1703.09327 - DART: Noise Injection for Robust Imitation Learning

- **Yunqiu Xu**
- Other resource:
  - http://bair.berkeley.edu/blog/2017/10/26/dart/
  - https://github.com/DanielTakeshi/Paper_Notes/blob/master/reinforcement_learning/DART_Noise_Injection_for_Robust_Im

---

1. Introduction

- Challenges of imitation learning:
  - For behavior cloning (off-policy), need superviser's demonstration
  - For on-policy methods, need human superviser $\rightarrow$ computation burder
- Our work:
  - Focus on **off-policy**, try to improve the performance of **behavior cloning**
  - Add noisy into supervisor's policy during demonstrating $\rightarrow$ demonstrate how to recover from errors
  - DART: Disturbances for Augmenting Robot Trajectories
    - Collect demonstrations with injected noise
    - Optimize the noise level to approximate the error of the robot’ s trained policy during data collection

2. Related Work

- Off-policy:
  - e.g. behavior cloning, ALVINN for self-driving
  - The robot passively observes the supervisor, then learns a policy mapping states to controls by approximating the supervisor’ s policy
  - Limitation: 不好举一反三, 和教的有一点点不一样就傻逼了
- On-policy:
  - e.g. DAgger, supervisor iteratively provides corrective feedback on the robot’ s behavior
  - Alleviate the problem of compounding errors
  - Limitation:
    - Providing feedback : human supervisors
    - Safety : require the robot to visit potentially dangerous region
    - Computation : require retraining the policy from scratch after each round of corrections

3. Problem Statement

- Policy π_θ : probability density over the set of trajectories of length T
  - x : state
  - u : action
  - $\xi = (x_0, u_0, x_1, u_1, \dots, x_T, u_T)$: trajectory, a finite sequence of T pairs of states visited and corresponding control inputs at these states

$$p(\xi|\pi_\theta) = p(x_0) \prod_{t=0}^{T-1} \pi_\theta(u_t|x_t) p(x_{t+1}|u_t, x_t)$$

- Imitation learning:
  - Surrogate loss: the difference between controls, $l(u_1, u_2) = ||u_1 - u_2||_2^2$
  - Total loss: $J(\theta_1, \theta_2|\xi) = \sum_{t=0}^{T-1} l(\pi_{\theta_1}(x_t), \pi_{\theta_1}(x_t))$

- Try to minimize expected surrogate loss along the distribution induced by the robot's policy
- The distribution on trajectories and the cumulative surrogate loss are coupled $\rightarrow$ hard to optimize

$$min_\theta E_{p(\xi|\pi_\theta)} J(\theta, \theta^*|\xi)$$

- Transform to behavior cloning $\rightarrow$ off policy
  - Sample from the supervisor's distribution
  - Performs expected risk minimization on the demonstrations

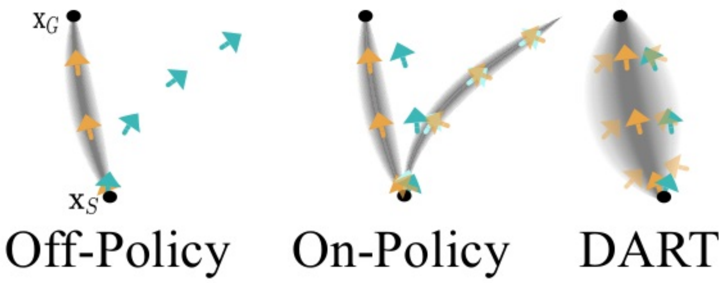
$$\theta^R = argmin_\theta E_{p(\xi|\pi_{\theta^*})} J(\theta, \theta^*|\xi)$$

- Thus the performance of the policy θ^R can be written as the sum of covariate shift and the standard loss

$$\begin{aligned} & E_{p(\xi|\pi_{\theta^R})} J(\theta^R, \theta^*|\xi) \\ &= \underbrace{E_{p(\xi|\pi_{\theta^R})} J(\theta^R, \theta^*|\xi) - E_{p(\xi|\pi_{\theta^*})} J(\theta^R, \theta^*|\xi)}_{\text{Shift}} + \underbrace{E_{p(\xi|\pi_{\theta^*})} J(\theta^R, \theta^*|\xi)}_{\text{Loss}}, \end{aligned}$$

- In this work, we focus on minimizing **covariate shift**

4. Off-Policy Imitation Learning with Noise Injection



- Robot tries to reach X_G, grey denotes the distribution over trajectories
  - Off-Policy:
    - The supervisor (orange arrows), provides demonstrations
    - The robot (teal arrows), deviates from the distributions and incurs high error
  - On-Policy:
    - Samples from the current robot's policy (light teal arrows) to receive corrective examples from the superviser
    - Computation expensive and unsafe
  - DART:
    - Injects noise to widen supervisor's distribution $\rightarrow$ provide corrective examples
    - Off-policy but robust
- DART:
  - $p(\xi|\pi_{\theta_R})$ is not known until the robot has been trained
  - We interatively sample from the superviser's distribution with current noise parameter ψ_k

$$\hat{\psi}_{k+1} = \underset{\psi}{\text{argmin}}\ E_{p(\xi|\pi_{\theta^*}, \psi_k)} - \sum_{t=0}^{T-1} \log \left[\pi_{\theta^*}(\pi_{\hat{\theta}}(\mathbf{x}_t)|\mathbf{x}_t, \psi)\right] \quad (3)$$

$$\psi^\alpha_{k+1} = \hat{\psi}_{k+1} * \underset{\beta \geq 0}{\text{argmin}}\ |\alpha - E_{p(\xi|\pi_{\theta^*}, \beta * \hat{\psi}_{k+1})} \sum_{t=0}^{T-1} l(\mathbf{u}_t, \pi_{\theta^*}(\mathbf{x}_t))| \quad (4)$$

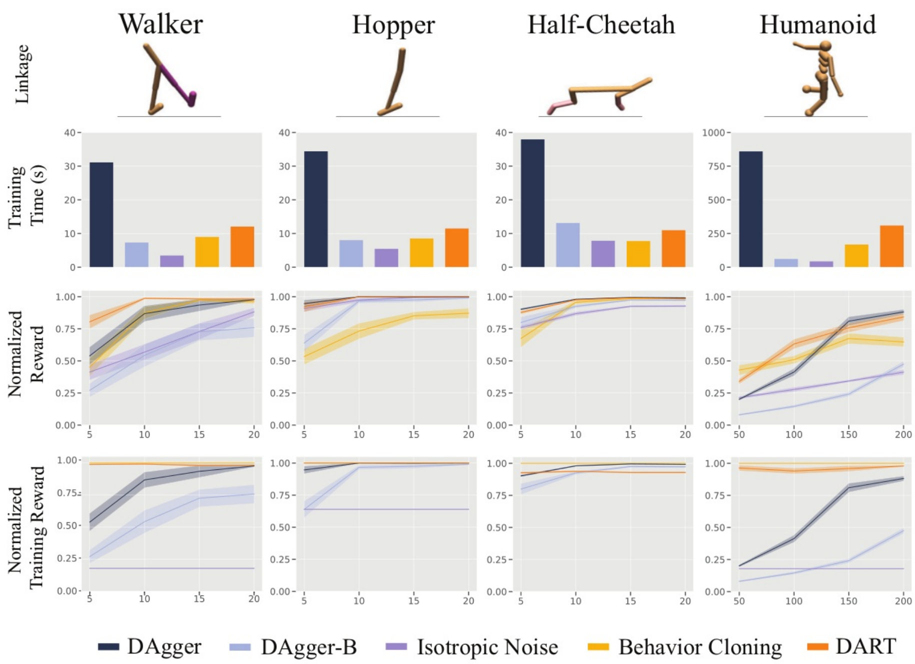
  - Pseudo code

Algorithm 1: DART

Input: ψ_1^α, α
for $k = 1$ **to** K **do**
  for $n = 1$ **to** N **do**
    $\xi_{k,n} \sim p(\xi | \pi_{\theta^*}, \psi_k^\alpha)$
  end for
  $\hat{\theta} = \underset{\theta}{\arg\min} \sum_{i=1}^{k} \sum_{n=1}^{N} J(\theta, \theta^* | \xi_{i,n})$
  $\hat{\psi}_{k+1}$ is set with Eq. 3
  ψ_{k+1}^α is set with Eq. 4
end for
$\theta^R = \underset{\theta}{\arg\min} \sum_{k=1}^{K} \sum_{n=1}^{N} J(\theta, \theta^* | \xi_{k,n})$

5. Experiments

- Questions:
  - Does DART reduce covariate shift as effectively as on-policy methods
  - How much does DART reduce the computational cost
  - How much does it decay the supervisor’ s performance during data collection
  - Are human supervisors able to provide better demonstrations with DART
- MuJoCo Locomotion



- Robotic Grasping in Clutter

Figure 3: Left: Experimental setup for the grasping in clutter task. A Toyota HSR robot uses a head-mounted RGBD camera and its arm to push obstacle objects out of the way to reach the goal object, a mustard bottle. The robot's policy for pushing objects away uses a CNN trained on images taken from the robot's Primesense camera, an example image from the robot's view point is shown in the orange box. Right: the Success Rate for Behavior Cloning, DART($\alpha = 3$) and DART($\alpha = 6$). DART($\alpha = 3$) achieves the largest success rate.

6. Conclusion

- Add noise to broaden supervisor's demonstration
- Try to make off-policy imitation learning (behavior cloning) more robust