Paper

- https://arxiv.org/ftp/arxiv/papers/2202/2202.08444.pdf

Preliminaries

- Definition 1
    - Defines domain and task
    - D_{XY} is a domain
    - X = {x_1,x_2,...,x_m}
    - Y = {y_1, y_2, … y_n}
    -
- Definition 2
    - DA = domain adaptation
    - TA = task adaptation
- Definition 3
- Markov Decision Processes
    - RL settings formulated by using MDP with tuple (S, A, p_s, r)
    - S is a either discrete or continuous state space
    - A = discrete or continuous action space
    - P_s = transition function
    - R = reward function
- Policy
    - $\pi(a \mid s)$

Model vs model free

Imagine training a fixed robot to throw a basketball into a fixed hoop. If you use model-free RL, then the bot would throw the ball in all possible directions (including vertically up and down) until it gets some reward in a particular direction due to the basket's presence.

On the other hand, a model-based RL could be something like the bot first notices that there is a force acting downwards on the object it's throwing (gravity), and makes similar generalizations to make a model of the environment and then uses that to decide where to throw the ball. This probably reduces the number of trials significantly in the long run for problems like these.

DRL based approaches for adaptation and generalization

- Adaptation -
    - Seems most rl algorithm are for adaptation

- Meta rl is for more generalization
- Offline and meta rl seems hot
-

Hierarchical RL (HRL)

- Having multiple levels of policies
  - With similar related problems tasks
  - Planner and
  -
- For problems that have smaller subproblems
- https://towardsdatascience.com/hierarchical-reinforcement-learning-56add31a21ab
- Vezhnevets et al 2017: Transition Policy Gradient Update for 2 Level hierarchical network
  - Creates a manager network that learns space and sets goals
  - Worker network that produces basic actions based on goals from manager
- Nachum et al: method for learning hierarchy of policies
  - Divides low level policies and high level policies
- Haarnoja et al: framework to automate hierarchy construction

Meta RL

- Based on idea of Learning to learn
- Used for developing algorithms that can generalize to unseen tasks
- Li et al 2018 - first attempts to develop a model agnostic algorithm
- Yoon et al 2018 - first bayesian fast adaptation method to solve overfitting problem in the vanilla meta learning algorithms
  - Adapt to unseen tasks by approximating the uncertainties in the them using **Chaser Loss**
  - Goal was to use meta learning methods to achieve faster adaptation to unseen tasks
- Song et al 2020: learning of adaptable policies
  - Batch hill climbing opreator integration with ES MAML
  -
- Flennerhag et al 2021: algorithm to overcome 2 bottlenecks
  - Bottleneck 1: meta objective constraint to the similar geometry of the learners
  - Bottleneck 2:

Skill Discovery

- Skill = latent conditioned policy that can be trained to perform useful tasks in a space or unknown reward environment

- Achiam et al 2018: variational inference based options discovery method for training agent to discover and learn skills through environment interaction without need for maximizing cumulative reward for a task
- Eysenbach et al 2018: reward free skill method using information theoretic objective with maximum entropy policy
- Coreyes et al 2018: novel hierarchical rl algorithm
    - Can learn skills in a continuous latent space
- Bagaria and knooidaris 2020: deep learning chaining algorithm by unifying neural networks
-

Imitation Learning

- Allows agent to learn by observing an expert doing the task
- Hussein et al 2017: off policy actor critci algorithm that focuses on learning to imitate agents past good experiences
    - Uses a replay buffer to store past experiences
    - Sort of off policy ? cuz learning from demonstrations .

Inverse RL:

Given $\pi^*$and T, can we recover R?
More generally, given execution traces, can we recover R?

Model based RL

- able to learn in a sample efficient manner by performing the policy optimization against the learned dynamics of the environment.
- However, model-free RL methods are generally considered highly sample-inefficient

Sim to Real Transfer

Inverse RL

Offline rl

- Enables policy learning from a pre collected data set where agent is not allowed to interact with environment
- Q learning
- Typical examples of model-free algorithms include Monte Carlo RL, Sarsa, and Q-learning.
-
-

Online

- Interact w env

Questions

-

More reading

- https://arxiv.org/abs/1609.05140?source=post_page-----56add31a21ab--------------------------------
- https://towardsdatascience.com/hierarchical-reinforcement-learning-a2cca9b76097
- https://arxiv.org/pdf/2301.08028.pdf