

PLATO: Policy Learning using Adaptive Trajectory Optimization

Yuwei Chen

March 5th, 2019

Overview

- 1 Introduction and Related Work
- 2 Preliminary and Overview
- 3 Policy Learning Using Adaptive Trajectory Optimization
- 4 Experiment

PLATO algorithm tries to imitate MPC teacher which could overcome following challenges:

- RL-based training of large NN needs large amount of experience (data). In contrast, supervised-learning methods, such as DAgger and GPS require a viable source of supervision.
- IN RL-based training, partially trained controller will perform unreasonable and unsafe actions which can cause the destruction of robot or damage to its surroundings.

Note that MPC teacher has access to all underlying states while learner policy could only act on observations.

Problem Set-up

- states \mathbf{x} , actions \mathbf{u} .
- The policy could only control the system from observations \mathbf{o} .
- The policy $\pi_{\theta}(\mathbf{u}|\mathbf{o}_t)$, parametrized by θ .
- At test time, the agent chooses actions according to $\pi_{\theta}(\mathbf{u}|\mathbf{o}_t)$ at each time step t , and experiences a loss $c(\mathbf{x}_t|\mathbf{o}_t) \in [0, 1]$.
- The next state is distributed by dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$.

The objective is to learn policy $\pi_{\theta}(\mathbf{u}|\mathbf{o}_t)$ s.t.

$$\arg \min_{\pi} J(\pi) = \mathbf{E}_{\pi} \left[\sum_{t=1}^T c(\mathbf{x}_t, \mathbf{u}_t) \right].$$

At the same time, let's define expected cost from state \mathbf{x}_t at time t as

$$J(\pi|\mathbf{x}_t) = \mathbf{E}_{\pi} \left[\sum_{t=1}^T c(\mathbf{x}_t, \mathbf{u}_t) | \mathbf{x}_t \right].$$

Adaptive MPC teacher

One naive way is to train the policy with supervised learning from data generated from MPC teacher. However, because state distribution for the teacher and learner are different. Learned policy might fail.

In order to overcome this challenge, an adaptive MPC teacher is used which generates actions from controller obtained by:

$$\pi_{\lambda}^t(\mathbf{u}|\mathbf{x}_t, \theta) \leftarrow \arg \min_{\pi} J_t(\pi|\mathbf{x}_t) + \lambda D_{KL}(\pi(\mathbf{u}|\mathbf{x}_t) || \pi_{\theta}(\mathbf{u}|\mathbf{o}_t)) \quad (1)$$

where λ determines the relative importance of matching the learner policy versus optimizing the expected return. Note that the particular MPC algorithm is based on iLQG.

Algorithm 1 PLATO algorithm

Initialize data $D \leftarrow \emptyset$

for $i = 1$ to N **do**

for $t = 1$ to T **do**

$\pi_{\lambda}^t(\mathbf{u}_t|\mathbf{x}_t, \theta) \leftarrow \arg \min_{\pi} J_t(\pi|\mathbf{x}_t) + \lambda D_{KL}(\pi(\mathbf{u}|\mathbf{x}_t)||\pi_{\theta}(\mathbf{u}|\mathbf{o}_t)).$

 Sample $\mathbf{u}_t \sim \pi_{\lambda}^t(\mathbf{u}|\mathbf{x}_t, \theta).$

$\pi^*(\mathbf{u}_t|\mathbf{x}_t) \leftarrow \arg \min_{\pi} J(\pi).$

 Sample $\mathbf{u}_t^* \sim \pi^*(\mathbf{u}|\mathbf{x}_t).$

 Append $(\mathbf{o}_t, \mathbf{u}_t^*)$ to dataset D .

 State evolves $\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t).$

end for

 Train π_{θ} on D .

end for

Adaptive MPC teacher revisited

Our iLQG-based MPC algorithm produces linear-Gaussian local controller $\pi_{\lambda}^t(\mathbf{u}_t|\mathbf{x}_t) = N(\mu_{\lambda}(\mathbf{x}_t), \Sigma_t)$ where $\mu_{\lambda} = \mathbf{K}_t \mathbf{x}_t + \mathbf{k}_t$.

Assume our learner policy is conditionally Gaussian i.e.

$\pi_{\theta}(\mathbf{u}|\mathbf{o}_t) = N(\mu_{\theta}(\mathbf{o}_t), \Sigma_{\pi_{\theta}})$ where $\mu_{\theta}(\mathbf{o}_t)$ is the output of nonlinear function, e.g. NN. Then the MPC objective can be expressed in closed form:

$$\min_{\pi} J_t(\pi|\mathbf{x}_t) + \frac{1}{2} \lambda \left[\ln \left(\frac{|\Sigma_{\pi_{\theta}}|}{|\Sigma_t|} \right) + \text{tr}(\Sigma_{\pi_{\theta}}^{-1} \Sigma_t) + \|\mu_{\lambda}^*(\mathbf{x}_t) - \mu_{\theta}(\mathbf{o}_t)\|_{\Sigma_{\pi_{\theta}}^{-\frac{1}{2}}}^2 + \text{const} \right].$$

Training the learner's policy

During the supervised learning phase, we minimize the KL-divergence between the learner policy π_θ and precomputed near-optimal policies π^* which is estimated by iLQG:

$$\theta \leftarrow \arg \min_{\theta} \sum_{(\mathbf{x}_t, \mathbf{o}_t) \in D} D_{KL}(\pi_\theta(\mathbf{u}|\mathbf{o}_t) \parallel \pi^*(\mathbf{u}|\mathbf{x}_t)).$$

Since both π_θ and π^* are conditionally Gaussian, the KL divergence could be expressed in closed-form if ignoring the terms not involving the learner policy means $\mu_\theta(\mathbf{o}_t)$.

$$\min_{\theta} \sum_{(\mathbf{x}_t, \mathbf{o}_t) \in D} \|\mu^*(\mathbf{x}_t) - \mu_\theta(\mathbf{o}_t)\|_{\Sigma_{\pi^*}^{-\frac{1}{2}}}^2.$$

In this paper, μ_θ is represented by a NN, and solved by SGD.

Theoretical Analysis

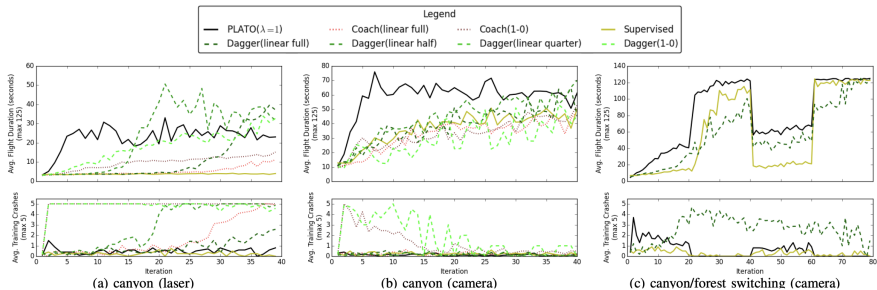
Let $Q_t(\mathbf{x}, \pi, \tilde{\pi})$ denote the cost of executing π for one time step starting from initial state, and then executing $\tilde{\pi}$ for the remaining $t - 1$ time steps. We assume the cost-to-go difference between the learned policy and the optimal policy is bounded $Q_t(\mathbf{x}, \pi, \pi^*) - Q_t(\mathbf{x}, \pi^*, \pi^*) \leq \delta$

Theorem

Let the cost-to-go $Q_t(\mathbf{x}, \pi, \pi^*) - Q_t(\mathbf{x}, \pi^*, \pi^*) \leq \delta$ for all $t \in \{1, \dots, T\}$. Then for PLATO, $J(\pi_\theta) \leq J(\pi^*) + \delta\sqrt{\epsilon_{\theta^*}}O(T) + O(1)$.

Therefore, the policy learned by PLATO converges to a policy with bounded cost.

Experiment



Comparison to DAgger

PLATO could be viewed as a generalization of DAgger, which samples from mixture policy

$$\pi_{mix,i} = \beta_i \pi^* + (1 - \beta_i) \pi_{\theta_i}.$$

Differences with the DAgger:

- (1) The training data is labelled with actions from π^* .
- (2) PLATO uses adaptive MPC policy to select actions at each time step, rather than the mixture policy $\pi_{mix,i}$ used.

- The learned policy does not need to be executed during training, because of the robustness of MPC. It minimizes the catastrophic failures.
- Learned policy can use a different set of observations than MPC because the policy is directly trained on raw input from onboard sensor.

The End

<https://www.youtube.com/watch?v=c1Hp6QgVyAU>