doi:10.11959/j.issn.1000-436x.2015047

支持合并的自适应 tile coding 算法

施梦宇[1]，刘全[1,2]，傅启明[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：针对自适应 tile coding 算法会产生多余划分的问题，提出一种支持合并的自适应 tile coding 算法——MATC。该算法能够消除传统自适应 tile coding 算法中产生的多余划分，进一步解决连续状态空间离散化的问题。将 MATC 算法应用于离散动作连续状态的 Mountain Car 问题上，实验结果表明，该算法在学习过程中能消除传统 tile coding 算法的误划分所产生的不良影响，更准确地自动调整划分的精度，并更快地收敛到最佳策略。
关键词：连续空间；离散化；强化学习；自适应；tile coding
中图分类号：TP181 **文献标识码**：A

Mergeable adaptive tile coding method

SHI Meng-yu[1], LIU Quan[1, 2], FU Qi-ming[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: In order to solve many unnecessary division, mergence supported adaptive tile coding algorithm was presented which would eliminate the unnecessary division. Simulation is conducted on mountain car problem with discrete actions and continuous state space. Results show that the proposed method can eliminate the influence of false division in the traditional tile coding method and achieve a more accurate adaptive partition of continuous state space. A higher convergence rate is achieved at the same time.
Key words: continuous space; discretization; reinforcement learning; adaptive; tile coding

1 引言

强化学习又称为增强学习、再励学习或激励学习，是一种从环境状态到动作映射的学习方法。在与环境的交互学习中，agent 选择动作，环境对这些动作做出响应，并产生新的场景给 agent，同时环境也将给出评价性回馈信号。来自环境的评价性反馈信号通常称为奖赏值或强化信号，随着时间的推进，agent 的目标就是极大化(或极小化)期望奖赏值。

许多领域的实际问题都可以描述为强化学习问题，因而强化学习具有广阔的应用前景。但是，由于实际系统的空间往往是大规模或连续的，使强化学习不可避免地存在状态变量的空间复杂度过大的问题，即“维数灾”问题。因此，与强化学习的理论模型相比，实际的应用问题要复杂得多，这导致强化学习理论在实际应用中存在非常多的困难。

目前，解决强化学习中大规模连续状态的表示问题主要有 2 类方法。参数化表示的函数逼近方法和离散化方法。相比于函数逼近方法，离散化方法更加简单、易于实施。对于离散化方法，如果简单地采用对连续空间进行等精度划分的方法，若划分过于粗略，则不能很好地表达状态空间；若划分过于细致，则状态数量随着划分精度呈指数型增长，面临“维数灾”问题。Tile coding 算法[1]是解决这一问题的一种基本算法。实际上，状态空间中与学习目标相关的状态只占整个离散状态空间的极少部分[2]，且这些相关状态也并不是均匀分布在整个状态空间中。因此，对于整个状态空间进行等精度

收稿日期：2013-11-15；**修回日期**：2014-03-10
基金项目：国家自然科学基金资助项目（61272005, 61472262）；江苏省自然科学基金资助项目（BK2012616）
Foundation Items: The National Natural Science Foundation of China (61272005, 61472262); The National Science Foundation of Jiangsu Province (BK2012616)

离散化并不能取得很好的效果。由 *K*-means 衍生出的一系列算法[3~6]虽然能够解决这一问题，但算法的效果过于依赖初始值的选取且计算量太大。此外，径向基生长算法[7]、可变分辨率算法[8]、基于树表示的算法[9]虽然能节约更多的存储空间，但是并未能取得更好的离散效果。Alexander 等[10]的研究成果表明，当泛化性随着学习过程的进行而不断地减小情况下，可以取得更好的学习效果。针对这一思想，为了获得更好的离散效果，Shimon 等[11,12]结合自适应算法提出了一种自适应的 tile coding(ATC, adapting tile coding)算法。Nokhbeh-Zaeem 等[13]把 ATC 算法与神经网相结合，使得离散状态空间能更精确地逼近实际状态空间，从而获得更优的学习效果。Stephen 等[14] 提出一种改进后的自适应规则也能大幅提高离散空间的表征能力。

本文提出一种支持合并的自适应 tile coding (MATC, merging to adapt tile coding)算法，该算法在传统的自适应 tile coding 算法中融入状态合并的思想，更进一步地解决连续状态空间离散化的问题，并对 MATC 算法给出了收敛性证明。将 MATC 算法用于有限动作连续状态的 Mountain Car 实验，结果表明，MATC 算法具有更快的收敛速度以及在同等精度下更强的表征能力。

2　背景知识

2.1　马尔科夫决策过程

马尔科夫决策过程由四元组 $<X,U,\rho,f>$ 定义。X：环境状态集，U：策略集合，ρ：奖惩函数，f：状态转移函数。本质是：当前状态向下一状态转移的概率和奖赏只取决于当前状态和选择的动作，与历史状态、动作无关。在每一个时间步 k，环境处于状态集合 X 中的某一状态 x_k，agent 选择动作集合 U 中的一个动作 u_k，收到立即奖赏 r_k，并转移至下一状态 x'_k。状态转移函数 $f(x_k,u_k,x'_k)$ 表示在状态 x_k 执行动作 u_k 转移到状态 x'_k 的概率。状态转移和奖赏函数都是随机的。

agent 目标就是寻求一个最优控制策略 $\pi^*:x\to u$，使得长期回报值最大化。在给定模型的强化学习问题中，agent 只需要找到最优的值函数 $V^*:X\to\rho$，$V^*(x)$ 是以 x 为起始状态 agent 所能获得的折扣长期回报。当获得 V^* 后，π^* 就很容易确定

$$\pi^*(u)=\arg\max_u[\rho(x,u)+\gamma V^*(f(x,u))]$$

其中，$\gamma\in[0,1]$ 是折扣因子。

2.2　自适应 tile coding

Tile coding 是一种连续空间离散化算法，这种算法的离散状态数目是人为设定的。Tile coding 将状态空间划分成若干个 tiling。通常来说，这些 tilings 只是彼此之间存在一定偏移量完全相同的划分，而 tiling 中的每个分量被称作一个 tile。只有当所给状态落在相应 tile 的区域时，这些二值特征才会被激活。图 1 给出了一个由 2 个二维 tiling 组合而成的 tile coding。

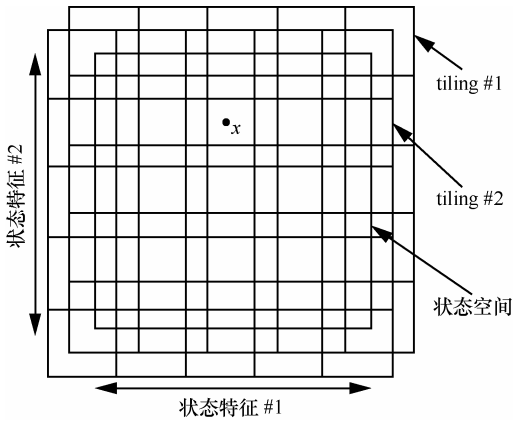

图 1　多个重叠的网格 tiling

Tile coding 所表示的值函数是由一系列的权值确定的，每一个权值代表一个 tile，例如

$$V(x)=\sum_{i=1}^{n}b_i(x)w_i$$

其中，n 为 tile 的总数，$b_i(x)$ 为状态 x 下第 i 个 tile 的值(0 或 1)，w_i 为 tile 的权值。事实上，并不需要计算所有 n 个 tile 的总和，因为在每一个 tiling 中只有一个 tile 是被状态 x 激活的。给定 m 个 tiling，可以很方便地计算出被激活的 m 个 tile 的索引以及它们的权值。

给定一个马尔科夫模型，可以利用动态规划计算 $\Delta V(x)$ 来更新状态 x 的估计值

$$\Delta V(x)=\max_u[\rho(x,u)+\gamma V(f(x,u))]-V(x)$$

此外，还可以调整每一个权值来减少 $\Delta V(x)$

$$w_i\leftarrow w_i+\frac{\alpha}{m}b_i(x)\Delta V(x)$$

其中，α 为学习速率参数。

ATC 算法是一种开始只确定一个简单划分的

tiling，然后再逐渐细分现有 tiling 的离散化算法。这使算法的精度随着划分数目的逐渐增加而不断提高。算法开始只具有少数的表征范围较大的 tile，在学习过程中通过对现有 tile 的划分逐渐地增加 tile 的数目。目前，ATC 算法多采取对已有 tile 进行均分的分割算法。图 2 描述了一个二维空间的分割过程。

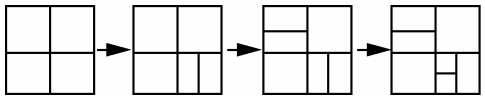

图 2 自适应 tile coding 分割过程

agent 通过分析当前的值函数和策略能决定何时对哪些 tile 进行分割。通过这种算法可以自动地获得一个有效的分割。

3 支持合并的自适应 tile coding

3.1 算法说明

ATC 算法相比于传统的 tile coding 算法具有更好的学习效率而且能更好地表达一个连续的空间，即能用更小的存储空间精确地表达整个连续空间。但是，这种逐渐分割的自适应算法也在很大程度上浪费许多不必要的存储空间。

一方面，ATC 算法是在学习过程中根据当前的值函数及策略对已有 tile 进行逐步划分。但是，在学习过程中的值函数还没有完全地收敛，所以并不能很好地表达当前状态的好坏程度。因此，基于当前值函数及策略的划分就很有可能造成很多的误划分，即会进行许多没有必要的划分。

另一方面，开始时只对状态空间进行了粗略的分割，每个离散区域的范围比较大，而学习过程中所得到的值函数则可以看作是区域内所有状态值函数的平均值，并且在进行分割时采取的是均分策略，所以可能导致值函数相似的局部地区也被划分(如图 3 所示)。

图 3 中左图显示了一个 tile 中各个不同区域的权值，而 tile 的左右两半的权值差达到了分割的阈值，则对 tile 进行分割得到图 3 中右侧的 2 个相邻的 tile。从图中可以看到，虽然对于 tile 的左侧区域和右侧区域来说，它们满足分割的要求，但是这种对整个 tile 进行均分的分割也不一定是必要的，例如 tile 中上半部分的 2 个相邻区域具有很强的相似性却被分在 2 个部分。这对于图 3 中 tile 的上半部分的 2 个相邻区域就造成一次误划分。

在精度足够高的情况下，这种误划分不仅会对当次的分割有效性造成影响，也会对之后的分割产生较大的影响，需要更多的划分来弥补(如图 4 所示)。为使它们都能够收敛，就需要更多的情节。而在精度确定的情况下，可能会出现没有足够多的划分来弥补这种误分割，这就使得最终的划分并不能达到理想的情况。所以，这种不合理的分割不仅会导致存储空间的浪费，而且对学习的收敛效果也有一定的影响。

图 4 经过多次分割后的相邻区域

图 4 是对图 3 中上半部分被分割开的左右 2 个区域又进行多次分割后的结果。从图中可以看出，在最差的情况下，3 对横向相邻的区域权值都非常相似，但却因为之前的一次误划分而都被分割成 2 部分。这与最好的划分情况相比，就要多进行一倍以上的分割，也多出了一倍的权值需要学习收敛，从而不仅大大地浪费存储空间，也减缓了学习的速率。针对这一问题，本文提出一种 MATC 算法。MATC 算法虽然不能阻止误划分的产生，但却可以在误划分产生后对其进行一定的消除，减弱误划分产生的影响。

剩下所要确定的问题就是：何时进行合并以及对哪些 tile 进行合并。合并的策略可以有多种，也存在不同的决定合并的因素。本文采用一种简单的整边合并算法，只有当 2 个相邻的 tile 在低一维空间中具有完全相同的表征范围时才可能发生合并，

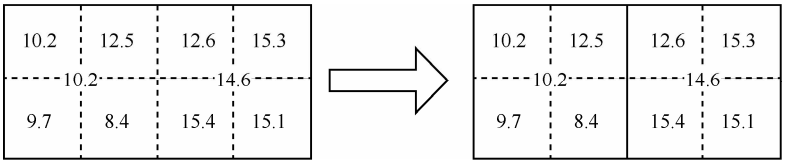

图 3 自适应 tile coding 的一次分割

即在二维空间中 2 个 tile 具有一条相同边而在三维空间中 2 个 tile 具有一个相同面才可能发生合并。这能确保合并后的 tile 仍是规则的，为算法的实现提供了很大的方便。而对于那些与当前更新的 tile 可能发生合并的 tile，计算它们与该 tile 的权值差 $\Delta w = |w(x') - w(x)|$。当 Δw 小于当前 Bellman 误差时，则进行相应 tile 的合并。具体算法如下。

算法 1　MATC 算法

1) 初始化一个具有 n 个 tile 的 tiling

step1　重复 n 次（i 为次数）

赋第 i 个 tile 和 $2k$ 个 sub-tile 的权值为 0;

step2　$c \leftarrow 0$ （当前最小 Bellman 误差维持的更新次数）;

2) Repeat

step1　$x \leftarrow$ 从 x 中随机选取一个状态

step2　$\Delta V(x) \leftarrow \max_u[\rho(x,u) + \gamma V(f(x,u))] - V(x)$

step3　$w \leftarrow$ 被 x 激活的 tile 权值

step4　$w \leftarrow w + \alpha\Delta V(s)$

step5　重复 $2k$ 次（d 为次数）

① $w_d \leftarrow$ 被激活的 sub-tile 权值

② $\Delta w_d \leftarrow \max_u[\rho(x,u) + \gamma V(f(x,u))] - w_d$

$w_d \leftarrow w_d + \alpha\Delta w_d$

③ $\Delta w \leftarrow |w(x') - w(x)|$

④ if $\Delta w <$ Bellman 误差合并两相邻 tile

step1　if $|\Delta V| <$ 所有被激活状态的 Bellman 误差

$c \leftarrow 0$

step2　else

$c \leftarrow c + 1$

step3　if $c > p$

对该 tile 进行分割

$c \leftarrow 0$

3) until 到达目标或训练结束。

MATC 算法在原有的 ATC 算法中加入相邻 tile 合并的部分，根据一定的阈值条件决定是否要对相应的 tile 进行合并。

3.2　收敛性分析

定理 1　在一个有有界回报 $(\forall x,u)|\rho(x,u) \leqslant c$ 的确定性 MDP 中，agent 用 $\hat{V}(x) \leftarrow \max_u(\rho(x,u) + \gamma\hat{V}(f(x,u)))$ 规则训练，将表 $\hat{V}(x)$ 初始化为任意有限值，并且使用折扣因子 γ，$0 \leqslant \gamma \leqslant 1$。令 $\hat{V}_n(x)$ 代表在第 n 次更新后 agent 的假设 $\hat{V}(x)$。如果每个状态都被无限频繁地访问，那么对所有 x，当 $n \to \infty$ 时，$\hat{V}_n(x)$ 收敛到 $V(x)$。

证明　因为每个状态无限频率发生，考虑连续的区间，其中每个状态至少发生一次。所需要证明的是，在 $\hat{V}$ 表中所有表项上的最大误差在区间内至少按因子 γ 减少。$\hat{V}_n$ 为 n 次更新后 agent 估计的 V 值表。令 Δ_n 为 $\hat{V}_n$ 中最大误差，即

$$\Delta_n \equiv \max_u |\hat{V}_n(x) - V(x)|$$

使用 x' 代表 $f(x,u)$，r 代表 $\rho(x,u)$，对在第 n+1 次迭代中更新的任意表项 $\hat{V}_n(x)$，修正后的估计 $\hat{V}_{n+1}(x)$ 的误差量为

$$\begin{aligned}
|\hat{V}_{n+1}(x) - V(x)| &= |\max_u(r + \gamma\hat{V}_n(x')) - \max_u(r + \gamma V(x'))| \\
&= \gamma|\max_u \hat{V}_n(x') - \max_u V(x')| \\
&\leqslant \gamma\max_u |\hat{V}_n(x') - V(x')| \\
&\leqslant \gamma\max_{x',u} |\hat{V}_n(x') - V(x')| \\
|\hat{V}_{n+1}(x) - V(x)| &\leqslant \gamma\Delta_n
\end{aligned}$$

因此，对任意 x，更新后的 $\hat{V}_{n+1}(x)$ 的误差最多为 $\hat{V}_n$ 表中最大误差 Δ_n 的 γ 倍。在初始表中的最大误差 Δ_0 是有界的，因为 $\hat{V}_0(x)$ 和 $V(x)$ 的值对所有 x 都有界。在每个 x 都被访问过的第一个区间内，此表中最大的误差最多为 $\gamma\Delta_0$。在 k 个区间后，误差最多为 $\gamma^k\Delta_0$。因为每个状态都被无限频率地访问，这样区间的数目是无限的，因此当 $n \to \infty$ 时，$\Delta_n \to 0$。　□

定理 2　MATC 算法的收敛性。在一个连续状态空间的确定性 MDP 中，agent 用 MATC 算法对状态空间进行逐步离散化的学习。将表 $\hat{V}(x)$ 初始化为任意有限值，并且使用折扣因子 γ $(0 \leqslant \gamma \leqslant 1)$，$x \in X$ (X 为离散化后的状态空间)。令 $\hat{V}_n(x)$ 代表在第 n 次更新后 Agent 的假设 $\hat{V}(x)$，X_n 代表第 n 次更新后的离散化状态空间。如果每个状态—动作对都被无限频繁的访问，那么对所有 $x \in X_n$，当 $n \to \infty$ 时，$\hat{V}_n(x)$ 收敛到 $V(x)$。

证明　在学习初期，离散化的状态空间还尚未达到稳定状态，即 X_n 尚未稳定或存在 x（$x \in X_n$）的离散分布尚未达到稳定状态。考虑离散化状态空间稳定后的情形，由于不再对离散化状态空间进行重划分，所以可以转化为证明在给定表 $\hat{V}(x)$ 初始化

值的情况下，对所有 x，当 $n\to\infty$ 时，$\hat{V}_n(x)$ 收敛到 $V(x)$。又因为定理 1 中表 $\hat{V}(x)$ 可以初始化为任意有限值，所以对所有 $x\in X_n$，当 $n\to\infty$ 时，$\hat{V}_n(x)$ 收敛到 $V(x)$。定理得证。

4 实验及结果分析

4.1 Mountain Car 实验介绍

Mountain Car 问题是强化学习中的一个经典问题，如图 5 所示，其中 S 为最低点，G 为右端最高点，A 为左端最高点。小车的任务是在动力不足的条件下，从 S 点以尽量短的时间运动到 G 点。系统的状态由 2 个连续变量 y 和 v 表示，其中 y 为小车的水平位移，v 为小车的水平速度，MC 问题的状态空间

$$\left\{\boldsymbol{x}\,|\,\boldsymbol{x}=[y,v]^{\mathrm{T}}\right\}\subseteq R^2,\ y\in[-1.2,\ 0.5],v\in[-0.07,\ 0.07]$$

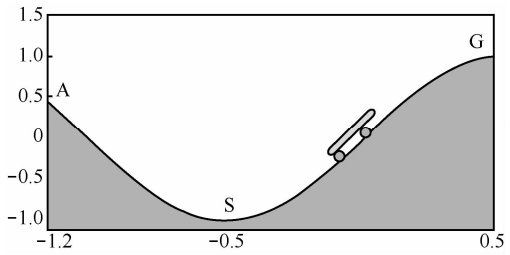

图 5 MC 问题

当小车位于 S 点、G 点和 A 时，y 的取值分别为−0.5、0.5 和−1.2。控制量为小车所受水平方向的力，取 3 个离散值，即+1、0 和−1，分别代表全油门向前、零油门和全油门向后 3 个控制行为。仿真实验中，系统的动力学特性描述为

$$\begin{cases}\dot{v}=\mathrm{bound}[v+0.001u-g\cos 3y]\\ \dot{y}=\mathrm{bound}[y+\dot{v}]\end{cases}$$

其中，g=0.002 5 为与重力有关的系数，u 为控制量。目标是在没有任何模型先验知识的前提下，控制小车从任意初始点以最短时间运动到 G 点。该控制问题可以用一个确定性 MDP 来建模，奖赏函数为

$$r_t=\begin{cases}-1, & y<0.5\\ 0, & y\geqslant 0.5\end{cases}$$

这是一个惩罚型的奖赏函数，其中 t 表示当前奖赏是在时间步 t 时获得。

4.2 实验结果及分析

图 6 是在不同的 tile 数目下，用 ATC 算法到达收敛后，分析它对存储空间的利用率。这里认为，当 2 个相邻的 tile 的权值差小于各自最小 Bellman 差时，则 2 个 tile 是存在误划分的。从图 6 中可以看出，ATC 算法在对状态空间进行划分时产生的误划分 tile 的数目会随着 tile 数目的增加而成比例地增加，误划分比例约为 0.5。为了进一步研究 MATC 算法对状态空间大小的影响，对 ATC 算法收敛后的状态空间再次运用 MATC 算法进行合并，结果显示可以节约近一半的状态空间。这对于 Mountain Car 这类连续空间问题十分关键，可以在很大程度上减弱维数灾。

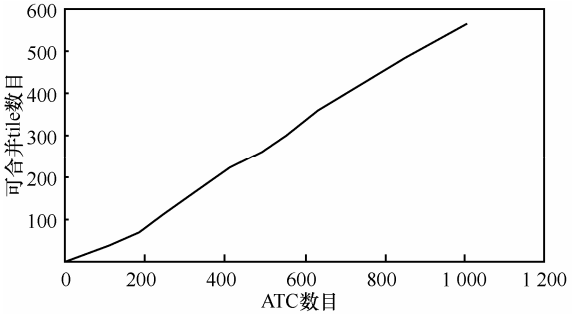

图 6 自适应 tile coding 可合并 tile 数目

表 1 中显示了在取不同最大 tile 数目的情况，ATC 算法和 MATC 算法到达收敛情节数的比较。MATC 算法的收敛速度明显要比 ATC 算法快很多，并且这种收敛性的差异随着最大 tile 数目的不断增长而逐渐变大。由此可以看出，ATC 算法确实在学习过程中产生许多误划分，为了使这些误划分达到收敛也在很大程度上减慢了算法的收敛速度。ATC 算法划分的 tile 数目越多，产生的误划分数也就越多，收敛速度减慢的幅度也就越大。MATC 算法能够减弱这些误划分对于收敛性的影响，因为 MATC 算法允许对产生的误划分进行弥补从而可以大幅减少最终的误划分数目，加快收敛速度。误划分的数目减少也就意味着，在同样的 tile 数目的情况下有区分意义的划分数目增多，能更好地表示状态空间。

表 1 算法收敛性比较

tile	ATC 情节数	MATC 情节数
60	593	168
100	735	247
200	2 200	892
300	6 492	1 217

图 7 给出了 ATC 算法和 MATC 算法的学习过程

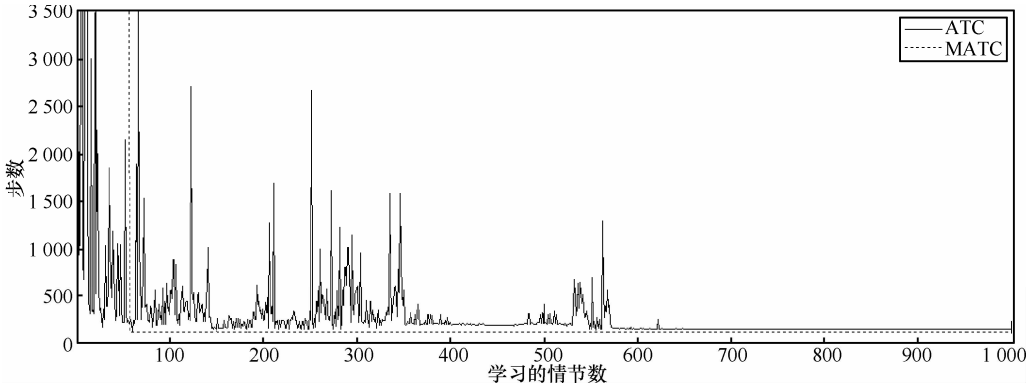

图 7　ATC 和 MATC 学习效果比较

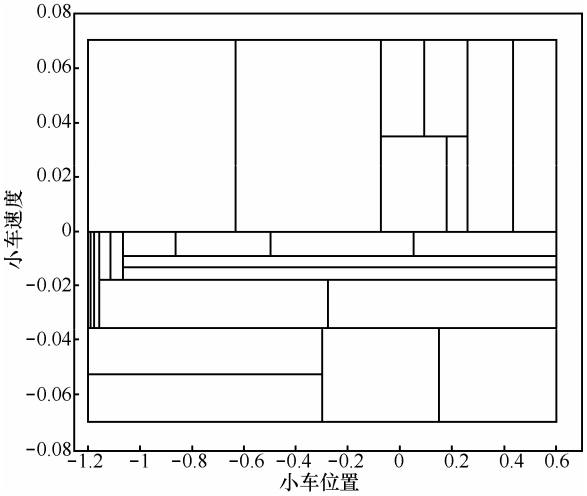

图 8　MATC 划分的状态空间

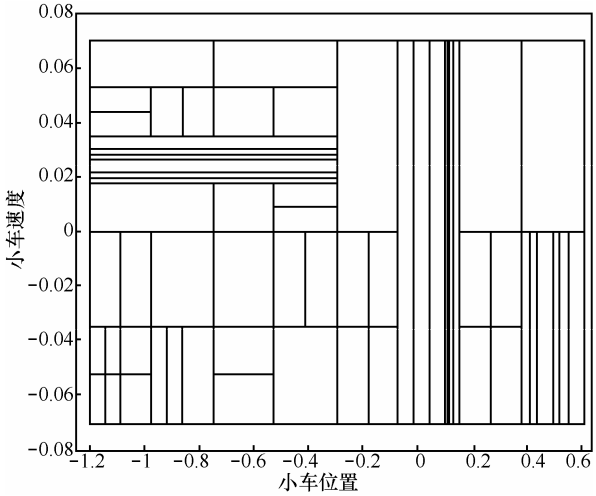

图 9　ATC 划分的状态空间

比较。学习开始时，由于 tile 数目还在不断变化，2 种算法都表现出了很大的振幅。但 MATC 算法在 50 个情节附近就收敛了，而 ATC 算法在 600 个情节附近才能收敛，并且 MATC 算法收敛到的步数比 ATC 算法收敛到的步数要少，因此 MATC 收敛结果更好。此外，实验中 MATC 算法选取的最大 tile 数目为 35，而 ATC 算法选取的最大 tile 数目为 60。图 8 和图 9 分别为 MATC 算法、ATC 算法划分后的状态空间，从图中可以看出，MACT 算法分割的状态空间并不像 ATC 算法分割后的状态空间那么规整，允许 tile 之间不对齐的错落分布，有更强的表征能力，能在给定的最大 tile 数目下更精确地逼近状态函数的实际分布。由此可见，MATC 算法的划分数更少，但是效果更好，对连续状态空间的自适应划分进行了优化，其收敛结果与收敛速度都有所提高。

5　结束语

本文针对自适应 tile coding 算法存在误划分，会导致存储空间增大以及学习速率减慢的问题，提出一种支持合并的自适应 tile coding 算法——MATC。该算法在学习过程中，根据一定的阈值条件可以对相邻的区域进行合并，从而消除误划分所产生的不良影响。本文从理论上分析了 MATC 算法的优越性，并通过仿真实验，对它的有效性进行验证。实验表明，这种算法不仅能大大缩减存储空间，而且有利于提高学习效率。

对于 tile 的合并条件，本文采用了一种简单的阈值条件，即当相邻的整边重合 tile 的权值差低于某一固定阈值并达到一定次数后进行合并，还存在许多其他可以用来合并的条件。如何找到更好的合并策略，这还有待于进一步的研究。

参考文献：

[1] SUTTON R S, BARTO A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 1998.

[2] LIN C S, KIM H. Selection of learning parameters for CMAC-based

adaptive critic learning[J]. IEEE Trans Neural Networks, 1999, 6(3):642-647.

[3] PELLEG D, MOORE A. *X*-means: extending *K*-Means with efficient estimation of the number of clusters[A]. Proc of the 17th International Conf on Machine Learning[C]. Boston:Morgan Kaufmann Press, 2000. 727-734.

[4] PELLEG D, MOORE A. Accelerating exact *k*-means algorithms with geometric reasoning[A]. Proc of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. 1999. 277-281.

[5] 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.
CHEN Z H, WEN F, NIE J B, *et al*. A reinforcement learning method based on node-growing *k*-means cluster algorithm[J]. Journal of Computer Research and Development, 2006, 43(4):661-666.

[6] 文锋, 陈宗海, 卓睿等. 连续状态自适应离散化基于 *K*-均值聚类的强化学习方法[J]. 控制与决策, 2006, 21(2):143-147.
WEN F, CHEN Z H, ZHUO R, *et al*. Reinforcement learning method of continuous state adaptively discretized based on *K*-means clustering[J]. Control and Decision, 2006, 21(2):143-147.

[7] 顾冬雷, 陈卫东, 席裕庚. 一种基于增强学习的自适应控制方法[J]. 控制与决策, 2002, 17(4):473-479.
GU D L, CHEN W D, XI Y G. A novel adaptive control algorithm based on reinforcement learning[J]. Control and Decision, 2002, 17(4): 473-479.

[8] MOORE A W, ATKESON C G. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state spaces[J]. Machine Learning, 1995, 21(3):199-233.

[9] UTHER W T B, VELOSO M M. Tree based discretization for continuous state space reinforcement learning[A]. AAAI'98[C]. Madison, Wisconsin, United States, 1998.

[10] SHERSTOV A A, STONE P. Function Approximation Via Tile Coding: Automating Parameter Choice Abstraction, Reformulation and Approximation[M]. Springer Berlin Heidelberg, 2005.194-205.

[11] WHITESON S, TAYLOR M E, STONE P. Adaptive tile Coding for Value Function Approximation[M]. Computer Science Department, University of Texas at Austin, 2007.

[12] WHITESON S, STONE P. Evolutionary function approximation for reinforcement learning [J]. The Journal of Machine Learning Research, 2006, 7: 877-917.

[13] NOKHBEH-ZAEEM M, KHASHABI D, TALEBI H A, *et al*. Adaptive tiled neural networks[A]. 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)[C]. 2011.2543-2548.

[14] LIN S, WRIGHT R. Evolutionary tile coding: an automated state abstraction algorithm for reinforcement learning[A]. AAAI Workshops[C]. 2010.

作者简介：

施梦宇（1989-），男，江苏淮安人，苏州大学硕士生，主要研究方向为强化学习。

刘全（1969-），男，内蒙古牙克石人，苏州大学教授、博士生导师，主要研究方向为强化学习、智能信息处理和自动推理。

傅启明（1985-），男，江苏淮安人，苏州大学博士生，主要研究方向为强化学习、贝叶斯推理和遗传算法。