
NICOLA PITZALIS
Trust Region Policy Optimization
INTELLIGENT SYSTEM FOR PATTER RECOGNITION

Introduction to the problem

In this paper, Schulman et al. extended the work of Kakade & Langford (2002) to obtain an algorithm able to optimize large nonlinear policies such as neural networks.

We want to minimize the expected discounted cost of the new policy $\eta(\tilde{\pi})$:

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)$$

discounted visitation frequencies

new policy

advantage function

But unfortunately, the complex dependencies between the new policy and the visitation frequencies make the equation hard to be optimize. Instead, a local approximation will be used:

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)$$

old discounted visitation frequencies

If we use a parametrized policy π_θ then L_π matches η to first order and hence, for a small enough step, improving L_π also improves η. Kakade & Langford proved that, for a mixture policy, the following bound holds:

$$\eta(\pi_{\text{new}}) \leq L_{\pi_{\text{old}}}(\pi_{\text{new}}) + \frac{2\epsilon\gamma}{(1-\gamma(1-\alpha))(1-\gamma)}\alpha^2$$

maximum expected advantage

discount factor

mixture policy

blending parameter of the mixture policy

Constrained problem reformulation

The main theoretical result of the paper is the improvement of the bound proposed by Kakade & Langford. Specifically, Schulman et al. proved that the bound can be extended to general stochastic policies, rather than mixture of policies (hardly used in practice).

$$\eta(\tilde{\pi}) \leq L_{\pi}(\tilde{\pi}) + C D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$

$$C = \frac{2\epsilon\gamma}{(1-\gamma)^2}$$

follows directly from the previous bound

maximum KL-divergence between the two stochastic policies

Then it follows, that minimizing the equation above, consequently minimizes the expected discounted cost of the new policy. We then must solve the unconstrained problem $\min_{\theta} [L_{\theta_{\mathrm{old}}}(\theta) + C D_{\mathrm{KL}}^{\max}(\theta_{\mathrm{old}}, \theta)]$. In practice, though, the constanc C is very small, resulting in trivial optimization steps.
The authors proposed the optimization of an equivalent constrained problem, where the constraint is the *trust region* constraint:

KEY IDEA

$$\min_{\theta} L_{\theta_{\mathrm{old}}}(\theta)$$

$$\text{subject to } \overline{D}_{\mathrm{KL}}^{\rho^{\theta_{\mathrm{old}}}}(\theta_{\mathrm{old}}, \theta) \leq \delta.$$

trust region constraint

average KL-divergence

Sample-Based Estimation

The problem reformulation permits us to use Monte Carlo simulation to approximate both the objective and the constraint.
After some minor mathematical manipulations, we end up with the following problem:

$$\min_{\theta} \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}},a\sim q}\left[\frac{\pi_\theta(a|s)}{q(a|s)}Q_{\theta_{\text{old}}}(s,a)\right]$$
$$\text{subject to } \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}\left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s)\parallel\pi_\theta(\cdot|s))\right]\leq\delta.$$

- we use importance sampling estimation to substitute the summation over the entire action space, thus the action a is drawn from an auxiliary distribution q (the only assumption needed is coverage; in general one can use $\pi_{\theta\,old}$)
- we substitute the advantage function $A_{\theta\,old}$, inside L_π, with the action-value function $Q_{\theta\,old}$ (which is admissibile since it changes the optimization by a constant factor)
- we sample the state s from the old distribution $\rho_{\theta\,old}$

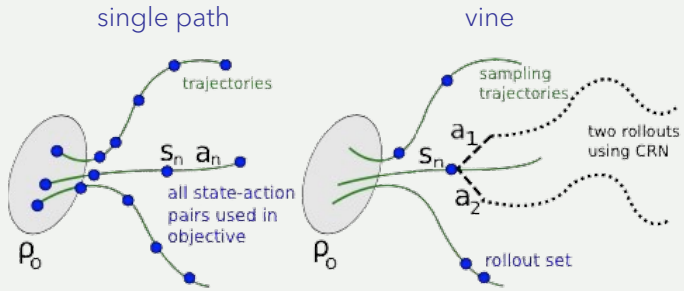
Proposed methodologies

Single Path

- generate a single trajectory (per episode) $s_0, a_0, \ \dots\ , \ s_T, a_T$ using the policy $\pi_{\theta_{old}}$
- compute $Q_{\theta_{old}}$ at each state-action pair in the trajectory, by taking the discounted sum of future costs along the trajectory

Vine

- generate a number of trajectories using the policy π_{θ_i}
- choose a subset of *N* states along the trajectories – *rollout set*
- for each state s_n in the rollout set, sample *K* actions $a_{n,k} \sim q(\cdot\,|s_n)$
  - $q(\cdot\,|s_n) = \pi_{\theta_i}(\cdot\,|s_n)$ for continuous problems (e.g. robotic locomotion)
  - $q(\cdot\,|s_n) = Uniform$ for discrete problems (e.g. atari games)
- for each action $a_{n,k}$ sampled at each state s_n, estimate $\hat{Q}_{\theta_i}(s_n, a_{n,k})$ by performing a rollout starting from state s_n and action $a_{n,k}$ (use *Common Random Numbers* to reduce the variance on Q-values)

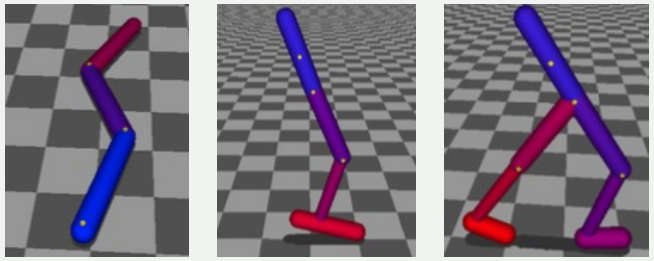


After collecting the Q-values, average over samples and solve the constrained optimization problem to update the policy's parameters.

Experiments (Simulated Robotic Locomotion)

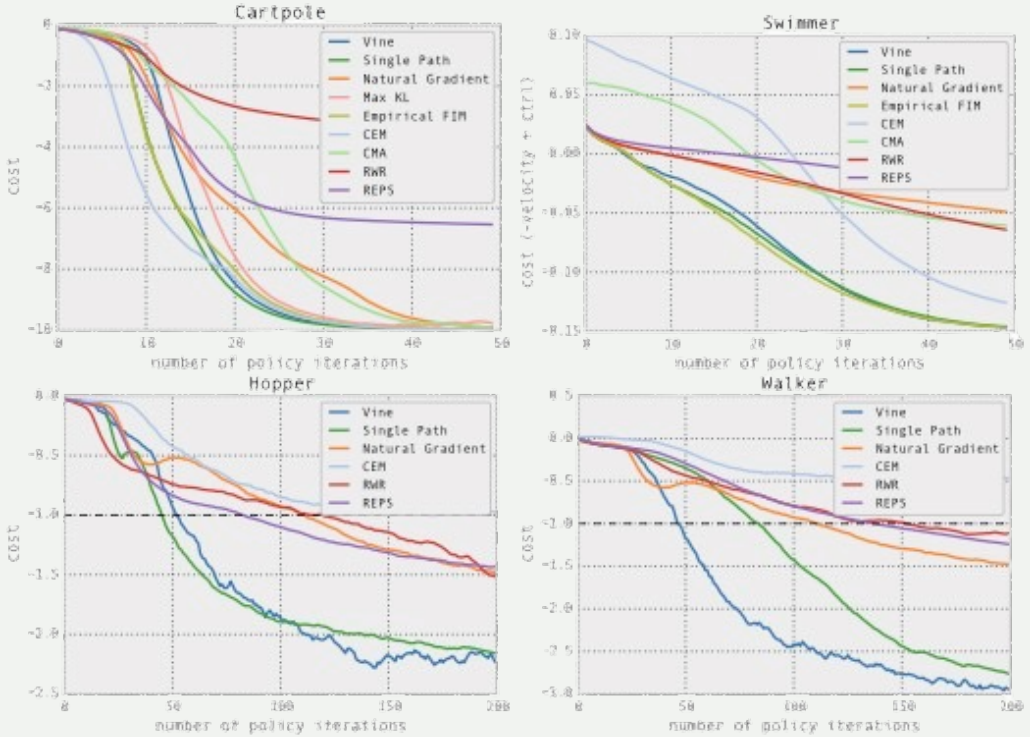
Experiments comprised *carpole balancing* (as a standard baseline), *swimmer*, *hopper* and *walker*.

swimmer hopper walker



The learning curves on the right show the cost averaged across five runs of each algorithm. *Single path* and *vine* TRPO solved the tasks yelding the best results between the compared algorithms.

TRPO learned with general purpose policies and simple cost functions, using minimal prior knowledge (in contrast to typical locomotion learning).



Experiments (Atari games from images)

Here, the authors trained policies for playing Atari games, using raw images as input.
Some challenges arise from the task at hand:

- the games require learning different behaviours
- delayed rewards
- some games require complex sequence of actions
- non-stationary image statistics

TRPO performed better than some of the other algorithms to which it has been compared and, in general, achieved reasonable results.

	B. Rider	*Breakout*	*Enduro*	*Pong*	*Q*bert*	*Seaquest*	*S. Invaders*
Random	354	1.2	0	−20.4	157	110	179
Human (Mnih et al., 2013)	7456	31.0	368	−3.0	18900	28010	3690
Deep Q Learning (Mnih et al., 2013)	4092	168.0	470	20.0	1952	1705	581
UCC-I (Guo et al., 2014)	5702	380	741	21	20025	2995	692
TRPO - single path	1425.2	10.8	534.6	20.9	1973.5	1908.6	568.4
TRPO - vine	859.5	34.2	430.8	20.9	7732.5	788.4	450.2

Conclusion

The authors provided strong theoretical foundations that not only serve as pillars for the proposed algorithms but also unify policy gradient and policy iteration methods, showing them to be special limiting cases of an algorithm that optimizes a certain objective subject to a trust region.

Some brightful ideas:

- transforming the unconstrained formulation into a constrained problem
- exploiting the trust region constraint over the first order approximation
- exploting Monte-Carlo estimation to solve a complex optimization problem

Some weaknesses:

- the *single path* algorithm yields high variance results
- the *vine* algorithm lowers the variance but it is way more computationally expensive (given the MC simulations it must perform)

Strong points:

- sound theoretical fundation
- strong empirical results: the algorithm is able to learn very complex policies (such as the CNN used for the Atari game and the controller for the simulated robotic locomotion) using generic policy search method and non-engineered policy representations.