Domain Gap

source domain/environment

\[\text{X}\]

target domain/environment
Domain randomization/Meta-learning

- Train a robust/meta policy by sampling configurations of the source environments from a certain distribution.

\[ \pi^* = \arg\max_{\pi} \mathbb{E}_{\theta \sim p_{\theta}} \mathbb{E}_{M_{\theta}} J(\pi) \]

Tobin et al., 2017; Mordatch et al., 2015; Antonova et al., 2017; Chebotar et al., 2019
Finn et al., 2017; Nagabandi et al., 2018
Domain randomization/Meta-learning

- Usually need a very large amount of training in the source environment.
- Perform suboptimally when the target environment lies out of the training distribution.

Tobin et al., 2017; Mordatch et al., 2015; Antonova et al., 2017; Chebotar et al., 2019
Finn et al., 2017; Nagabandi et al., 2018
Policy adaptation

- Suppose we have a policy that achieves high rewards in one source environment
- Use the source policy and source environment as guidance for adaptation
- The two MDPs share the state space and reward function
Imitation learning

Learning from expert action  
Ross et al., 2011

Learning from expert observation  
Tobin et al., 2017; Tobin et al., 2019; Sun et al. 2019; Yang et al., 2019

Recover source policy’s trajectory in the source environment  \textit{(policy adaptation)}!
Policy adaptation with data aggregation

- The source MDP: $\mathcal{M}^{(s)} := \{\mathcal{S}, \mathcal{A}^{(s)}, f^{(s)}, H, R\}$
- The target MDP: $\mathcal{M}^{(t)} := \{\mathcal{S}, \mathcal{A}^{(t)}, f^{(t)}, H, R\}$
- The source policy: $\pi^{(s)}$
- Our goal is just to learn a model $\hat{f}$ that well approximates $f^{(t)}$
- The target policy:

$$\pi^{(t)}(s) \triangleq \arg\min_{a \in \mathcal{A}^{(t)}} \| \hat{f}(\cdot|s, a) - f^{(s)}(\cdot|s, \pi^{(s)}(s))\|$$
Policy adaptation with data aggregation

\[ \hat{f} \quad \rightarrow \quad \pi^{(t)}(s) \triangleq \arg\min_{a \in A^{(t)}} \| \hat{f}(\cdot | s, a) - f^{(s)}(\cdot | s, \pi^{(s)}(s)) \| \]

Train $\hat{f}$ with $D$

Collect data with $\pi^{(t)}$ in

Store $\{s, a, s'\}$ into $D$

\[ \hat{f}_{e+1} = \arg\max_{f \in \mathcal{F}} \sum_{s, a, s' \in D} \log f(s' | s, a) \]
How fast can we adapt?

\[
\| f^{(s)}(\cdot | s, a) - f^{(t)}(\cdot | s, a') \| \leq \epsilon_{s,a}
\]

\[ f^{(t)} \in \mathcal{F} \]
Main result

Source policy’s trajectory distribution in source env

\[ \left\| \rho_{\hat{\pi}} - \rho_{\pi(s)}^{(s)} \right\| \leq O \left( H A T^{-1/2} + H \epsilon \right) \]

# iterations

\[ |A(t)| \]

Horizon

\[ \mathbb{E}_{s \sim d_{\hat{\pi}}} \left[ \epsilon_{s,\pi(s)}(s) \right] \]

target policy’s trajectory distribution in target env
A practical algorithm

Previously we assumed two oracles:

\[ \hat{f}_{e+1} = \arg\max_{f \in \mathcal{F}} \sum_{s,a,s' \in \mathcal{D}} \log f(s'|s,a) \]

\[ \pi^{(t)}(s) \triangleq \arg\min_{a \in \mathcal{A}^{(t)}} \| \hat{f}(.|s,a) - f^{(s)}(.|s,\pi^{(s)}(s)) \| \]
The deviation model

Objective: \( \Delta \pi^{(s)}(s, a) \triangleq \hat{f}^{(s)}(s, \pi^{(s)}(s)) - f^{(t)}(s, a) \)

Model \( \mathcal{F} = \{ \delta_\theta(s, a) + \hat{f}^{(s)}(s, \pi^{(s)}(s)), \forall s, a : \theta \in \Theta \} \)

Replay buffer & SDG

\[
\theta \leftarrow \theta - \frac{\eta}{|B|} \nabla_\theta \left( \sum_{i=1}^{|B|} \left\| \hat{f}^{(s)}(s_i, \pi^{(s)}(s_i)) + \delta_\theta(s_i, a_i) - s_i' \right\|^2_2 \right)
\]
CEM

- Just use the output of DM as cost!

\[
\underset{a \in A_t}{\text{argmin}} \| \delta_{\theta}(s, a) \|_2
\]

- One step look-ahead is enough!
Results

Source

Target
Results

Our methods

Christiano et al. 2016

PPO

Zhu et al. 2016
Thank you!

Website: https://yudasong.github.io/PADA

Code: https://github.com/yudasong/policy_adapt