计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200700044

# 基于相对熵的元逆强化学习方法

吴少波[1,2,3]　傅启明[1,2,3]　陈建平[2,3]　吴宏杰[1,2]　陆　悠[1,2]
1 苏州科技大学电子与信息工程学院　江苏 苏州 215009
2 苏州科技大学江苏省建筑智慧节能重点实验室　江苏 苏州 215009
3 苏州科技大学苏州市移动网络技术与应用重点实验室　江苏 苏州 215009
(wushaobo_1@163.com)

**摘　要**　针对传统逆强化学习算法在缺少足够专家演示样本以及状态转移概率未知的情况下,求解奖赏函数速度慢、精度低甚至无法求解的问题,提出一种基于相对熵的元逆强化学习方法。利用元学习方法,结合与目标任务同分布的一组元训练集,构建目标任务学习先验,在无模型强化学习问题中,采用相对熵概率模型对奖赏函数进行建模,并结合所构建的先验,实现利用目标任务少量样本快速求解目标任务奖赏函数的目的。将所提算法与 REIRL 算法应用于经典的 Gridworld 和 Object World 问题,实验表明,在目标任务缺少足够数目的专家演示样本和状态转移概率信息的情况下,所提算法仍能较好地求解奖赏函数。

**关键词**:逆强化学习;元学习;奖赏函数;相对熵;梯度下降
**中图法分类号**　TP311

## Meta-inverse Reinforcement Learning Method Based on Relative Entropy

WU Shao-bo[1,2,3],FU Qi-ming[1,2,3],CHEN Jian-ping[2,3],WU Hong-jie[1,2] and LU You[1,2]
1 School of Electronics and Information Engineering,Suzhou University of Science and Technology,Suzhou,Jiangsu 215009,China
2 Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency,Suzhou University of Science and Technology,Suzhou,Jiangsu 215009,China
3 Suzhou Key Laboratory of Mobile Network Technology and Application,Suzhou University of Science and Technology,Suzhou,Jiangsu 215009,China

**Abstract**　Aiming at the problem that traditional inverse reinforcement learning algorithms are slow,imprecise,or even unsolvable when solving the reward function owing to insufficient expert demonstration samples and unknown state transition probabilitie,a meta-reinforcement learning method based on relative entropy is proposed. Using meta-learning methods,the target task learning prior is constructed by integrating a set of meta-training sets that meet the same distribution as the target task. In the model-free reinforcement learning problem,the relative entropy probability model is used to model the reward function and combined with the prior to achieve the goal of quickly solving the reward function of the target task using a small number of samples of the target task. The proposed algorithm and the RE IRL algorithm are applied to the classic Gridworld and Object World problems. Experiments show that the proposed algorithm can still solve the reward function better when the target task lacks a sufficient number of expert demonstration samples and state transition probabilities information.

**Keywords**　Inverse reinforcement learning,Meta-learning,Reward function,Relative entropy,Gradient decent

## 1　引言

强化学习(Reinforcement Learning,RL)是一种从环境状态到行为映射的学习,通过智能体(agent)不断地与环境交互得到的奖赏修改自身的策略,经过数次迭代学习后,智能体最终学到完成当前任务的最优策略[1]。作为强化学习问题中一个关键的因素,奖赏函数必须能够准确地描述当前任务,往往需要人为设定。然而,面对现实世界中的复杂任务时,人为设定奖赏函数是一项充满挑战性的任务,往往带有很大的主观性和经验性,并且由于奖赏函数设置的不同,智能体最终学习到的最优策略之间也会存在较大差异,不恰当的奖赏函数甚至会导致强化学习算法最终无法收敛。以机器人学习骑自行车为例,很难用一组具体的变量作为描述骑车任务的奖赏,机器人通过自身的经验难以直接完成这个任务,但是通过学习人类骑车的演示可以完成这个任务,也就是说,智能体可以通过恢复“专家”演示对应的奖赏函数,然后利用该奖赏来学习

到稿日期:2020-07-08　返修日期:2021-01-09　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金项目(61876217,61876121,61772357,61750110519,61772355,61702055,61672371);江苏省重点研发计划项目(BE2017663)
This work was supported by the National Natural Science Foundation of China(61876217,61876121,61772357,61750110519,61772355,61702055,61672371) and Primary Research and Development Plan of Jiangsu Province(BE2017663).
通信作者:傅启明(fqm_1@126.com)

到最优策略，这种方法就是逆向强化学习(inverse reinforcement learning)[2]。

逆向强化学习是一种旨在从任务的演示中推断奖赏函数来解决奖赏函数难以人为设定的问题的方法。此前关于逆强化学习问题的处理方法大体可分为基于最大边际形式化和基于概率模型的形式化两类。Abbeel 等通过收集所求解任务的专家演示样本，结合所构建的线性奖赏函数模型，利用学徒学习的方法求解奖赏函数参数，进一步恢复所求解任务的奖赏函数，最后利用正向强化学习方法求解最优策略[3]。Ratliff 等扩展了最大边际规划(Maximum Margin Planning，MMP)框架，将其用于学习更强大的非线性奖赏函数，并且通过获得最佳轨迹取代获得最佳策略的方法，避免寻找最佳策略的复杂过程[4]。为了克服基于最大边际形式化的逆强化学习固有的歧义问题，即同一组专家演示对应多个奖赏函数，此时通过专家演示学习到的奖赏函数往往带有某种随机的偏好，Ziebart 等通过最大熵概率模型来构造奖赏函数，将求解奖赏函数的问题转化为求解专家策略所对应的所有轨迹的概率分布，通过求解熵最大的概率分布式来避免歧义问题[5]。针对最大熵逆强化学习中状态转移概率必须已知的问题，Boularias 等在最大熵模型上引入相对熵的概念，通过求解与已知熵最大分布轨迹最为接近的轨迹分布构建奖赏函数模型，并通过重要性采样的方法表示后续梯度下降过程中的梯度，实现了在状态转移函数未知的情况下对奖赏函数的学习[6]。相比带有主观经验的人工设定奖赏函数的方法，逆向强化学习通过专家演示轨迹这一数据驱动的方法来确定奖赏函数，从中得出的最优策略在实际应用中效果往往好于前者。然而，在实践中，由于提供包含现实世界任务中常见的可变性的专家演示的代价较为昂贵，很多现实任务的专家演示数量较少，在少量样本的情况下通过函数值逼近的方法恢复奖赏函数的速度非常缓慢，因此本文结合元学习的方法来解决该问题。

元学习(meta-learning)是一种系统地观察不同机器学习方法如何在广泛的学习任务中执行，然后从这种经验或元数据中学习，以比其他方法更快的速度学习新任务的机器学习概念。当智能体学习新任务时，将之前从相关任务中学习到的技能作为先验，随着每一项新任务的学习，智能体学习新任务变得更容易，需要更少的例子和尝试，出现错误的概率更小。简而言之，元学习是一种实现智能体跨任务学习的机器学习方法。此前关于元学习问题的方法主要可以分为基于记忆机制的方法[7]、学习优化器和初始化的方法[8]和比较学习度量空间中新任务数据点的方法[9]。

本文关注的问题是如何有效地利用元训练集得到关于奖赏函数的学习先验，而不是提高智能体学习单个任务的奖赏函数的能力，从而能够在少量专家演示的情况下学习新任务的奖赏函数。我们的任务是发现不同任务之间的共同结构，并以一种可用于从一些演示中恢复奖赏函数的方式对该结构进行编码。具体地说，在面对一个拥有少量专家演示的新任务时，我们假设可以访问一组相关任务及这些任务所需的足够多的专家演示，我们称之为元训练集，从这些任务中，智能体学习了一组关于奖赏函数的参数，并将这组参数用于初始化目标任务的奖赏函数，有效实现了对目标任务的少样本学习。同时，针对无模型强化学习中状态转移概率未知的情况，本文采用相对熵概率模型对奖赏函数进行建模，并通过重要性采样的方法来求解奖赏函数参数的梯度，将从元训练任务集中学习到的奖赏函数参数作为初始值并对参数进行梯度下降，实现在少量专家演示情况下对新目标任务的奖赏函数的求解。

## 2 相关理论

### 2.1 基于最大熵的逆强化学习方法

在强化学习问题中，标准的马尔可夫决策过程通常可以表示为一个四元组：$M=(S,A,r,P)$，其中 $S$ 表示所有状态的集合，$A$ 表示所有动作的集合，$r$ 表示奖赏函数，$P$ 表示状态转移函数。通常情况下，强化学习的目标是最大化带有折扣的期望回报，该期望回报可以表示为 $R=\sum_{t=1}^{T}\gamma^{t-1}r(s_t,a_t)$，其中 $\gamma$ 表示折扣因子，$\gamma\in(0,1)$。

在逆强化学习问题中，奖励函数是未知的，但我们可以访问一组专家演示：$D=\{\tau_1,\tau_2\cdots,\tau_k\}$，其中 $\tau_k=\{s_1,a_1,\cdots,s_k,a_k\}$。逆强化学习问题的目标是从一组专家演示中恢复未知的奖赏函数 $r$。本文针对线性结构的奖赏函数，将奖赏函数表示为 $r=\theta_1 f_1+\theta_2 f_2+\cdots+\theta_k f_k$，其中 $\{f_1,f_2,\cdots,f_k\}$ 表示当前状态的特征函数。

由于每个策略对应的所有轨迹的概率分布是不同的，因此求解专家策略对应的奖赏函数可以进一步转化为求解该策略对应的所有可能轨迹的概率分布，Ziebart 等提出的最大熵框架将逆强化学习问题转化为求解如下条件极值问题：

$$\max -p\log p$$

$$\text{s.t.}\sum_{\tau_i}P(\tau_i)f=\tilde{f}$$

$$\sum P=1$$

此时的约束条件为：$\sum_{\tau_i}P(\tau_i)f=\tilde{f}$，其中 $\tau_i$ 表示强化学习问题中的轨迹，$f$ 表示当前轨迹对应的所有状态的特征期望之和，$\tilde{f}$ 表示所有专家轨迹的特征期望。由于熵最大的性质，求解该条件极值问题得到的是除约束条件外不带任何偏好的专家策略对应的奖赏函数。后续通过梯度下降的方法求解得出奖赏函数参数。

### 2.2 元学习

元学习算法的目标是提高优化智能体学习新任务的能力，其挑战在于以一种系统的、数据驱动的方式从以前的经验中学习，即元学习的重点是将智能体的学习能力泛化到新的任务中，而不是泛化到旧任务中的新数据点上。在元学习设定中存在两个不相交的任务集：元训练集 $T^{\text{tr}}=\{T_1,T_2,\cdots,T_N\}$ 和元测试集 $T^{\text{test}}=\{T_1,T_2,\cdots,T_K\}$，它们均来自 $P(T)$ 的分布。在元训练期间，智能体试图学习元训练集中任务的结构，这样当它面对一个测试任务时，可以利用这个结构从有限样本的新任务中有效地学习。

具体来说，$f_\theta$ 表示学习者，任务 $X$ 和 $Y$ 的训练样本集表示为 $X_T^{\text{tr}}=\{x_1,x_2,\cdots,x_k\}$，$Y_T^{\text{tr}}=\{y_1,y_2,\cdots,y_k\}$；测试集表示为 $X_T^{\text{test}}=\{x_1,x_2,\cdots,x_{k'}\}$，$Y_T^{\text{test}}=\{y_1,y_2,\cdots,y_{k'}\}$。元学习的一种方法是直接用表达模型(如基于任务训练数据和测试任务输入的递归或循环神经网络)对元学习器进行参数化：

$f_\theta(Y|X_T^{\text{test}},X_T^{\text{tr}},Y_T^{\text{tr}})$，该模型在所有任务上都使用最大似然估计的方法进行优化。在这种元学习方法中，由于神经网络被用作通用的函数逼近器，因此任务之间的任何所需结构都可以通过这种方式进行隐式编码。

相比这种学习单一任务黑盒函数的方法，元学习的另一类方法是学习学习过程中的组成部分，比如初始化。本文扩展了 Finn 等提出的基于深度神经网络参数的模型无关元学习方法(MAML)[8]，该方法学习一个初始化，并通过梯度下降来对新任务参数进行快速自适应。具体地，在监督学习中，给定一个损失函数 $L(\theta,X_T,Y_T)$(如交叉熵)，MAML 算法执行如下优化：

$$\min_\theta \sum_T L(\phi_T,X_T^{\text{test}},Y_T^{\text{test}})$$
$$=\min_\theta \sum_T L(\theta-\alpha\nabla_\theta L(\theta,X_T^{\text{tr}},Y_T^{\text{tr}}),X_T^{\text{test}},Y_T^{\text{test}})$$

该优化器使用一组参数 $\theta$ 作为初始参数，利用测试集 $X_T^{\text{test}}$ 上的损失函数帮助学习器在训练集 $X_T^{\text{tr}}$ 上通过梯度下降法学习参数 $\theta$(步长为 $\alpha$)。本文将该算法与基于概率模型的逆强化学习算法相结合，针对线性结构的奖赏函数 $r=\theta_1 f_1+\theta_2 f_2+\cdots+\theta_k f_k$，通过构建先验的方式为少样本逆强化学习任务中奖赏函数参数 $\theta$ 的梯度下降过程确定合适的初始值，确保智能体在目标任务缺少足够专家演示样本的条件下能够快速稳定地恢复奖赏函数。

## 3 基于相对熵的元逆强化学习方法

逆强化学习的目的是通过专家样本来恢复奖赏函数，在此过程中涉及复杂的迭代计算，但是缺少专家样本会导致学习速率缓慢甚至最终无法收敛的问题，同时，针对无模型强化学习情况下状态转移概率未知的问题，传统逆强化学习方法无法对奖赏函数进行建模，因此，针对这两个问题本文提出一种基于相对熵的元逆强化学习方法。

首先，针对无模型强化学习中状态转移概率未知、基于最大熵的逆强化学习方法无法对奖赏函数进行建模的情况，本文采用相对熵概率模型对奖赏函数进行建模，通过重要性采样的方法求解出参数梯度下降过程中的梯度，然后在初始参数 $\theta$ 的设置上，结合 Finn 等提出的模型无关快速自适应深度神经网络的元学习方法(MAML)，通过为满足同分布 $P(T)$ 的元训练集和目标任务构建先验，来确定奖赏函数参数梯度下降算法中的初始值，使智能体能够在少量专家样本的情况下对新任务的奖赏函数进行快速自适应。基于相对熵的元逆强化学习框架如图 1 所示。

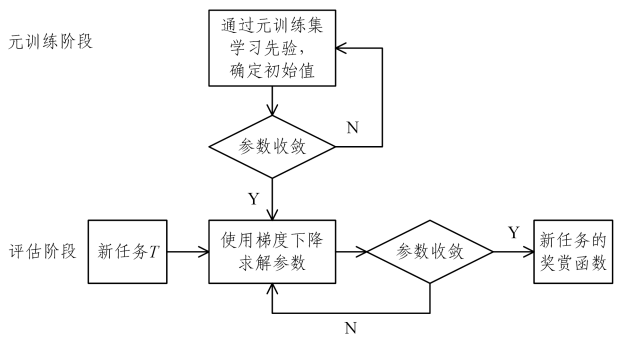


图 1 基于相对熵的元逆强化学习框架

Fig. 1 Framework of meta-inverse reinforcement learning based on relative entropy

### 3.1 基于相对熵的奖赏函数建模

为了能够有效地求解奖赏函数，在图 1 对参数 $\theta$ 的迭代过程中需要找到一个合适的目标函数，并对参数进行梯度下降来获取奖赏函数。2.1 节中基于最大熵模型的奖赏函数求解过程中需要利用轨迹的概率 $P(\tau)$，该概率可表示为：

$$P(\tau|\theta,T)\propto d_0(s_1)\exp(\sum_{i=1}^{k}\theta_i f_i^\tau)\prod_{t=1}^{H}T(s_{t+1}|s_t,a_t)$$

求解该式的前提是系统的状态转移概率 $T(s_{t+1}|s_t,a_t)$ 是已知的。针对无模型强化学习问题中该概率未知的情况，我们在建模过程中引入相对熵的概念，将求解熵最大的轨迹概率分布问题转化为求解与最大熵分布之间差异最小即相对熵最小的轨迹概率分布问题。由于均匀分布的系统为熵最大的系统，设 $Q(\tau)$ 为利用均匀分布策略产生的轨迹分布，需要求解的概率分布为 $P(\tau)$，则该问题可形式化为：

$$\begin{aligned}&\min\ \sum_\tau P(\tau)\ln\frac{P(\tau)}{Q(\tau)}\\&\text{s.t. for all } i\in\{1,2,\cdots,k\}:\\&\quad\sum_\tau P(\tau)f_i^\tau=\tilde{f}_i\\&\quad\sum_\tau P(\tau)=1\\&\quad\forall\tau\in T:P(\tau)\geqslant 0\end{aligned}\tag{1}$$

利用拉格朗日乘子法和 KKT 条件，该优化问题可以转化为：

$$\min L(P,\theta,\eta)=\sum_\tau P(\tau)\ln\frac{P(\tau)}{Q(\tau)}-\sum_{i=1}^{k}\theta_i(\sum_\tau P(\tau)f_i^\tau-\tilde{f}_i)+\eta(\sum_\tau P(\tau)-1)\tag{2}$$

在式(2)中对 $P$ 求偏导并令其为 0，可得：

$$\partial_{P(\tau)}L(P,\theta,\eta)=\ln\frac{P(\tau)}{Q(\tau)}-\sum_{i=1}^{k}\theta_i f_i^\tau+\eta+1=0\tag{3}$$

$$P(\tau)=Q(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^\tau-\eta-1)\tag{4}$$

因为 $\sum_\tau P(\tau)=1$，所以式(4)中的 $\exp(\eta+1)$ 可看作归一化常量：

$$\exp(\eta+1)=\sum_\tau Q(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^\tau)\overset{def}{=}Z(\theta)\tag{5}$$

最终可以得到每条轨迹对应的概率为：

$$P(\tau|\theta)=\frac{1}{Z(\theta)}Q(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^\tau)\tag{6}$$

由此可得该最优化问题的拉格朗日对偶函数：

$$g(\theta)=\sum_{i=1}^{k}\theta_i\tilde{f}_i-\ln Z(\theta)\tag{7}$$

此函数即为该模型中对奖赏函数参数 $\theta$ 进行梯度下降的目标函数，即对参数 $\theta$ 朝对偶函数 $g(\theta)$ 最大值的梯度方向进行梯度下降。由于对偶函数 $g(\theta)$ 在 $\theta_i\neq 0$ 时处处为凹并可微，所以采用次梯度上升的方法最大化 $g(\theta)$，次梯度为：

$$\frac{\partial g(\theta)}{\partial\theta_i}=\tilde{f}_i-\sum_\tau P(\tau|\theta)f_i^\tau\tag{8}$$

由于用来计算 $P(\tau)$ 和 $Q(\tau)$ 的状态转移函数 $T$ 是未知的，因此该梯度无法被直接求解。本文采用基于经验的梯度估计方法来求解，该方法通过对遵循任意策略 $\pi$ 的轨迹进行重要性采样，近似求解出梯度 $\frac{\partial g(\theta)}{\partial\theta_i}$。具体来说，可以将函数 $Q(\tau)$ 分解为 $Q(\tau)=D(\tau)U(\tau)$，其中 $U(\tau)=\prod_{t=1}^{H}P(a_t|s_t)$ 表示轨迹 $\tau=s_1a_1,\cdots,s_Ha_H$ 中动作选择的联合概率，$D(\tau)=d_0$

$(s_1)\prod_{t=1}^{H}T(s_t,a_t,s_{t+1})$表示轨迹 $\tau$ 中状态转移联合概率，因此式(6)可表示为：

$$P(\tau|\theta)=\frac{D(\tau)U(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^{\tau})}{\sum_{\tau}D(\tau)U(\tau)\exp(\sum_{i=1}^{k}\theta_i f_i^{\tau})} \tag{9}$$

式(8)中的$\sum_{\tau}P(\tau|\theta)f_i^{\tau}$可利用重要性采样的方法，通过执行给定的策略 $\pi$ 并采样 $N$ 条轨迹 $\tau_N^{\pi}$ 近似求解。式(8)具体的求解过程如下：

$$\begin{aligned}\frac{\partial g(\theta)}{\partial\theta_i}&=\tilde{f}_i-\frac{1}{N}\sum_{\tau\in T_N^{\pi}}\frac{P(\tau|\theta)}{D(\tau)\pi(\tau)}f_i^{\pi}\\&=\tilde{f}_i-\frac{1}{N}\frac{\sum_{\tau\in T_N^{\pi}}\frac{D(\tau)U(\tau)\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}{D(\tau)\pi(\tau)}f_i^{\tau}}{\sum_{\tau}D(\tau)U(\tau)\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}\\&=\tilde{f}_i-\frac{\frac{1}{N}\sum_{\tau\in T_N^{\pi}}\frac{D(\tau)U(\tau)\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}{D(\tau)\pi(\tau)}f_i^{\tau}}{\frac{1}{N}\sum_{\tau\in T_N^{\pi}}\frac{D(\tau)U(\tau)\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}{D(\tau)\pi(\tau)}}\\&=\tilde{f}_i-\frac{\sum_{\tau\in T_N^{\pi}}\frac{\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}{\pi(\tau)}f_i^{\tau}}{\sum_{\tau\in T_N^{\pi}}\frac{\exp(\sum_{j=1}^{k}\theta_j f_j^{\tau})}{\pi(\tau)}}\\&=\frac{\partial r_\theta}{\partial\theta}[\mu_D-E[\mu]]\\&=\frac{\partial r_\theta}{\partial\theta}\left[\mu_D-\frac{\sum_{\tau\in T_N^{\pi}}\frac{\exp(\sum_{j=1}^{k}\theta\mu_\tau)}{\pi(\tau)}\mu_\tau}{\sum_{\tau\in T_N^{\pi}}\frac{\exp(\sum_{j=1}^{k}\theta\mu_\tau)}{\pi(\tau)}}\right]\end{aligned} \tag{10}$$

其中，$\mu_D$ 表示所有专家轨迹中对每个状态访问次数的平均值；$E[\mu]$表示在当前奖赏函数参数 $\theta$ 对应的策略下，对每个状态访问次数的期望值；$\mu_\tau$ 表示智能体通过执行给定策略并采样得到的轨迹中对每个状态的访问次数。

### 3.2 基于相对熵的元逆强化学习算法

基于相对熵的元逆强化学习算法(ReRnt-MIRL)的目标是学习如何在多个任务之间学习奖赏函数，以便模型可以通过少量专家演示样本来推断目标任务的奖赏函数。直观地说，我们可以把该问题看作在专家演示之前学习关于奖赏函数的先验，当只给出少量新任务的专家演示时，我们可以把学习的先验与新数据结合起来，从而有效地确定专家的意图，即恢复出新任务的奖赏函数。由于与特定任务相关的奖赏函数的空间要比所有可能的奖赏函数空间小得多，所以这种先验在逆强化学习问题中能够有效地加快奖赏函数参数的收敛速度，有助于在少样本设定下对奖赏函数进行求解。

首先，通过对特定任务 $T$ 的奖赏函数 $r_\theta$ 定义损失函数 $L_T(\theta)$，将奖赏函数好坏的概念形式化，本文采用 3.1 节中的相对熵模型对此进行建模，得到如下梯度：

$$\nabla_\theta L_T(\theta)=\frac{\partial r_\theta}{\partial\theta}[\mu_{D_T}-E[\mu_T]] \tag{11}$$

在元训练阶段，我们有一组任务$\{T_i;i=1,\cdots,N\}$，每个任务 $T_i$ 都有一组专家演示 $D_T=\{\tau_1,\cdots,\tau_k\}$，并将其分为不相交的训练集 $D_T^{\mathrm{tr}}$ 和测试集 $D_T^{\mathrm{test}}$。在元训练过程中，这些演示集被用来编码奖赏函数通用的结构，为满足 $P(T)$分布的任务构建参数初始值，这样本文提出的基于相对熵的奖赏函数模型就可以快速地从少量专家演示中获得新任务的奖赏函数。在评估阶段，本文算法能够从新任务的少量专家演示中恢复出奖赏函数 $r_\phi(s)$的参数，为了满足模型无关元学习方法(MAML)的设定，目标任务与元训练集中的任务均满足 $P(T)$的分布。该算法可形式化为如下最优化问题：

$$\min_\theta\sum_{i=1}^{N}L_{T_i}^{\mathrm{test}}(\phi_{T_i})=\sum_{i=1}^{N}L_{T_i}^{\mathrm{test}}(\theta-\alpha\nabla_\theta L_{T_i}^{\mathrm{tr}}(\theta))$$

通过优化该损失函数获得目标任务奖赏函数参数的先验。本文算法具体如算法 1 所示。

**算法 1** 基于相对熵的元逆强化学习方法(ReEnt-MIRL)

输入：元训练任务集$\{T\}^{\text{meta-train}}$，每个任务的专家演示集$\{\tau_1,\cdots,\tau_N\}$，步长参数 $\alpha,\beta,\lambda$；目标任务专家演示$\{\tau_1,\cdots,\tau_k\}$

输出：目标任务奖赏函数参数 $\theta$

1. function 基于相对熵的元逆强化学习梯度 ReEntMIRL-GRAD($r_\theta$, T, D, $\pi$)
2. ＃枚举法统计专家演示状态访问次数
3. $\mu_D$=STATE-VISITATIONS-TRAJ(T, D)
4. 执行给定策略 $\pi$ 并采样 N 条轨迹
5. ＃枚举法统计采样得到的每个轨迹的状态访问次数
6. $\mu_\tau$=STATE-VISITATIONS(T, $\tau$)
7. 根据式(10)计算 E[$\mu$]
8. $\partial L/\partial r_\theta=\mu_D-E[\mu]$
9. Return $\partial L/\partial r_\theta$
10. end function
11. 随机初始化奖赏函数参数 $\theta$，设 $\Delta=\infty$
12. while $\Delta>10^{-3}$：
13. 从满足 P(T)的元训练任务集$\{T\}^{\text{meta-train}}$中采样得到一组任务 $T_i$
14. for all $T_i$ do
15. 采样得到专家演示训练集$\{\tau_1,\cdots,\tau_k\}$
16. $\frac{\partial L_{T_i}^{\mathrm{tr}}(\theta)}{\partial r_\theta}$=ReEntMIRL-GRAD($r_\theta$, T, $D^{\mathrm{tr}}$, $\pi$)
17. 通过链式法则计算得出$\nabla_\theta L_{T_i}^{\mathrm{tr}}(\theta)$
18. ＃用梯度下降法对参数 $\theta$ 进行更新
19. $\phi_{T_i}=\theta+\alpha\nabla_\theta L_{T_i}^{\mathrm{tr}}(\theta)$
20. 将专家演示集中剩下的轨迹作为测试集$\{\tau_1',\cdots,\tau_{n-k}'\}$
21. $\frac{\partial L_{T_i}^{\mathrm{test}}(\theta)}{\partial r_\theta}$=ReEntMIRL-GRAD($r_{\phi_{T_i}}$, T, $D^{\mathrm{test}}$, $\pi$)
22. 通过链式法则计算得出$\nabla_\theta L_{T_i}^{\mathrm{test}}(\theta)$
23. end for
24. $\theta'\leftarrow\theta+\beta\sum_i\nabla_\theta L_{T_i}^{\mathrm{test}}$
25. $\Delta=\|\theta'-\theta\|_2$
26. end while
27. 返回收敛后的参数 $\theta$
28. ＃利用元训练过程中返回的参数 $\theta$ 作为初始值对参数 $\theta$ 进行梯度下降至参数收敛
29. $\nabla L(\theta)=\frac{\partial r_\theta}{\partial\theta}$ * ReEntMIRL-GRAD($r_\theta$, T, D, $\pi$)
30. $\theta\leftarrow\theta+\lambda\nabla L(\theta)$

## 4 实验

为了验证所提算法的有效性，将本文提出的 ReEnt-

MIRL 算法与 Re-IRL[6] 算法应用于 Gridworld 和 Object World 两种经典问题，分别在简单的导航任务和带有障碍物的多目的地导航任务两种格子世界场景下比较两种算法在少样本条件下求解奖赏函数的性能。

### 4.1 Girdworld 问题

#### 4.1.1 实验描述

如图 2 所示，Gridworld 问题是一个 格子的网格世界问题。状态对应于 Agent 在网格中的位置，Agent 从网格左下角出发，在每一个状态下可以进行 4 种操作：上、下、左、右。此外，由于网格世界中的干扰因素即风的存在，Agent 行动成功的概率是 0.7，失败的结果是 Agent 随机地转变到一个相邻的状态。

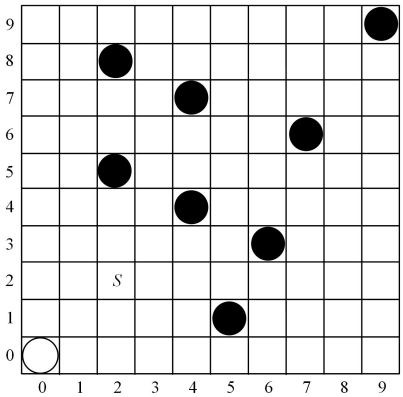


图 2 10 * 10 格子世界

Fig. 2 10 * 10 Gridworld

此外，网格中存在 $n$ 个目标终点（黑色圆），从起点（白色圆）出发至每个目标终点构成一个任务，人为设定目标终点状态奖赏函数为 1，其余所有状态的奖赏函数均为 0。对每个任务，通过正向强化学习算法得到最优策略，即专家策略，并采样得到 15 条专家轨迹作为样本，从而构建出 $n$ 个还原人为设定的奖赏函数的逆强化学习任务。同时，将最终的实验任务设定为从左下角出发，终点在格子世界的右上角，用相同的方法得到 3 条专家轨迹作为专家样本，构建出少样本条件下还原奖赏函数的逆强化学习任务。实验中设定折扣率为 0.99，两个梯度下降步长参数 $\alpha$ 和 $\beta$ 均设为 0.01，最大迭代次数设为 300，当找到最优奖赏函数或者迭代次数达到 300 时，实验结束。

将最终实验任务的 3 条专家轨迹作为专家样本，在 16 * 16 的格子世界上对 ReEnt-MIRL 算法和 Re-IRL 算法进行恢复奖赏函数功能的测试，分别从恢复奖赏函数的速度即参数 $\theta$ 的收敛速度和最终求解得到的奖赏函数的精度两个方面对两种算法进行比较。

#### 4.1.2 实验分析

图 3 是 ReEnt-MIRL 算法和 Re-IRL 算法[6] 在相同的少量样本条件下，迭代次 100 次时各自恢复出的 16 * 16 格子世界每个状态的奖赏函数值，右侧刻度轴表示每个状态的奖赏函数值的取值范围，通过渐变颜色的刻度轴使实验结果更为直观。表 1 为此时两种算法的预测值与真实奖赏函数值之间的均方根误差（RMSE）。可以看出，在相同样本相同迭代次数的情况下，相比 Re-IRL 算法，ReEnt-MIRL 算法恢复出的奖赏函数更接近真实值，即后者奖赏函数参数的收敛速度比前者快。这主要是由相比 Re-IRL 算法随机设置的奖赏函数参数 $\theta$ 的初始值，本文提出的 ReEnt-MIRL 算法结合了模型无关的元学习方法，在通过专家样本对参数进行梯度下降之前学习关于参数 $\theta$ 的先验，为后续的梯度下降提供了良好的起点，提高了参数的收敛速度。

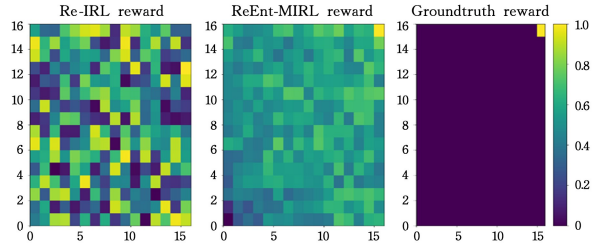


图 3 迭代 100 次时 ReEnt-MIRL 算法和 Re-IRL 算法的奖赏构建效果比较（电子版为彩色）

Fig. 3 Comparison of reward construction effects between ReEnt-MIRL algorithm and Re-IRL algorithm at 100 iterations

表 1 两种算法预测值与实际值的均方根误差（100 次迭代时）

Table 1 *RMSE* of predicted and actual values of two algorithms (100 iterations)

| Algorithm | *RMSE* |
|---|---|
| Re-IRL | 0.4282 |
| ReEnt-MIRL | 0.3223 |

此外，为了更好地说明所提算法在少量样本条件下的有效性，图 4 给出了 ReEnt-MIRL 算法和 Re-IRL 算法最终收敛后的奖赏函数求解结果对比，右侧刻度轴表示每个状态的奖赏函数值的取值范围，其中 ReEnt-MIRL 算法最终收敛时的情节数为 150，而 Re-IRL 算法则为 200。表 2 列出了此时两种算法的均方根误差。可以看出，ReEnt-IRL 算法不仅在收敛速度上优于 Re-IRL 算法，而且在少量专家样本的情况下对奖赏函数的恢复效果也比后者更好，这是因为缺少专家样本会导致基于概率模型的逆强化学习方法在对奖赏函数参数进行梯度下降的过程因梯度不精确而易陷入局部最优，而本文所提出的 ReEnt-MIRL 算法在面对仅有少量专家样本的新任务时，通过元学习的方法对奖赏函数参数进行预训练，从而设置了更为合适的初始值，相比 Re-IRL 算法能够有效避免最终结果陷入局部最优。综上所述，在少量样本条件的情况下，ReEnt-MIRL 算法比 Re-IRL 算法具有更快的学习速率，在最终奖赏函数的恢复效果上也好于后者。

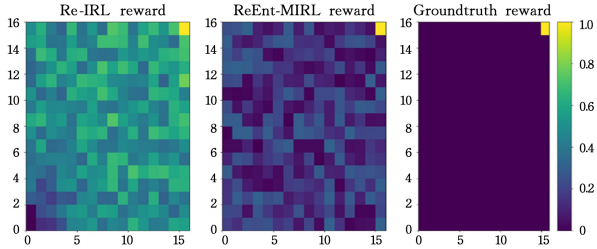


图 4 Gridworld 中两种算法最终收敛后奖赏构建效果对比（电子版为彩色）

Fig. 4 Comparison of reward construction effects of two algorithms after final convergence in Gridworld

表 2 Gridworld 中两种算法最终收敛后预测值和实际值的均方根误差

Table 2 *RMSE* of predicted and actual values of two algorithms after final convergence in Gridworld

| Algorithm | *RMSE* |
|---|---|
| Re-IRL | 0.2976 |
| ReEnt-MIRL | 0.1564 |

## 4.2 Object World 问题

### 4.2.1 实验描述

如图 5 所示，Object World 问题是一个格子的网格世界问题，Agent 在每一个状态下可以进行 5 种操作：上、下、左、右以及保持当前状态。在学习过程中，Agent 通过当前所执行的动作实现状态转移。此外，风的存在使其有 30%的可能性移动到随机方向。

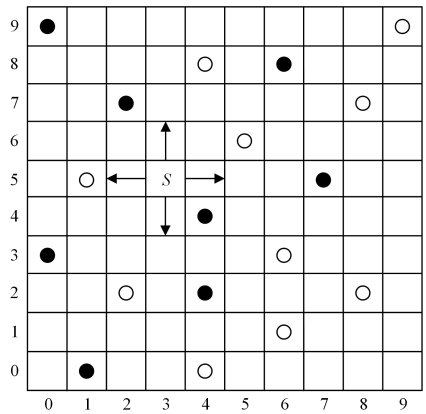


图 5 10 * 10 Object 世界

Fig. 5 10 * 10 Object World

网格中随机分布着一些“object”，在本文中将其设置为“白色圆”与“黑色圆”，如果 Agent 既在白色圆的 3 个单元格内，又在黑色圆的 2 个单元格内，则奖励为+1；如果只在白色圆的 3 个单元格内，而不在黑色圆的 2 个单元格内，则奖励为−1；其余状态的奖励为 0。

在本实验中将“object”数目设置为 15，随机放置在网格世界中，其中每种“object”位置分布构成一个任务。对每个任务，Agent 从随机状态开始，通过正向强化学习算法结合人为设定的奖赏函数求解出专家策略并采样得到 20 条专家轨迹作为样本，在这里将轨迹长度设为 3 倍网格世界的尺寸，从而构建出一个还原人为设定奖赏函数的逆强化学习任务。任选一种“object”位置分布的情况作为最终的实验任务，通过相同的方法采样得到 5 条专家轨迹作为还原奖赏函数的样本，构建出少量样本条件下还原专家奖赏函数的逆强化学习任务。

实验中设定网格世界尺寸为 16，折扣率为 0.99，两个梯度下降步长参数 $\alpha$ 和 $\beta$ 均为 0.01，最大迭代次数为 300，当找到最优奖赏函数或者迭代次数达到 300 时，实验结束。

### 4.2.2 实验分析

图 6 是 ReEnt-MIRL 算法和 Re-IRL 算法在最终收敛后的奖赏函数恢复效果，右侧刻度轴表示每个状态的奖赏函数值的取值范围，表 3 所列为此时两种算法的均方根误差。可以明显看出，相比 ReEnt-MIRL 算法，Re-IRL 算法仅仅对于奖赏值为-1 的状态处有恢复奖赏函数的效果，而对其他状态下的奖赏函数的恢复效果较差。这是由于在少量样本的条件下，Re-IRL 算法中关于奖赏函数参数的梯度不准确且随机设定初始值，使得算法在执行过程中极易陷入局部最优，从而导致最终恢复出的奖赏函数精度较低。本文所提出的算法通过模型无关元学习的方法对奖赏函数参数进行预训练，得到一个易于微调的参数初始值，仅需要少量专家样本就可以快速准确地恢复出奖赏函数的参数值，相比 Re-IRL 算法，极大地避免了陷入局部最优的情况。

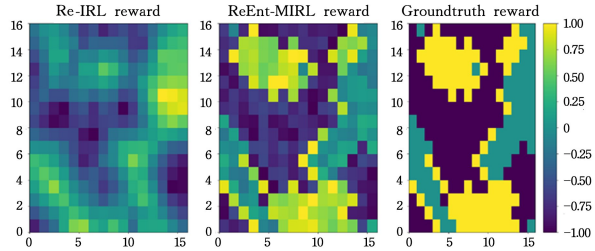


图 6 Object World 中两种算法最终收敛后奖赏构建效果的对比（电子版为彩色）

Fig. 6 Comparison of the reward construction effects of two algorithms after final convergence in Object World

表 3 Object World 中两种算法最终收敛后预测值和实际值的均方根误差

Table 3 *RMSE* of the predicted and actual values of two algorithms after final convergence in Object World

| Algorithm | *RMSE* |
|---|---|
| Re-IRL | 0.3534 |
| ReEnt-MIRL | 0.1787 |

图 7 是不同学习率（梯度下降超参数 $\alpha$ 和 $\beta$）的 ReEnt-MIRL 算法最终得到的奖赏函数效果对比图，右侧刻度轴表示每个状态的奖赏函数值的取值范围。当学习率分别为 0.05，0.001 和 0.01 时，奖赏函数参数最终收敛的迭代次数分别为 180，300 和 200。

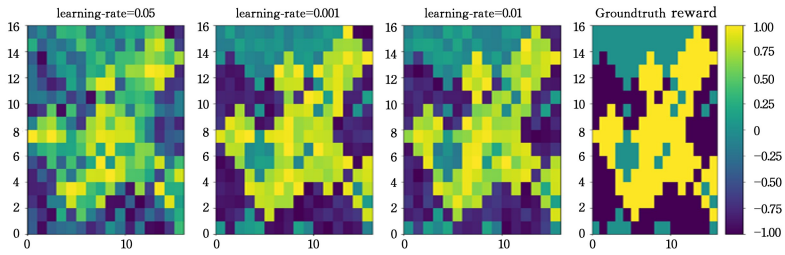


图 7 不同学习率的 ReEnt-MIRL 算法奖赏构建效果对比（电子版为彩色）

Fig. 7 Comparison of ReEnt-MIRL algorithm reward construction effects with different learning rates

表 4 所列为 3 种不同学习率的 ReEnt-MIRL 算法的奖赏函数预测值与真实值的均方根误差。可以看出，当学习率为 0.001时算法的奖赏函数构建效果与学习率为 0.01 时的构建效果基本接近，但算法的收敛速度比后者慢，原因是学习率较小时，参数在相同时间内梯度下降的幅度较小，导致收敛速度变慢。而当学习率为 0.05 时，虽然收敛速度比学习率为 0.01 时稍快，但是奖赏函数的构建效果精度较低，收敛效果不稳定，原因是学习率较大使得参数在梯度下降过程中陷入了局部最优。综上所述，为了更加快速和稳定地恢复奖赏函数，这里学习率 $\alpha$ 和 $\beta$ 的取值均为 0.01。

表 4 不同学习率的 ReEnt-MIRL 算法的预测值和真实值之间的均方根误差

Table 4 *RMSE* between predicted value and true value of ReEnt-MIRL algorithm with different learning rates

| Learning-rate | *RMSE* |
|---|---|
| 0.05 | 0.2316 |
| 0.001 | 0.1578 |
| 0.01 | 0.1605 |

**结束语** 本文针对传统逆强化学习算法在缺少足够专家演示样本以及状态转移概率信息的情况下，求解奖赏函数时速度慢、精度低甚至无法求解的问题，提出了一种基于相对熵的元逆强化学习方法。在学习过程中，结合与目标任务同分布的一组元训练集，构建目标任务学习先验，利用元学习方法，实现了利用目标任务少量样本快速求解目标任务奖赏函数。同时针对缺少状态转移函数的情况，本文采用相对熵概率模型的方法对奖赏函数进行建模，在先验的基础上仅需少量专家样本即可求解出奖赏函数。实验表明，基于相对熵的元逆强化学习算法可以在少量专家样本的情况下提高求解奖赏函数的精度并加快参数收敛速度。

由于本文中两个实验环境均为小规模离散状态空间，接下来的工作是将所提算法用于大规模连续状态空间，以提升算法性能，使得算法可以适用于实际问题，并可应用到智能建筑环境调控领域。同时，我们会尝试将不同的元学习的方法应用到逆强化学习问题中，并与本文方法进行比较。

## 参 考 文 献


[1] SUTTON R S, BARTO A G. Reinforcement learning: An introduction[M]. MIT Press, 2018.

[2] NG A Y, RUSSELL S J. Algorithms for inverse reinforcement learning[C]// Proceedings of the International Conference on Machine Learning. California, USA, 2000: 663-670.

[3] ABBEEL P, NG A Y. Apprenticeship learning via inverse reinforcement learning[C]// Proceedings of the International Conference on Machine Learning. Banff, Canada, 2004: 1.

[4] RATLIFF N D, SILVER D, BAGNELL J A. Learning to search: Functional gradient techniques for imitation learning[J]. Autonomous Robots, 2009, 27(1): 25-53.

[5] ZIEBART B D, MAAS A L, BAGNELL J A, et al. Maximum Entropy Inverse Reinforcement Learning[C]// Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008). Chicago, Illinois, USA, 2008: 13-17.

[6] BOULARIAS A, KOBER J. Relative Entropy Inverse Reinforcement Learning[C]// Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 2011. Fort Lauderdale, FL, USA, 2011.

[7] WANG Y X, HEBERT M. Learning to learn: Model regression networks for easy small sample learning[C]// European Conference on Computer Vision. Springer, Cham, 2016: 616-634.

[8] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]// Proceedings of the 34th International Conference on Machine Learning. 2017: 1126-1135.

[9] SNELL J, SWERSKY K, ZEMEL R. Prototypical networks for few-shot learning[C]// Advances in Neural Information Processing Systems. 2017: 4077-4087.

[10] MISHRA N, ROHANINEJAD M, CHEN X, et al. Meta-learning with temporal convolutions[J]. arXiv: 1707.03141.

[11] ANDRYCHOWICZ M, DENIL M, COLMENAREJO S G, et al. Learning to learn by gradient descent[C]// 30th Conference on Neural Information Processing Systems (NIPS 2016). Barcelona, Spain. 2016.

[12] CHEN X L, CAO L, HE M, et al. A Summary of Research on Deep Reverse Reinforcement Learning[J]. Computer Engineering and Applications, 2018, 54(5): 24-35.

[13] XIA C, KAMEL A E. Neural inverse reinforcement learning in autonomous navigation[J]. Robotics & Autonomous Systems, 2016, 84: 1-14.

[14] YI Z, ZHANG H, TAN P, et al. Dualgan: Unsupervised dual learning for image-to-image translation[C]// Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy, 2017: 2849-2857.

[15] BYRAVAN A, MONFORT M, ZIEBART B, et al. Graph-based inverse optimal control for robot manipulation[C]// Proceedings of the Association for the Advance of Artificial Intelligence. Austin, USA, 2015: 1874-1890.


**WU Shao-bo**, born in 1996, postgraduate. His main research interests include reinforcement learning, inverse reinforcement learning and building energy conversation.

**FU Qi-ming**, born in 1985, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include reinforcement learning, deep learning and building energy conservation.