6)　执行 o

7)　根据$\pi(s_t)$选择动作 a

8)　观察 s', r

9)　$\delta \leftarrow r+\gamma \max_{a'} Q(s',a')-Q(s,a)$

10)　$Q(s,a) \leftarrow Q(s,a)+\alpha\delta$

11)　将 s_t, s', r 保存到队列 QE 中

12)　if $\beta(s')$=1 then

13)　　for s in QE do

14)　　$\psi \leftarrow r_{t+1}+\gamma r_{t+2}+\mathrm{L}+\gamma^n r_{t+n}+\gamma^n \max_{o'}(s_{t+n},o')-Q(s_t,o_t)Q(s_{t,},o_t) \leftarrow Q(s_t,o_t)+\alpha\psi$

15)　　选择一个新的 Option o'

16)　end if

17)　$s \leftarrow s'$

18)　终止执行 o

19) until s'是终止状态

20) return Q

3.4　算法分析

对任意的 MDP、任意的 Option 集合 O 以及任意的马尔可夫策略 u ,定义一个新的 Option 集合 O'，这 2 个 Option 集合之间存在一一映射：对每个 $o=<I,\pi,\beta>\in O$ 定义一个相应的 $o'\in O$，其中，当 $Q^u(h,o)\geqslant V^u(s)$ 时，$\beta=\beta'$，h 表示历史，s 表示 h 的最后一个状态，选择让 o' 在状态 s 处终止：$\beta'(s)=1$。所有的以这种方式中断的历史称为中断历史。令 u' 作为在 o' 上和 u 相应的策略，则有 $u(s,o)=u'(s,o')$，那么：

1) 对所有的 $s\in S$，有 $V^{u'}(s)\geqslant V^u(s)$；

2) 如果从状态 $s\in S$ 出发，存在一个非零的概率遇到中断历史，那么有 $V^{u'}(s)>V^u(s)$；

3) 对所有的 $s\in S, o\in O$ 有 $\lim_{k\to\infty} V_k(s)=V_o^*(s)$，即算法能够最终收敛到一个不动点。

证明　对任意的状态 s，执行对终止条件改进了的策略 u'，随后再跟随策略 u，即证明下面不等式是成立的。

$$\sum_{o'} u'(s,o')[r_s^{o'}+\sum_{s'} p_{ss'}^{o'} V^u(s')] \geqslant V^u(s) \quad (10)$$

其中，$V^u(s)=\sum_o u(s,o)[r_s^o+\sum_{s'} p_{ss'}^{o'} V^u(s')]$。

如果不等式(10)成立，扩展左式，重复使用 $\sum_{o'} u'(x,o')[r_x^{o'}+\sum_{x'} p_{xx'}^{o'} V^u(x')]$ 替换左式 $V^u(x)$。在极限的情况下，左式变成 $V^{u'}$，即可证明 $V^{u'}\geqslant V^u$。

因为 $u'(s,o')=u(s,o), \forall s\in S$，需要证明

$$r_s^{o'}+\sum_{s'} p_{ss'}^{o'} V^u(s') \geqslant r_s^o+\sum_{s'} p_{ss'}^o V^u(s') \quad (11)$$

令 Γ 表示所有的中断历史 $\Gamma=\{h\in\Omega: \beta(h)\neq\beta'(h)\}$。那么式(11)左边可以写为

$$\mathrm{E}\{r+\gamma^k V^u(s') \mid \varepsilon(o',s), h_{ss'}\notin\Gamma\}+$$
$$\mathrm{E}\{r+\gamma^k V^u(s') \mid \varepsilon(o',s), h_{ss'}\in\Gamma\}$$

其中，s',r,k 分别表示下一个状态，立即奖赏及从状态 s 处跟随 Option o 执行的步数，$h_{ss'}$ 表示从状态 s 到状态 s' 的历史。由于轨迹中碰到了 $h_{ss'}\notin\Gamma$ 而从没有碰到 $h_{ss'}\in\Gamma$，所以轨迹会终止，而且在状态 s 处执行 o 之后会以同样的概率和相同的期望出现。所以，不等式(11)的右边可以改写为

$$\mathrm{E}\{r+\gamma^k V^u(s') \mid \varepsilon(o',s), h_{ss'}\notin\Gamma\}+$$
$$\mathrm{E}\{\beta(s')[r+\gamma^k V^u(s')]+(1-\beta(s'))\cdot$$
$$[r+\gamma^k Q^u(h_{ss'},o)] \mid \varepsilon(o',s), h_{ss'}\in\Gamma\}$$

因为对所有的 $h_{ss'}\in\Gamma$，$Q_o^u(h_{ss'},o)\leqslant V^u(s')$，所以式(10)得证，从而有 $V^{u'}\geqslant V^u$。如果至少存在一条以非零的概率终止于由 o' 产生轨迹的历史有 $Q_o^u(h_{ss'},o)<V^u(s')$，那么不等式(11)就成立，即 $V^{u'}>V^u$。

4　仿真实验

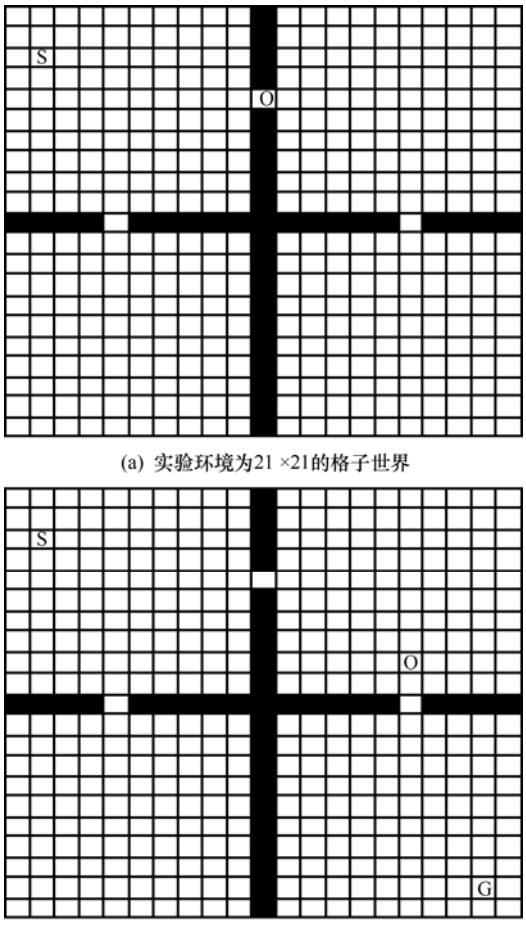
本文在格子世界实验的基础上，模拟动态和静态的环境进行仿真实验。通过与 Q-learning 做实验对比并给出实验结果来仿真验证 IMQ 的可行性和有效性。在仿真实验中，agent 使用 ε-greedy 进行探索，初始探索概率 $\varepsilon=0.1$，学习率 $\alpha=0.1$，Q 值都初始化为 0，也可以被随机初始化。根据问题规模的不同，将提供不同的抽象动作集合。

4.1　动态环境的描述

到目前为止，强化学习大多数的研究都用于解决一些简单的学习任务，如房间导航问题、平衡杆问题、直流电机问题、过山车问题等。但是这些问题大多都是设定为静态环境的。如房间导航问题中，只有固定的墙壁或障碍物。然而，在实际的应用环境往往是未知的或者会发生变化。相应地，房间导航问题的设定中，障碍物应该是随机出现的，而且出现的位置也应该是随机的。本文的一个目标就是在动态的、不断变化的环境中找到最优策略。

在图 1(a)所示的动态格子世界的仿真实验中，

共有 21×21 个网格，标记为“S”的格子表示 agent 的出发点，标记为“G”的格子表示 agent 的目标终点，标记为“O”的格子表示障碍物。动态格子世界环境是会动态变化的，包括 2 种变化的对象：agent 和障碍物的位置。对比图 1(a)和图 1(b)可以发现，在不同的时间，障碍物的位置是不一样的。



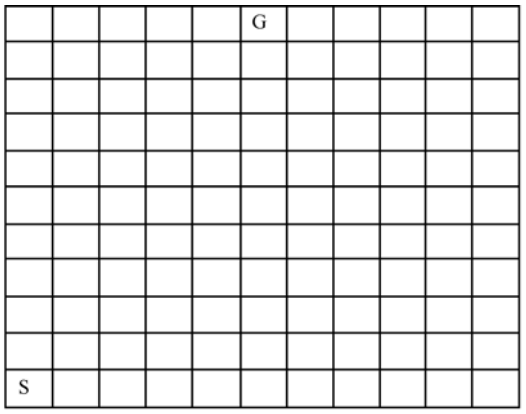
(a) 实验环境为21 ×21的格子世界

(b) 障碍物的位置发生了变化

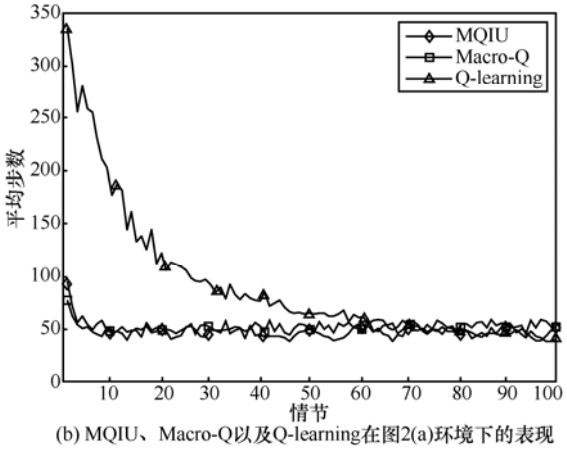
图 1　一个 21×21 的网格的动态环境示意

4.2　MQIU 在格子世界中的性能

为了衡量 MQIU 的性能，本文在仿真实验环境下同时实现了 Macro-Q、Q-learning 和 MQIU。实验环境为一个 11×11 的格子世界，如图 2(a)所示，agent 的出发点设在左下方，用“S”表示，目标点设在格子顶部的中间，用“G”表示。Agent 的任务是从“S”出发，以最快的方式到达目标点“G”，agent 所能采取的元动作为上、下、左和右。在算法 Macro-Q 和 MQUI 中，agent 所能采取的动作除了上、下、左、右这 4 个元动作外，对每个状态还有 4 个可选的抽象动作，分别沿 4 个方向移动，直到碰到墙为止。



(a)实验环境为11×11的格子世界



(b) MQIU、Macro-Q以及Q-learning在图2(a)环境下的表现

图 2　11×11 的格子世界中 MQIU、Macro-Q 和 Q-learning 的算法性能比较

从图 2(b)可以看出，MQIU 和 Macro-Q 比 Q-learning 收敛更快，而且在整个学习过程中 MQIU 和 Macro-Q 都保持了很好的性能，平均每个情节步数维持在 50 步内。对比 Macro-Q 可以看出，MQIU 在前 15 个情节稍差，但是在第 15 个情节之后，MQIU 算法的性能就好于 Macro-Q。产生这种现象的原因是 MQIU 在对抽象动作更新的同时更新了元动作的 Q 值，从而会加快值的收敛速度。

4.3　IMQ 在 4 房间静态格子世界中的性能

本文首先对 IMQ 在静态环境下的表现做了深入的说明，如图 3 所示。4 个房间静态格子世界实验如图 3(a)所示，其中，“S”代表出发点，“G”代表目标点。Agent 从“S”出发，经过房间之间的通道到达“G”，则一个情节结束。为了更好地说明算法的性能，IMQ 和 Macro-Q 所使用的抽象动作是完全一样的。实验中的抽象动作设为每个房间内 2 个，一共 8 个，每个抽象动作能够把 agent 从房内任意一点带到房间的出口处。

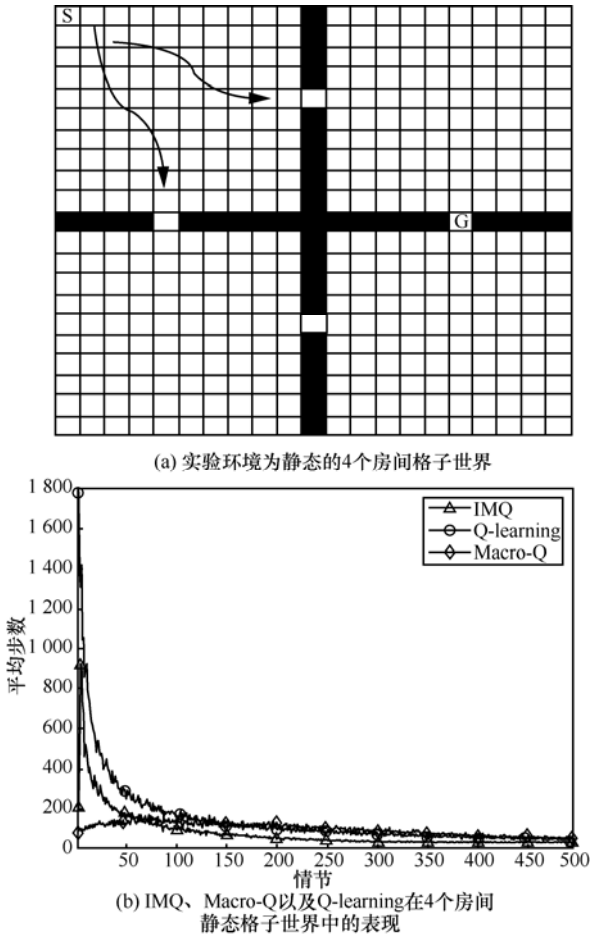


图 3　Macao-Q 和 Q-learning 在 4 个房间格子世界中学习性能曲线对比

从图 3(b)中可以看出，在 4 个房间格子世界中，IMQ 和 Macro-Q 的算法性能比 Q-learning 好很多。Macro-Q 性能较为稳定，在整个学习的过程中一直保持很低的学习步数，然而其收敛速度和 Q-learning 一样，在 500 个情节后收敛。IMQ 注重探索，在前 50 个情节性能比 Q-learning 好，略差于 Macro-Q，但是 IMQ 收敛效果很好，在 200 个情节的时候就达到了收敛，并且一直保持很稳定。

4.4　IMQ 在 4 个房间动态格子世界中的性能

4 个房间动态格子世界实验如图 4(a)所示。由于在这个实验中，环境被设置为动态变化的，因此更能检验算法的性能。目标状态“G”被放置在右下角的房间，起始状态“S”被放置在左上角房间的角落里。每个情节会随机初始化 25 个障碍物“O”，用来表示随机的环境。元动作是 4 个方向的动作：上、下、左和右。Agent 在贪心动作（元动作或者抽象动作）的选择概率为$1-\varepsilon+\frac{\varepsilon}{|A|_s+|O|_s}$，其他方向上，元动作或者抽象动作的选择概率为$\frac{\varepsilon}{|A|_s+|O|_s}$。Agent 每走一步的奖赏都是−1，到达目标点的奖赏是 0。由于本文关注的重点是抽象动作在动态环境中的应用，因此这里的抽象动作是预先定义好的。实验中对比了 IMQ 和 Q-learning，没有对比 Macro-Q 以及基于规划的中断方法，是因为在动态环境下，这 2 种算法性能都很差。Macro-Q 没有引

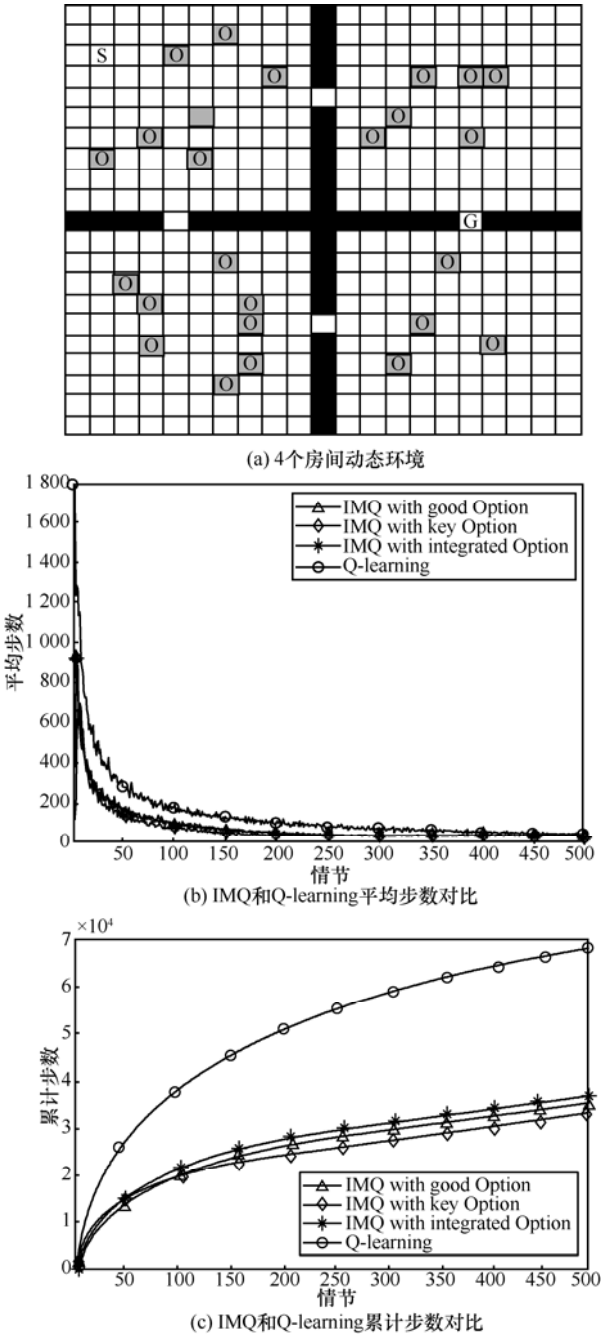


图 4　在 4 个房间动态环境下带有不同抽象动作集合的 IMQ 算法与 Q-learning 学习性能比较

入中断机制，导致如果抽象动作的执行过程被破坏，那么将无法继续按照抽象动作的内部策略继续执行。而基于规划的中断方法用于在线的算法中并不是很合理，而且需要模型，因此这里没有对比这 2 种算法。

图 4(b)显示了在 100 次重复实验的基础上，agent 从起始状态到达目标状态的平均步数，对比了 IMQ 和 Q-learning 在动态的格子世界中的性能。从图中可以看出带有不同抽象动作集合的 3 种 IMQ 算法无论是在收敛速度还是在学习时的表现上均好于 Q-learning。其中，IMQ with integrated Option 在性能上略差于另外 2 个 IMQ 算法，IMQ with good Option 的性能总体上和 IMQ with key Option 相当；但是从图 4(c)可以看出，IMQ with key Option 仅在前 50 个情节略差于 IMQ with good Option，从长期学习来看，IMQ with key Option 学习效率更高，收敛更快。仿真实验证明了算法在动态环境下的有效性。为了更精确地说明几种算法的性能对比，在表 1 中给出了 4 个房间动态格子世界中各算法性能的对比数据。

表 1　4 个房间实验中，不同抽象动作集 IMQ 和 Q-learning 的对比实验结果数据

算法	收敛情节	收敛累计步数
IMQ with good Option	250	25 000
IMQ with key Option	200	23 000
IMQ with integrated Option	250	27 000
Q-learning	500	70 000

4.5　IMQ 在 6 个房间动态格子世界中的性能

作为 IMQ 的第 3 个实验，在更大规模的环境下进行实验验证。本文使用 6 个房间的动态格子世界来进行仿真实验。Agent 的任务和前面描述的基本一样，从起始点状态“S”走到目标状态“G”。实验的环境如图 5(a)所示，其中，起始状态“S”靠近左上角，目标状态靠近右边。随机环境以及元动作的设定和前面介绍的一样，随机生成 25 个障碍物，用“O”表示。提供的抽象动作和前一节介绍的一样，但是由于房间的增多，这里提供的抽象动作的数量也会相应的变化。

图 5(b)显示了在 100 次重复实验的基础上，agent 从起始状态到达目标状态的平均步数，这个图与 4 个房间实验中的图相比，区别在于状态的增多、环境的复杂度更高，导致 agent 在学习的前期到达目标点所需的步数的增加，同时收敛速度也有所减缓。从图 5 可以看出，随着环境规模的增大，各算法间的区别更加明显。实验图 5(b)表明，3 种 IMQ 算法表现均优于 Q-learning，其中，IMQ with key Option 达到收敛所需的总步数最少，情节数也最少，这说明，关键的抽象动作能够更有效地加快 agent 的学习效率。

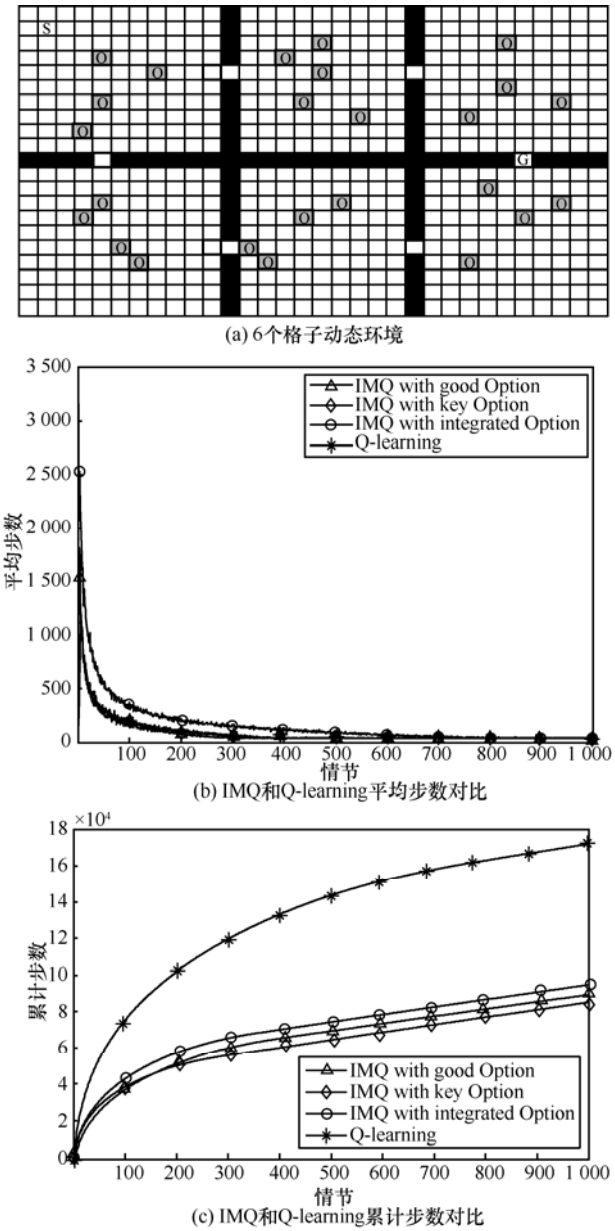


图 5　在 6 个房间动态环境下带有不同抽象动作集合的 IMQ 算法与 Q-learning 学习性能比较

5　结束语

本文的工作主要包括以下几个方面。首先，针对传统 SMDP 方法不能解决动态环境下的学习

和控制问题，本文提出一种在线学习的使用可中断动作抽象的算法—— IMQ。借助于分层强化学习的方法，IMQ 算法能够有效解决大数据环境下一般强化学习算法由于时间复杂度过高而不能解决的问题。相比于离线算法，IMQ 算法能够在线地进行学习和采样，从而在加快学习效率的同时又保证了算法的性能。实验结果表明，IMQ 算法比 Q-learning 算法和 Macro-Q 算法具有更快的收敛速度。

其次，针对 Macro-Q 算法样本利用率不高的问题，本文提出了一种基于同步替代更新的算法——MQIU 算法。在算法中，对抽象动作的值函数进行更新的同时，也更新元动作的值函数。实验结果表明，MQIU 算法较 Macro-Q 效果略好，收敛速度上略快。

第三，针对传统的抽象动作不能很好地解决动态环境的问题，本文将中断的方式引入抽象动作的概念中，提出了中断式动作抽象的概念，使之能很好地适应环境的变化，并在此基础上提出了一种基于中断动作抽象的无模型学习算法。实验结果表明，在动态的环境下，适当地利用抽象动作能够加快任务的求解，并且有助于 agent 在学习的过程中保持性能的稳定。

然而，在本文中的抽象动作是预先定义好的，如何快速有效地自动发现合适的抽象动作来加快长期学习 agent 的学习效率，是将要研究的一个重要内容。另外，在动态的环境下，如何充分利用样本的模型学习以及如何将抽象动作用于多任务、多 agent 协作也是主要的一项工作。

参考文献：

[1] OTTERLO M V, WIERING M. Reinforcement learning and Markov decision processes[J]. Adaptation Learning & Optimization, 2012, 206(4):3-42.

[2] VAN H H. Reinforcement learning: state of the art[M]. Berlin: Springer, 2007.

[3] 沈晶, 顾国昌, 刘海波. 未知动态环境中基于分层强化学习的移动机器人路径规划[J]. 机器人, 2006, 28(5):544-547.
SHEN J, GU G C, LIU H B. Mobile robot path planning based on hierarchical reinforcement learning in unknown dynamic environment[J]. ROBOT, 2006, 28(5): 544-547.

[4] 刘全, 闫其粹, 伏玉琛, 等. 一种基于启发式奖赏函数的分层强化学习方法[J]. 计算机研究与发展, 2011, 48(12): 2352-2358.
LIU Q, YAN Q C, FU Y C, et al. A hierarchical reinforcement learning method based on heuristic reward function[J]. Journal of Computer Research and Development, 2011, 48(12): 2352-2358.

[5] 陈兴国, 高阳, 范顺国, 等. 基于核方法的连续动作 Actor-Critic 学习[J]. 模式识别与人工智能, 2014(2): 103-110.
CHEN X G, GAO Y, FAN S G, et al. Kernel-based continuous-action actor-critic learning[J]. Pattern Recognition and Artificial Intelligence, 2014(2):103-110.

[6] 朱斐, 刘全, 傅启明, 等. 一种用于连续动作空间的最小二乘行动者-评论家方法[J]. 计算机研究与发展, 2014, 51(3): 548-558
ZHU F, LIU Q, FU Q M, et al. A least square actor-critic approach for continuous action space[J]. Journal of Computer Research and Development, 2014, 51(3): 548-558.

[7] 唐昊, 张晓艳, 韩江洪, 等. 基于连续时间半马尔可夫决策过程的 Option 算法[J]. 计算机学报, 2014(9): 2027-2037.
TANG H, ZHANG X Y, HAN J H, et al. Option algorithm based on continuous-time semi-Markov decision process[J]. Chinese Journal of Computers, 2014(9): 2027-2037.

[8] SUTTON R S, PRECUP D, SINGH S. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1): 181-211.

[9] MCGOVERN A, BARTO A G. Automatic discovery of subgoals in reinforcement learning using diverse density[J]. Computer Science Department Faculty Publication Series, 2001(8):361-368.

[10] ŞIMŞEK Ö, WOLFE A P, BARTO A G. Identifying useful subgoals in reinforcement learning by local graph partitioning[C]//The 22nd International Conference on Machine Learning. ACM, c2005: 816-823.

[11] ŞIMŞEK Ö, BARTO A G. Using relative novelty to identify useful temporal abstractions in reinforcement learning[C]//The Twenty-first International Conference on Machine Learning. ACM, c2004: 751-758.

[12] CHAGANTY A T, GAUR P, RAVINDRAN B. Learning in a small world[C]//The 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems. c2012: 391-397.

[13] SUTTON R S, SINGH S, PRECUP D, et al. Improved switching among temporally abstract actions[J]. Advances in Neural Information Processing Systems, 1999: 1066-1072.

[14] CASTRO P S, PRECUP D. Automatic construction of temporally extended actions for mdps using bisimulation metrics[C]//European Conference on Recent Advances in Reinforcement Learning. Springer-Verlag, c2011: 140-152.

[15] 何清, 李宁, 罗文娟, 等. 大数据下的机器学习算法综述[J]. 模式识别与人工智能, 2014, 27(4): 327-336.
HE Q, LI N, LUO W J, et al. A survey of machine learning algorithms for big data[J]. Pattern Recognition and Artificial Intelligence, 2014,27(4): 327-336.

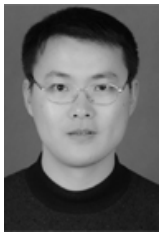
[16] SUTTON R S, PRECUP D, SINGH S P. Intra-option learning about temporally abstract actions[C]//ICML. c1998, 98: 556-564.

[17] 石川, 史忠植, 王茂光. 基于路径匹配的在线分层强化学习方法[J]. 计算机研究与发展, 2008, 45(9): 1470-1476
SHI C, SHI Z Z, WANG M G. Online hierarchical reinforcement learning based on path-matching[J]. Journal of Computer Research and Development, 2008, 45(9): 1470-1476.

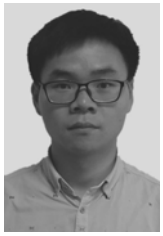
[18] BOTVINICK M M. Hierarchical reinforcement learning and decision making [J]. Current Opinion in Neurobiology, 2012, 22(6): 956-962.

[19] 王爱平, 万国伟, 程志全, 等. 支持在线学习的增量式极端随机森林分类器[J]. 软件学报, 2011, 22(9):2059-2074.
WANG A P, WAN G W, CHENG Z Q, et al. Incremental learning extremely random forest classifier for online learning[J], Journal of Software, 2011, 22(9):2059-2074.

作者简介：



朱斐（1978-），男，江苏苏州人，博士，苏州大学副教授，主要研究方向为机器学习、人工智能、生物信息学等。



许志鹏（1991-），男，湖北荆州人，苏州大学硕士生，主要研究方向为强化学习、人工智能等。

刘全（1969-），男，内蒙古牙克石人，博士后，苏州大学教授、博士生导师，主要研究方向为多强化学习、人工智能、自动推理等。

伏玉琛（1968-），男，江苏徐州人，博士，苏州大学教授、硕士生导师，主要研究方向为强化学习、人工智能等。

王辉（1968-），男，陕西西安人，苏州大学讲师，主要研究方向为强化学习、人工智能等。