LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

Domain Adaptation with Rewards from Classifiers

Presented by Yafei Hu

2021.01.24

目录

- 决策问题的概率推理建模
- DARC算法

Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review

https://arxiv.org/pdf/1805.00909.pdf

决策问题的概率推理建模

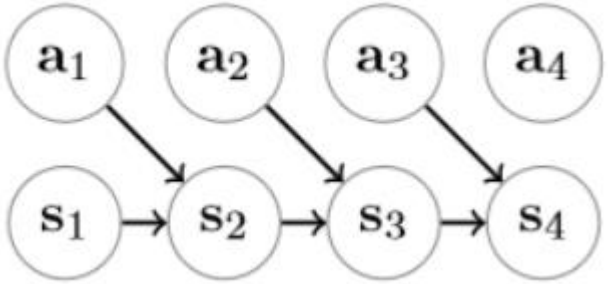
对于某个策略$\pi_\theta(\mathrm{a}|s)$, 定义其相应的策略轨迹分布:

$$p(\tau|\theta) = p(s_1, a_1, \ldots, s_T, a_T|\theta) = p(s_1)\prod_{t=1}^{T} p(a_t|s_t, \theta)\, p(s_{t+1}|s_t, a_t)$$

如何定义一个概率图模型，使得其计算出的轨迹分布等价于使用最优策略采样出的轨迹分布?

决策问题的概率推理建模

现有的状态，动作以及下一状态之间的图模型：



没有奖励信息，无法表示策略的最优性。

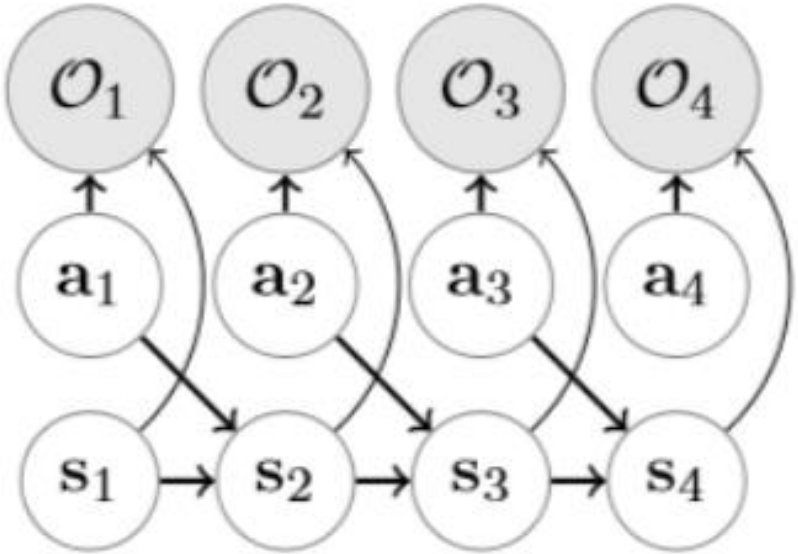
决策问题的概率推理建模

引入一个随机变量$\mathcal{O}_t$,且$\mathcal{O}_t$ =1表示在t时间步的决策是最优的，$\mathcal{O}_t$ = 0则表示不是最优的。另将$\mathcal{O}_t$ 的分布定义为：

$$p(\mathcal{O}_t = 1|s_t, a_t) = exp\big(r(s_t, a_t)\big)$$

决策问题的概率推理建模

引入o_t后，得到有向图模型：



决策问题的概率推理建模

由此，按该有向图可计算最优轨迹的分布$p(\tau, o_{1:T})$：

$$p(\tau, o_{1:T}) = p(s_1)\prod_{t=1}^{T} p(\mathcal{O}_t = 1|s_t, a_t)p(s_{t+1}|s_t, a_t)$$

$$= p(s_1)\prod_{t=1}^{T} exp\big(r(s_t, a_t)\big)p(s_{t+1}|s_t, a_t)$$

$$= \left[p(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\right] exp\left(\sum_{t=1}^{T} r(s_t, a_t)\right)$$

决策问题的概率推理建模

假设要求的最优策略为$p\left(a_t|s_t,o_{1:T}\right)$，那么由其导出的轨迹分布为：

$$\hat{p}(\tau)=p(s_1|o_{1:T})\prod_{t=1}^{T}p(s_{t+1}|s_t,a_t,o_{1:T})\ p(a_t|s_t,o_{1:T})$$

$$=p(s_1)\prod_{t=1}^{T}p(s_{t+1}|s_t,a_t\ )\ \pi^*(a_t|s_t\ )$$

决策问题的概率推理建模

故求解最优策略可以转化为分布匹配的问题，也就是使策略导出的轨迹分布匹配最优轨迹分布$p(\tau, o_{1:T})$，若使用KL散度衡量分布间的距离，则要优化的目标是：

$$-D_{KL}(\hat{p}(\tau)|p(\tau, o_{1:T})) = E_{\tau\sim\hat{p}(\tau)}[\log p(\tau, o_{1:T}) - \log \hat{p}(\tau)]$$

$$= E_{\tau\sim\hat{p}(\tau)}\left[\log p(s_1) + \sum_{t=1}^{T}\begin{pmatrix}\log p(s_{t+1}|s_t, a_t) & + \\ r(s_t, a_t) & \end{pmatrix} - \log p(s_1) - \sum_{t=1}^{T}\begin{pmatrix}\log p(s_{t+1}|s_t, a_t) & + \\ \log \pi^*(a_t|s_t) & \end{pmatrix}\right]$$

$$= E_{\tau\sim\hat{p}(\tau)}\left[\sum_{t=1}^{T}(r(s_t, a_t) - \log \pi^*(a_t|s_t))\right]$$

$$= \sum_{t=1}^{T} E_{(s_t,a_t)\sim\hat{p}(s_t,a_t)}[r(s_t, a_t)] + E_{s_t\sim\hat{p}(s_t)}[\mathcal{H}(\pi(a_t|s_t))]$$

可由上式的最后结果看出，如此形式化后求解的目标与普通的强化学习有所不同，除了最大化期望回报外，还使策略的条件熵尽量大。

Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers

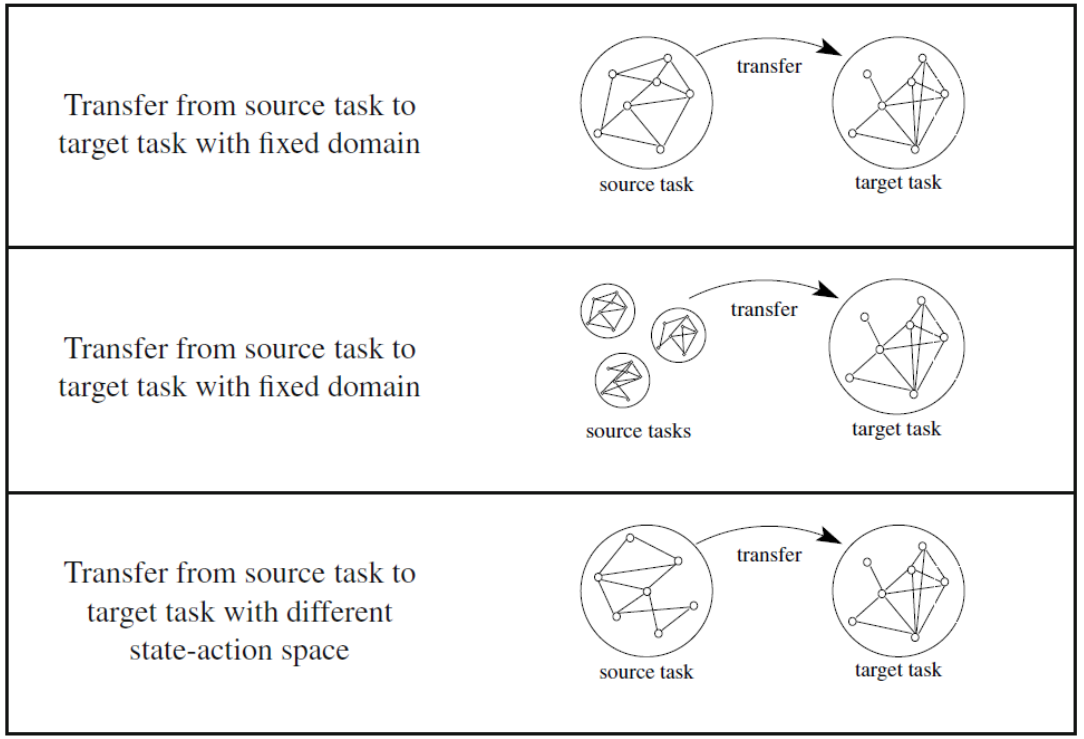
https://arxiv.org/abs/2006.13916

背景

MDP $\{S, A, T, R, \rho, \gamma\}$

可迁移的知识包括样本，学习到的表示，模型参数等。

迁移问题可以划分为右图所示三类：

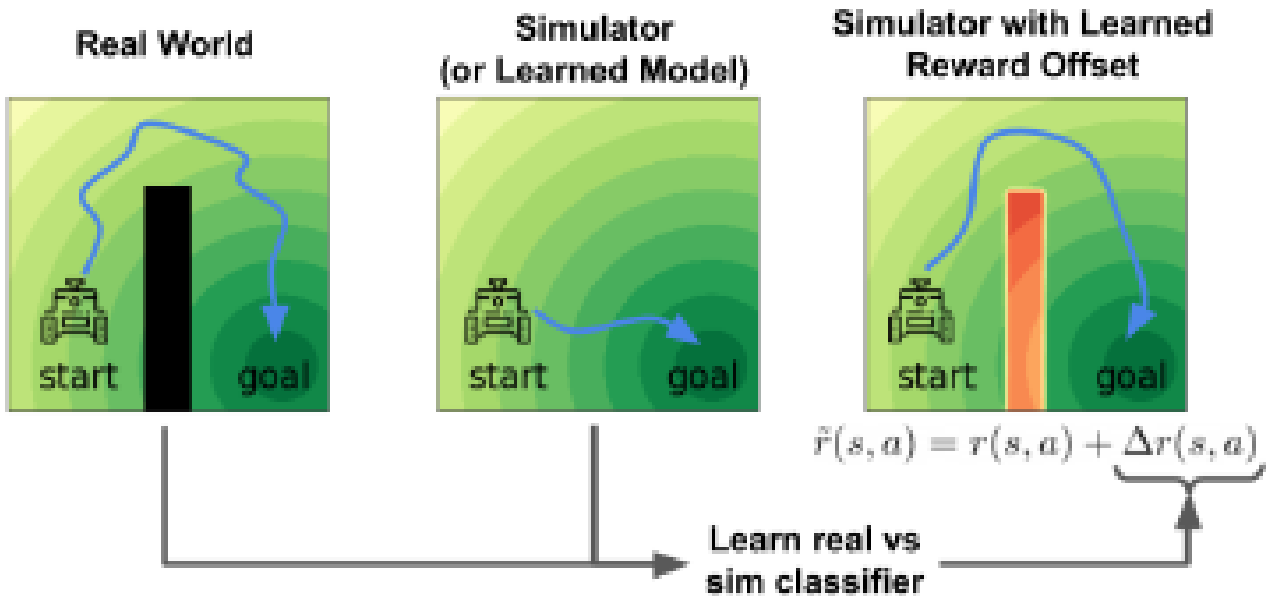


问题设置

- 源MDP与目标MDP仅仅状态转移方程不同。
- 目标是使用与源MDP交互产生的transition以及少量目标MDP产生的transition来求解目标MDP的最优策略。
- 一个假设：$p_{target}(s_{t+1}|s_t, a_t) > 0 \Rightarrow p_{source}(s_{t+1}|s_t, a_t) > 0$

解决思路

修改奖励函数来反映dynamic的不同，具体来说，在智能体到达与目标MDP不同的状态和相应动作的时候对它施加惩罚。



左一图为目标MDP，中间图为源MDP，右一为奖励函数修改后的MDP

- 怎么确定$\triangle r$?
- 有什么理论保证?

DARC算法

回顾上一篇论文中的概率推理建模，关键是表示出最优轨迹的分布$p(\tau, o_{1:T})$和最优策略分布$\hat{p}(\tau)$，在本算法的环境设置中，要求的是在目标MDP中效果很好的策略，故$p(\tau, o_{1:T})$是在目标MDP上计算的；而训练是在源MDP上的，即$\hat{p}(\tau)$是基于源MDP计算的。故

$$p(\tau, o_{1:T}) = \left[p(s_1) \prod_t p_{target}(s_{t+1}|s_t, a_t) \right] exp\left(\sum_t r(s_t, a_t) \right)$$

$$\hat{p}(\tau) == p(s_1) \prod_t p_{source}(s_{t+1}|s_t, a_t) \ \pi^*(a_t|s_t)$$

DARC算法

则优化目标:

$$-D_{KL}(\hat{p}(\tau)|p(\tau,o_{1:T})) = E_{\tau\sim\hat{p}(\tau)}[\log p(\tau,o_{1:T}) - \log\hat{p}(\tau)]$$

$$= E_{\tau\sim\hat{p}(\tau)}\left[\log p(s_1) + \sum_t \begin{pmatrix}\log p_{target}(s_{t+1}|s_t,a_t) & + \\ r(s_t,a_t) & \end{pmatrix} - \log p(s_1) - \sum_t \begin{pmatrix}\log p_{source}(s_{t+1}|s_t,a_t) & + \\ \log\pi^*(a_t|s_t) & \end{pmatrix}\right]$$

$$= E_{\tau\sim\hat{p}(\tau)}\left[\sum_t \left(r(s_t,a_t) - \log\pi^*(a_t|s_t) + \log p_{target}(s_{t+1}|s_t,a_t) - \log p_{source}(s_{t+1}|s_t,a_t)\right)\right]$$

$$= \sum_{t=1}^{T} E_{(s_t,a_t)\sim\hat{p}(s_t,a_t)}[r'(s_t,a_t,s_{t+1})] + E_{s_t\sim\hat{p}(s_t)}[\mathcal{H}(\pi(a_t|s_t))]$$

$\triangle r\,(s_t,a_t,s_{t+1})$

$$r'(s_t,a_t,s_{t+1}) = r(s_t,a_t) + \triangle r\,(s_t,a_t,s_{t+1})$$

DARC算法

由此确定$\triangle r\left(s_t, a_t, s_{t+1}\right) = \log p_{target}(s_{t+1}|s_t, a_t) - \log p_{source}(s_{t+1}|s_t, a_t)$，这是理想形式，在实际情况中转移方程一般是不容易知道的。DARC算法使用两个二分类器估计$\triangle r$：

$$\triangle r\left(s_t, a_t, s_{t+1}\right)$$
$$= \left(\log p(target|s_t, a_t, s_{t+1}) - \log p(target|s_t, a_t)\right)$$
$$- \left(\log p(source|s_t, a_t, s_{t+1}) - \log p(source|s_t, a_t)\right)$$

DARC算法

Algorithm 1 Domain Adaptation with Rewards from Classifiers [DARC]

for $t = 1, \cdots,$ num iterations **do**
  $\mathcal{D}_{\text{source}} \leftarrow \mathcal{D}_{\text{source}} \cup \textsc{Rollout}(\pi, \mathcal{M}_{\text{source}})$
  if $t \mod r = 0$ **then**
    $\mathcal{D}_{\text{target}} \leftarrow \mathcal{D}_{\text{target}} \cup \textsc{Rollout}(\pi, \mathcal{M}_{\text{target}})$
  $\theta \leftarrow \theta - \eta \nabla_\theta \ell(\theta)$
  $\tilde{r}(s_t, a_t, s_{t+1}) \leftarrow r(s_t, a_t) + \Delta r(s_t, a_t, s_{t+1})$
  $\pi \leftarrow \textsc{MaxEnt RL}(\pi, \mathcal{D}_{\text{source}}, \tilde{r})$
return π

实验

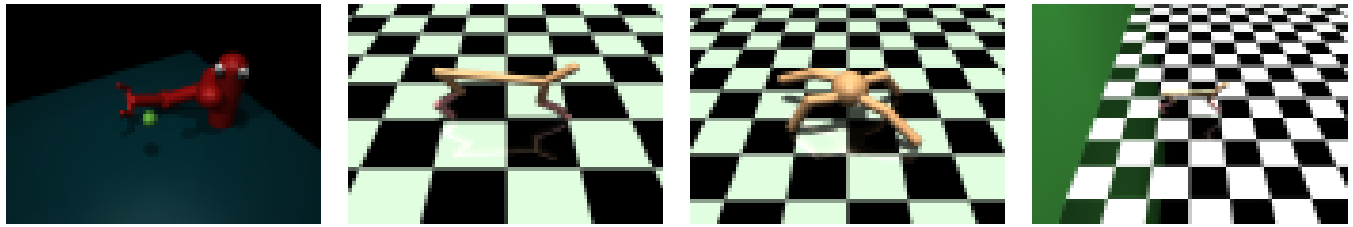
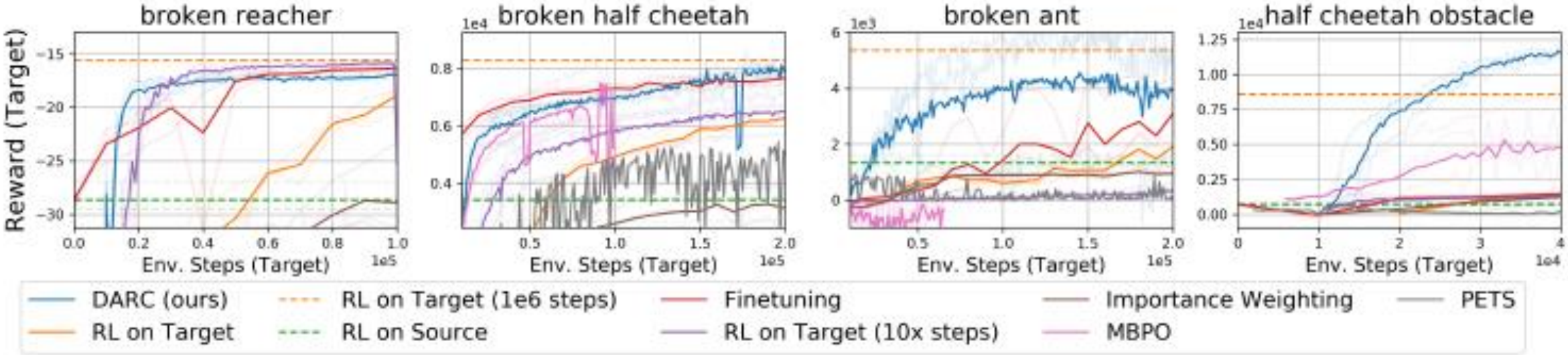


Figure 3. **Environments:** (L to R) broken reacher, broken half cheetah, broken ant, and half cheetah obstacle.



实验


Training without Δr
Training with Δr
Reward Correction (Δr)
0
−2
−4
−6
−8
−10
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
Training Steps
Agent finds a strategy where the source and target domains are indistinguishable.

总结

回顾一下，第一篇文章将强化学习的决策问题建模成一个概率推理模型，并揭示了其优化目标与最大熵强化学习的优化目标相同；第两篇文章基于第一篇文章，给了我们一个新的视角去处理仅有dynamic不同的MDP之间的样本迁移问题，即改变奖励来将这种MDP间Dynamic的不同反应给智能体，使其更倾向使用类似目标域的transition训练策略。

Thanks!