AdaRL

: What, Where, and How to Adapt in Transfer Reinforcement Learning

Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, Kun Zhang
ICLR 2022

IIE8557-01 동적계획법과 강화학습

경영과학연구실 김지원

Adaptations in Reinforcement learning

- Issue: Most of early successes of RL focus on a fixed task in a fixed environment

-> In real applications we often have changing environments, and the optimal policy learned in a specific domain may not be generalized to other domains while <u>humans are usually good at transferring acquired knowledge</u>

The study aims to make quick adaptations when faced with new environments

Dealing with changes across domains with a few samples from the target domain

Two research lines in transfer RL

1) finding policies that are robust to environment variations
- maximizing a risk-sensitive objective over a distribution of environments
- extracting a set of invariant states

2) adapting policies from the source domain to the target domain as efficiently as possible
- use importance reweighting on samples
- start from the optimal source policy to initialize a learner in the target domain
- a model is pretrained on a source domain and the output layers are finetuned via backpropagation in the target domain

➔ In a new environment not all parameters need to be updated, so we can force the model to only adapt a set of context parameters

Limitation: Previous methods mostly focus on MDPs and model all changes as a black-box, which may be less efficient for adaptation

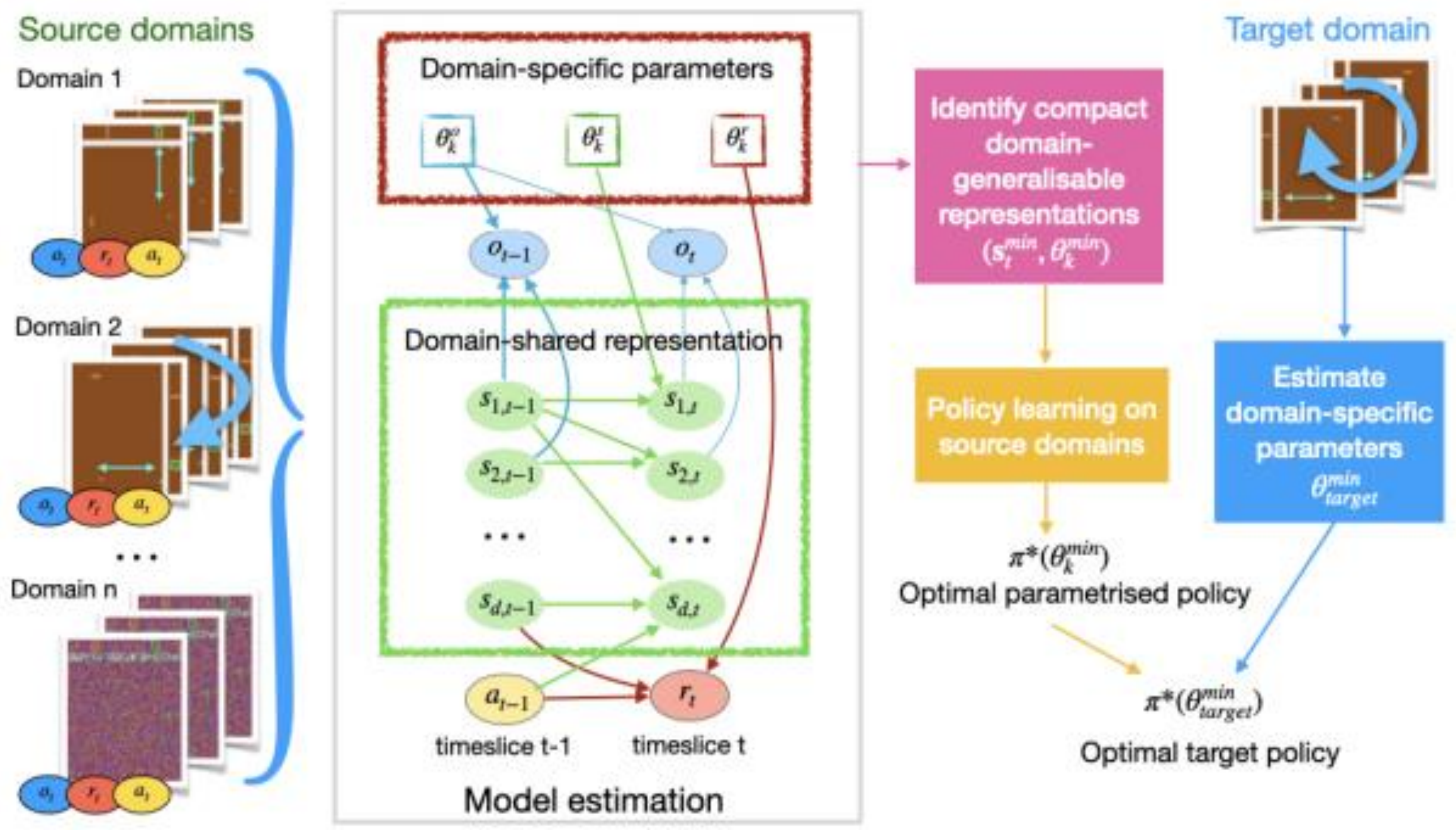
AdaRL

Motivation: The distribution shifts are usually localized – they are often due to the changes of only a few variables in the generative processes, so we can just adapt the distribution of a small portion of variables

AdaLR

- achieves low-cost, reliable, and interpretable transfer for partially observable Markov decision processes
- learns a parsimonious graphical representation that is able to characterize structural relationships among different dimensions of states, change factors, the perception, the reward variable, and the action variable

Framework


Source domains
Domain 1
Domain 2
Domain n
Domain-specific parameters
Domain-shared representation
timeslice t-1
timeslice t
Model estimation
Identify compact domain-generalisable representations $(s_t^{min}, \theta_k^{min})$
Policy learning on source domains
$\pi^*(\theta_k^{min})$
Optimal parametrised policy
Target domain
Estimate domain-specific parameters θ_{target}^{min}
$\pi^*(\theta_{target}^{min})$
Optimal target policy

A Compact Representation of Environmental Shifts

S: underlying latent state
O: perceived signals at time t (e.g., images)
A: executed action
R: reward
C :binary vectors or scalars that represent structural relationships from one variable to the other
θ : low-dimensional change factors that have a constant value in each domain

if states s are directly observed, in which case the observation function of o is not needed

$$\begin{cases} s_{i,t} &= f_i(\mathbf{c}_i^{\mathbf{s}\to\mathbf{s}} \odot \mathbf{s}_{t-1}, c_i^{a\to\mathbf{s}} \cdot a_{t-1}, \mathbf{c}_i^{\theta_k\to\mathbf{s}} \odot \boldsymbol{\theta}_k^{\mathbf{s}}, \epsilon_{i,t}^{s}), \text{ for } i = 1, \cdots, d, \\ o_t &= g(\mathbf{c}^{\mathbf{s}\to o} \odot \mathbf{s}_t, c^{\theta_k\to o} \cdot \theta_k^{o}, \epsilon_t^{o}), \\ r_t &= h(\mathbf{c}^{\mathbf{s}\to r} \odot \mathbf{s}_{t-1}, c^{a\to r} \cdot a_{t-1}, c^{\theta_k\to r} \cdot \theta_k^{r}, \epsilon_t^{r}), \end{cases}$$

Structural relationships and graphs

- The paper introduced graph structure over variables

- Perceived signals o are generated from the underlying states s
- The actions at directly influence the latent states st+1

- Often the action variable at−1 does not influence every dimension of st
- The reward rt may not be influenced by every dimension of st−1

Structural relationships and graphs

Environment model G is encoded in the binary masks c

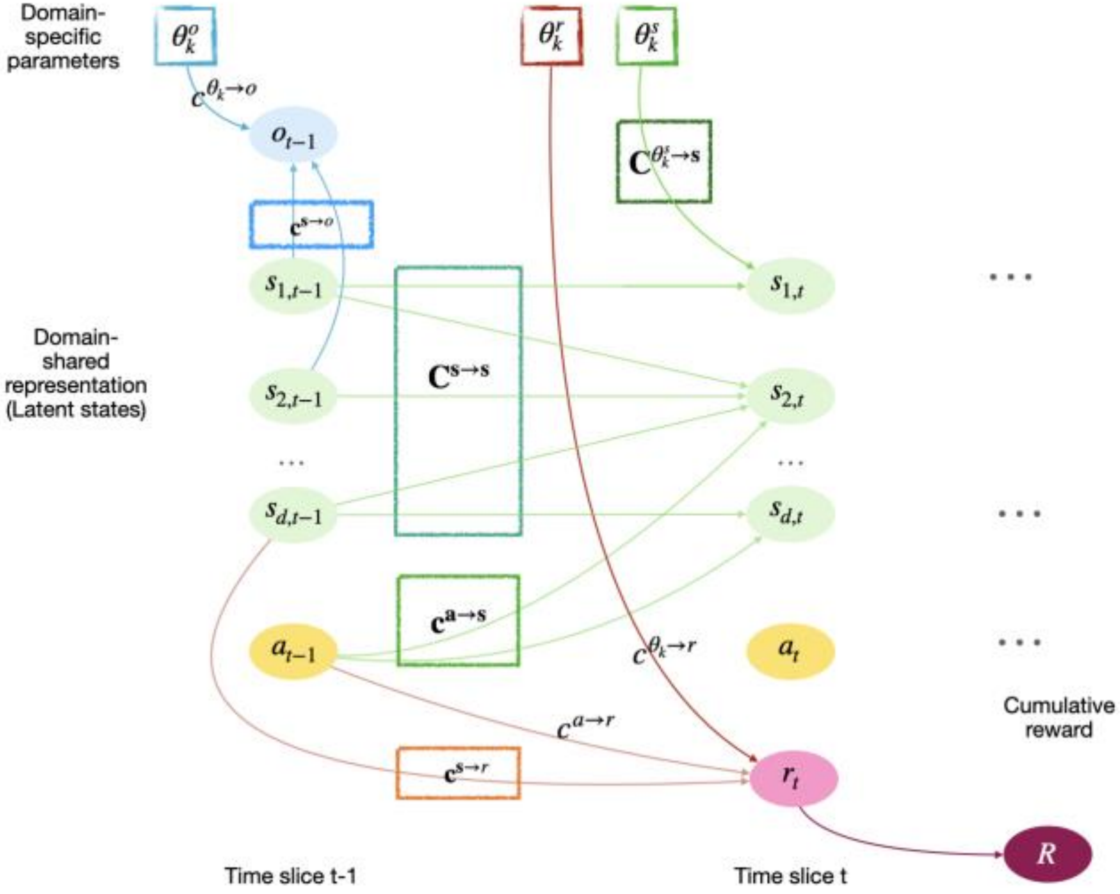
Compact domain-shared representations $\mathbf{s}_t^{min}$

- The latent state components that have an edge to the reward in the next time-step $\mathbf{c}_i^{\mathbf{s}\to r}=1$
or have an edge to another state component in the next time-step $\mathbf{c}_{j,i}^{\mathbf{s}\to\mathbf{s}}=1$

Compact domain-specific representations $\boldsymbol{\theta}_k^{min}$

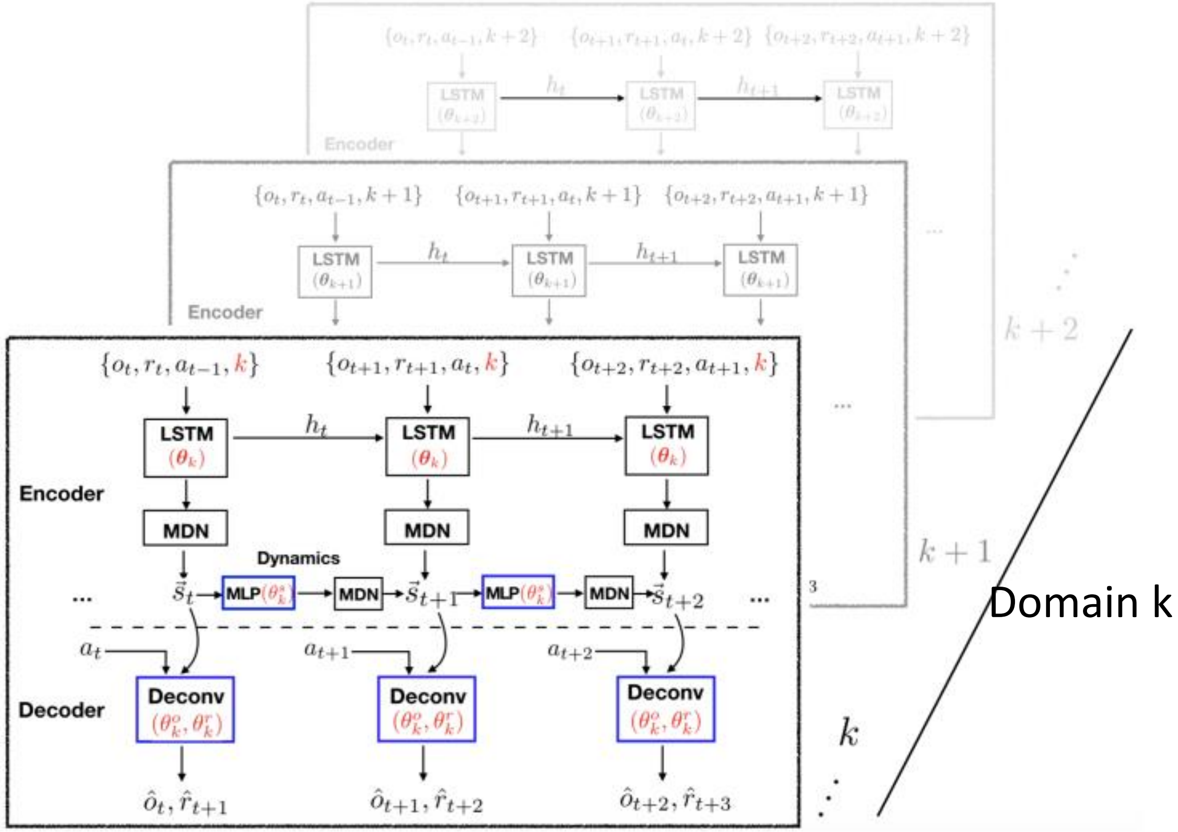
- The latent change factors that have an edge to the reward in the next time-step $c^{\theta_k\to r}=1$
or have an edge to a state component $\mathbf{c}_{j,i}^{\theta_k\to s}=1$

Structural relationships and graphs



MiSS-VAE

MiSS-VAE estimates models from different domains simultaneously, by exploiting commonalities across domains while at the same time preserving specific information for each domain



1) Sequential VAE: handles the sequential data, with the underlying latent states satisfying an MDP
2) Multi-model: handles models from different domains at the same time learning the domain-specific factors θ
3) Structure: exploits the structural information that is explicitly encoded with the binary masks c

Policy Transfer

Instead of learning the optimal policy in each domain separately, policies in different domains are optimized at the same time exploiting both commonalities and differences across domain

$$a_t = \pi^*(\mathbf{s}_t^{min}, \boldsymbol{\theta}_k^{min})$$

- Obtaining optimal policy in the target domain by learning $\pi *$ in the source domains, and estimating the value of the change factor θ and inferring latent states s from the target domain

Method

Algorithm 1 (AdaRL with Domains Shifts)

Initialize action-value function Q, target action-value function Q', and replay buffer $\mathcal{B}$.
Record multiple rollouts for each source domain $k (k = 1, \cdots, n)$ and estimate the model in Eq.1.
Identify the dimension indices of $\mathbf{s}_t^{min}$ and the values of $\boldsymbol{\theta}_k^{min}$ according to the learned model.
for episode = 1, . . . , M **do**
  for source domain k = 1, . . . , n **do**
    Receive initial observations $o_{1,k}$ and $r_{1,k}$ for the k-th domain.
    Infer the posterior $q(\mathbf{s}_{1,k}^{min} | o_{1,k}, r_{1,k}, \boldsymbol{\theta}_k^{min})$ and sample initial inferred state $\mathbf{s}_{1,k}^{min}$.
  end for
  for timestep t = 1, . . . , T **do**
    for source domain k = 1, . . . , n **do**
      Select $a_{t,k}$ randomly with probability ϵ; otherwise $a_{t,k} = \arg\max_a Q(\mathbf{s}_{t,k}^{min}, a, \boldsymbol{\theta}_k^{min})$.
      Execute action $a_{t,k}$, and receive reward $r_{t+1,k}$ and observation $o_{t+1,k}$ in the kth domain.
      Infer the posterior $q(\mathbf{s}_{t+1,k}^{min} | o_{\leq t+1,k}, r_{\leq t+1,k}, a_{\leq t,k}, \boldsymbol{\theta}_k^{min})$ and sample $\mathbf{s}_{t+1,k}^{min}$.
      Store transition $(\mathbf{s}_{t,k}^{min}, a_{t,k}, r_{t+1,k}, \mathbf{s}_{t+1,k}^{min}, \boldsymbol{\theta}_k^{min})$ in reply buffer $\mathcal{B}$.
    end for
    Randomly sample a minibatch of N transitions $(\mathbf{s}_{i,j}^{min}, a_{i,j}, r_{i+1,j}, \mathbf{s}_{i+1,j}^{min}, \boldsymbol{\theta}_j^{min})$ from $\mathcal{B}$.
    Set $y_{i,j} = r_{i+1,j} + \lambda \max_{a'} Q'(s_{i+1,j}^{min}, a', \boldsymbol{\theta}_j^{min})$.
    Update action-value function Q by minimizing the loss:

$$L = \frac{1}{n * N} \sum_{i,j} (y_{i,j} - Q(s_{i,j}^{min}, a_{i,j}, \boldsymbol{\theta}_j^{min}))^2.$$

  end for
  Update the target network Q': $Q' = Q$.
end for
Record a few rollouts from the target domain.
Estimate the values of $\boldsymbol{\theta}_{\text{target}}^{min}$ for the target domain, with all other parameters fixed.

data collection
from n source domains
and model estimation

learning the optimal policy π ∗
with deep Q-learning

Modified Cartpole

two change factors for the state dynamics θ_s^k : varying gravity and varying mass of the cart
change factor for the observation θ_s^o : Gaussian noise on the image

	Oracle Upper bound	Non-t lower bound	CAVIA (Zintgraf et al., 2019)	PEARL (Rakelly et al., 2019)	AdaRL* Ours w/o masks	AdaRL Ours
G_in	2486.1 (±369.7)	1098.5 • (±472.1)	1603.0 (±877.4)	1647.4 (±617.2)	1940.5 (±841.7)	2217.6 (±981.5)
G_out	693.9 (±100.6)	204.6 • (±39.8)	392.0 • (±125.8)	434.5 • (±102.4)	439.5 • (±157.8)	**508.3** (±138.2)
M_in	2678.2 (±630.5)	748.5 • (±342.8)	2139.7 (±859.6)	1784.0 (±845.3)	1946.2 • (±496.5)	2260.2 (±682.8)
M_out	1405.6 (±368.0)	371.0 • (±92.5)	972.6 • (±401.4)	793.9 • (±394.2)	874.5 • (±290.8)	**1001.7** (±273.3)
G_in & M_in	1984.2 (±871.3)	365.0 • (±144.5)	1012.5 • (±664.9)	1260.8 • (±792.0)	1157.4 • (±578.5)	**1428.4** (±495.6)
G_out & M_out	939.4 (±270.5)	336.9 • (±139.6)	648.2 • (±481.5)	544.32 • (±175.2)	596.0 • (±184.3)	**689.4** (±272.5)

Modified Pong

change factors for the state dynamics θ_s^k : rotate the images ω degrees clockwise
change factors for the observation θ_s^o : different image sizes, different image colors, and different noise levels

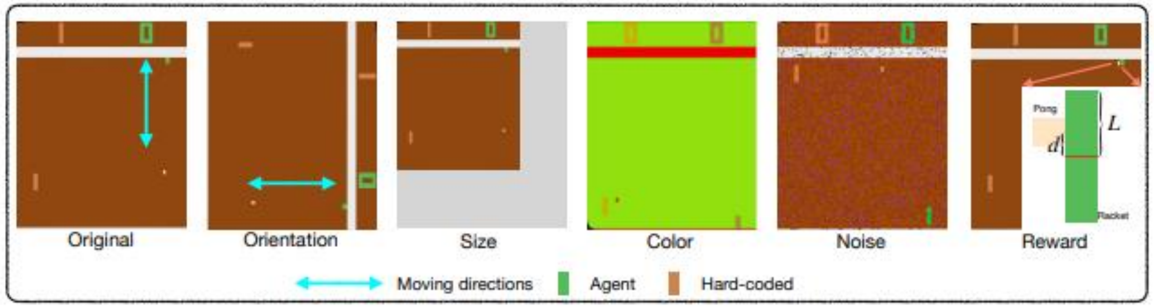


Figure 2: Illustrations of the change factors on modified Pong game.

	Oracle Upper bound	Non-t lower bound	PNN (Rusu et al., 2016)	PSM (Agarwal et al., 2021)	MTQ (Fakoor et al., 2020)	AdaRL* Ours w/o masks	AdaRL Ours
O_in	18.65 (±2.43)	6.18 • (±2.43)	9.70 • (±2.09)	11.61 • (±3.85)	15.79 • (±3.26)	14.27 • (±1.93)	**18.97** (±2.00)
O_out	19.86 (±1.09)	6.40 • (±3.17)	9.54 • (±2.78)	10.82 • (±3.29)	10.82 • (±4.13)	12.67 • (±2.49)	**15.75** (±3.80)
C_in	19.35 (±0.45)	8.53 • (±2.08)	14.44 • (±2.37)	19.02 (±1.17)	16.97 • (±2.02)	18.52 • (±1.41)	19.14 (±1.05)
C_out	19.78 (±0.25)	8.26 • (±3.45)	14.84 • (±1.98)	17.66 • (±2.46)	15.45 • (±3.30)	17.92 (±1.83)	19.03 (±0.97)
S_in	18.32 (±1.18)	6.91 • (±2.02)	11.80 • (±3.25)	12.65 • (±3.72)	13.68 • (±3.49)	14.23 • (±3.19)	**16.65** (±1.72)
S_out	19.01 (±1.04)	6.60 • (±3.11)	9.07 • (±4.58)	8.45 • (±4.51)	11.45 • (±2.46)	12.80 • (±2.62)	**17.82** (±2.35)
N_in	18.48 (±1.25)	5.51 • (±3.88)	12.73 • (±3.67)	11.30 • (±2.58)	12.67 • (±3.84)	13.78 • (±2.15)	**16.84** (±3.13)
N_out	18.26 (±1.11)	6.02 • (±3.19)	13.24 • (±2.55)	11.26 • (±3.15)	15.77 • (±2.12)	14.65 • (±3.01)	**18.30** (±2.24)

Conclusions

- AdaRL learns a latent representation with domain-shared and domain-specific components across source domains and uses it to learn an optimal policy parameterized by the domain-specific parameters

- As opposed to previous work, AdaRL can model changes in the state dynamics, observation function and reward function in an unified manner, and exploit the factorization to improve the data efficiency and adapt faster with fewer samples