1 Problem formulations and assumptions

1. Source domain $D_S = (S_S, A_S, T_S, R_S)$, target domain $D_T = (S_T, A_T, T_T, R_T)$, each is a MDP.
2. S, A, T can be very different.
3. The reward functions have similarity, in that the state distribution of an optimal policy in D_S will resemble the state distribution of an optimal policy in D_T when projected into some common feature space.

1.1 Formalizing common feature space assumption

1. Let $\pi_S(s_S)$ denotes the state distribution of the optimal policy in D_S.
2. Let $\pi_T(s_S)$ denotes the state distribution of the optimal policy in D_T.
3. It is possible to learn 2 functions f and g such that $p(f(s_S)) = p(g(s_T))$ for $s_S \sim \pi_S$ and $s_T \sim \pi_T$.
4. Meaning that the images of π_S under f and of π_T under g corresponds to the same distribution.
5. Degenerate case: $f(s_S) = g(s_T) = 0$.
6. To over the degenerate case, use reconstruction to maximize the amount of information contained in the shared feature space.

1.2 Learning with multiple skills

1. They need samples from both domains to learn a common feature space.
2. They assume that the agents in D_S and D_T have prior knowledge about each other, in the form of other skills that each have learnt.
3. To formalize the setting where the agents can perform multiple tasks, they divide the state-space in each of the domains into an agent-specific state s_r and a task-specific state s_{env}.
4. For simplicity, they assume that there are just two type of skills:
5. The proxy skill that has been learnt by both agents.
6. The test skill that has been learnt by some agent in D_S and is currently being transferred to the target agent in domain D_T.
7. D_{S_p}, D_{T_p} denotes the proxy task domains in the source and target agent.
8. They assume that D_S, D_{S_p} differs only in reward function and task-specific state. The agent-specific state space S_r and action spaces are the same.
9. The main idea is that the agents will have already learnt the proxy tasks. They can use how they perform in this task to determine the common feature space.

1.3 Estimating correspondences from proxy skills

1. They want to use the proxy skill to learn how to pair agent-specific states across both domains.
2. They use the fact that the skills are episodic to learn this pairing, denoted by P.
3. Time-based alignment: assume the 2 agents perform the tasks at the same rate, pair states that are visited at the same time in the 2 proxy domains.
4. This is sensitive to how the fast the agents perform the task relative to each other.
5. Alternating optimization: using dynamic time wrapping.
6. Dynamic time wrapping is a method for learning correspondence between sequences which vary in speed.

7. Alternate between learning a common feature space using the currently estimated correspondence and re-estimating correspondences using the currently learned feature space.

8. Dynamic time wrapping needs a metric space to compare elements in the sequences to compute an optimal alignment between the sequences.

9. Initialize with the weak time-based alignment.

2 Learning common feature spaces for skill transfer

1. They want to find functions f and g such that for states s_{S_p}, s_{T_p} along the optimal policies $\pi_S p^*$ and $\pi_T p^*$, f and g approx satisfies $p\left(f\left(s_{Sp,r}\right)\right)=p\left(g\left(s_{Tp,r}\right)\right)$.

2. If they can find the common feature space by learning f and g, they can optimize π_T by directly mimicking the distribution over $f\left(s_{Sp,r}\right)$, where $s_{Sp,r} \sim \pi_S$.

2.1 Learning the embedding functions from a proxy task

1. The loss function is

$$L_{\text{sim}}\left(s_{Sp}, s_{Tp}; \theta_f, \theta_g\right)=\left\|f\left(s_{Sp,r}; \boldsymbol{\theta}_f\right)-g\left(s_{Tp,r}; \boldsymbol{\theta}_g\right)\right\|_2$$

where the θ are function parameters, $\left(s_{Sp,r}, s_{Tp,r}\right) \in P$.

2. Degenerate case, f and g both output 0.

3. They intuit that the embedding functions are good when they are invertible, so that as information about what's being embedded is being preserved as possible.

4. They thus train decoder networks with reconstruction loss.

5. The reconstruction losses are

$$L_{\mathrm{AE}_{\mathrm{S}}}\left(s_{Sp,r}; \boldsymbol{\theta}_f, \theta_{\mathrm{Dec}_S}\right)=\left\|s_{Sp,r}-\mathrm{Dec}_S\left(f\left(s_{Sp,r}; \boldsymbol{\theta}_f\right); \theta_{\mathrm{Dec}_S}\right)\right\|_2$$

$$L_{\mathrm{AE}_{\mathrm{T}}}\left(s_{Tp,r}; \theta_g, \theta_{\mathrm{Dec}_T}\right)=\left\|s_{Tp,r}-\mathrm{Dec}_T\left(g\left(s_{Tp,r}; \theta_g\right); \theta_{\mathrm{Dec}_T}\right)\right\|_2$$

6. All 3 loss functions are jointly optimized.

2.2 Using the common embedding for knowledge transfer

1. They argues that because f and g are not invertible, directly mapping from a state in the source domain to a state in the target domain is not feasible. (but can't we use the decoder as the inverse mapping?)

2. They instead aim to match the distributions of optimal trajectories across the domains.

3. That is, they would like $p\left(f\left(s_{Sp,r}\right)\right)$ and $p\left(g\left(s_{Tp,r}\right)\right)$ to be as closed as possible.

4. However, it might be necessary for the target agent to learn some aspects of the skill from scratch, since not all intricacies will transfer in the presence of morphological differences.

5. They thus use a RL algorithm to learn π_T, but add an additional term to shape the reward using $f\left(s_{Sp,r}\right)$:

$$r_{\text{ transfer }}\left(s_{T,r}^{(t)}\right)=\alpha\left\|f\left(s_{S,r}^{(t)}; \theta_f\right)-g\left(s_{T,r}^{(t)}; \theta_g\right)\right\|_2$$

where $s_{S,r}^{(t)}$ is the agent-specific state along the optimal policy in the source domain at time step t and $s_{T,r}^{(t)}$ is the agent-specific state along the current policy that is being learned in the target domain at time step t.

6. They showed that in sparse reward environments, where task performance is highly dependent on directed exploration, this incentive to match the trajectory distribution in the embedding space provides a strong guidance for task performance.