Multi-objective Model-based Policy Search for Data-efficient Learning with Sparse Rewards

Journée "Robotics et Neuroscience", 19 Oct 2018

Rituraj Kaushik • Jean-Baptiste Mouret

Team LARSEN, INRIA Nancy Grand-Est

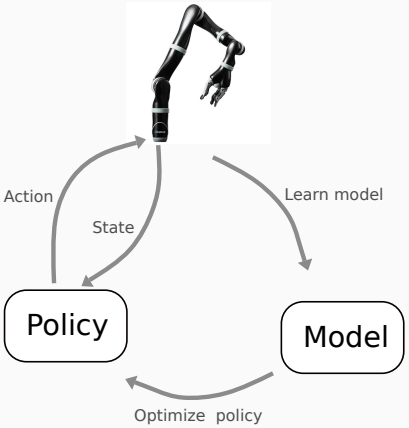
Model Based Policy Search (MB-PS)

MB-PS are the most data-efficient RL algorithms[1].

Examples: **Black-DROPS**[2], **PILCO**[3]

MB-PS involves two steps:

- Learns a dynamical model of the robot.
- Optimizes a policy to maximize the expected return given the model and its uncertainties.



[1] Chatzilygeroudis et. al. "A survey on policy search algorithms for learning robot controllers in a handful of trials." arXiv preprint:1807.02303, 2018.

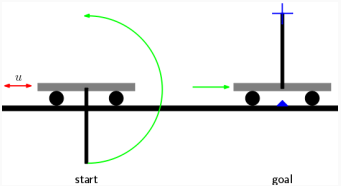
[2] Chatzilygeroudis et. al., Black-Box Data-efficient Policy Search for Robotics. In Proc. of IROS, 2017.

[3] Deisenroth et. al., Gaussian processes for data-efficient learning in robotics and control. IEEE Trans. Pattern Anal. Mach. Intell., 2015.

Model Based Policy Search (MB-PS)

Black-DROPS and PILCO solves standard benchmarking tasks in only few seconds of interaction.

- Eg: Cart-pole swing up task in 20 seconds of interaction

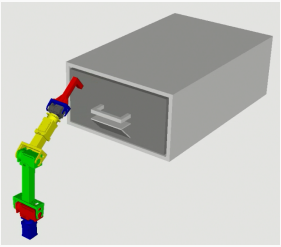


Current MB-PS works great for simple tasks where reward functions are not sparse

Problem with current MB-PS

However

- Many interesting real world tasks might have naturally sparse rewards.
- For example:



Reward function

$reward = k * drawer\_displacement$

i.e Reward is proportional to drawer displacement

- Current MB-PS such as PILCO , Black-DROPS do not have explicit strategy to deal with sparse reward.

How to Deal with Sparse Rewards in Robot Learning?

RL with Continuous State/Action

- **Action perturbation** and **policy parameter perturbation** with noise
- **Count-based** approaches, **Bayesian RL** to cover state space as uniformly as possible
- **Hindsight Experience Replay**[4]: learn from unsuccessful episodes also by associating them to a pseudo goal
- **Reverse curriculum**[5]: Learn form an easy initial state and progressively start from more difficult state
- **Intrinsic reward**: Encourage actions by additional reward that reduces the uncertainty about the agent's prediction of its actions.

[4] Marcin Andrychowicz et. al., Hindsight experience replay. In Proc. of NIPS, 2017

[5] Carlos Florensa et. al., Reverse curriculum generation for reinforcement learning. In Conference on Robot Learning, 2017.

Developmental and Evolutionary Robotics

Idea is to cover a user defined space called "Outcome space" as uniformly as possible

- **Novelty search**[6]: Search for policy that gives novel outcomes.
- **Quality-diversity**[7,8]: Search for diverse high performing solutions.
- **Curiosity driven learning**[9]: Accomplish goals progressively based on increasing difficulty.

[6] Joel Lehman et. al., Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation, 2011.

[7] Antoine Cully et. al., Quality and diversity optimization: A unifying modular framework. IEEE Trans. on Evolutionary Computation, 2018.

[8] Jean-Baptiste Mouret et. al., Illuminating search spaces by mapping elites. arxiv:1504.04909, 2015.

[9] Sebastien Forestier et. al., Curiosity-driven development of tool use pre-cursors: a computational model. In Proc. of COGSCI, 2016.

How to Deal with Sparse Rewards in Robot Learning?

Unfortunately, all the these approaches need thousands of trials to find a working policy.

For example:

TRPO-VIME

Around 10,000 trials to solve cart-pole swing up task with sparse reward

Hindsight Experience Replay (HER)

Around 8,000 trials to find policy so that a manipulator pushes an object to a desired location

It is infeasible to perform that many trials on real systems

Our motivation

We take the inspiration from 3 ideas:

1. Novelty search

Continues to explore the task-space (or behavior space) even if no reward is observed

2. Model-based policy search

- Optimizes the policy to improve the reward using the learned dynamical model
- Fewer interaction time with the system

3. Pareto-based multi-objective optimization

- No need to aggregate the objectives with different weights.
- Give a set of Pareto-optimal solutions

Multi-objective Data-efficient Exploration: Multi-DEX

- Combining the 3 ideas to address the problem of sparse reward in Robot Learning in a data efficient manner.
- We frame it as a **Pareto based multi-objective model-based policy search** problem with 3 objectives:

1. Maximize novelty

produce maximally novel state trajectories in the system

2. Maximize reward

produce maximally rewarding state trajectories

3. Minimize model prediction variance

keep the system as close as to the more certain region of the model

Problem formulation

We consider the following dynamical systems of the form:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w} \tag{1}$$

$\mathbf{x} \in \mathbb{R}^E$: continuous-valued states

$\mathbf{u} \in \mathbb{R}^F$: continuous-valued controls

$\mathbf{w}$: i.i.d. Gaussian system noise

f : unknown transition dynamics

Problem Formulation

- Our goal is to find a *policy* $\pi(\mathbf{u}|\mathbf{x}, \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^{\Theta}$ that maximizes the $J_r(\boldsymbol{\theta})$

$$J_r(\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{t=1}^{T} r(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) \Big| \boldsymbol{\theta}\right] \tag{2}$$

  where $r() \in \mathbb{R}$ is the reward for being in state $\mathbf{x}_t$, taking action $\mathbf{u}_t$, and reaching state $\mathbf{x}_{t+1}$.

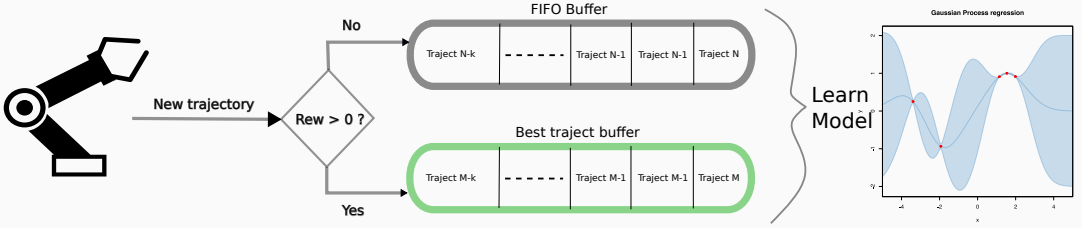
- We assume that r can be sparse or may have plateaus; *i.e.*, it can be zero for most values of $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ or have large regions with constant value.

Multi-DEX Approach: Transition Dynamics and Reward Model learning

- We learn the forward transition dynamics of the system with *Gaussian Process*
- **Training inputs**: tuples of states $\mathbf{x}_t$ and actions $\mathbf{u}_t$ (*i.e.*, $\tilde{\mathbf{x}}_t = (\mathbf{x}_t, \mathbf{u}_t) \in \mathbb{R}^{E+F}$)
- **Training targets**: The difference between the current and the next state vector, $\boldsymbol{\Delta}_{\mathbf{x}_t} = \mathbf{x}_{t+1} - \mathbf{x}_t \in \mathbb{R}^E$
- Then, E independent GPs are used to model each dimension of the difference vector $\boldsymbol{\Delta}_{\mathbf{x}_t}$.
- If required, we additionally learn a reward function $r(\mathbf{x}_t) : \mathbb{R}^E \mapsto \mathbb{R}$ using Random Forest.

Multi-DEX Approach: Learning System Dynamics with Sparse Transitions

- The intuition here is to have a balanced blend of ordinary trajectories and trajectories with rare transitions (leading to high reward) for model learning.
- We maintain two fixed sized buffers to keep non-rewarding and best rewarding trajectories for model learning



Multi-DEX Approach: Exploration-Exploitation Objectives

Cumulative Return

$$\hat{J}_r(\boldsymbol{\theta}) = \sum_{t=1}^{T} r(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{x}_{t-1} + f_\mu(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})) \tag{3}$$

where $\mathbf{u}_{t-1} = \pi(\mathbf{x}_{t-1}|\boldsymbol{\theta})$

Novelty

It is the minimum Euclidean distance to the expected state trajectories (using the current model) of already executed policies from that of the policy to be evaluated. Produce novel state trajectories $\boldsymbol{\beta}_{\boldsymbol{\theta}}$ w.r.t policy parameter $\boldsymbol{\theta}$

$$\hat{J}_n(\boldsymbol{\theta}) = min(||\boldsymbol{\beta}_{\boldsymbol{\theta}} - \boldsymbol{\beta}||^2)_{\forall \boldsymbol{\beta} \in \mathbb{B}} \tag{4}$$

Multi-DEX Approach: Exploration-Exploitation Objectives

Cumulative Model-Variance

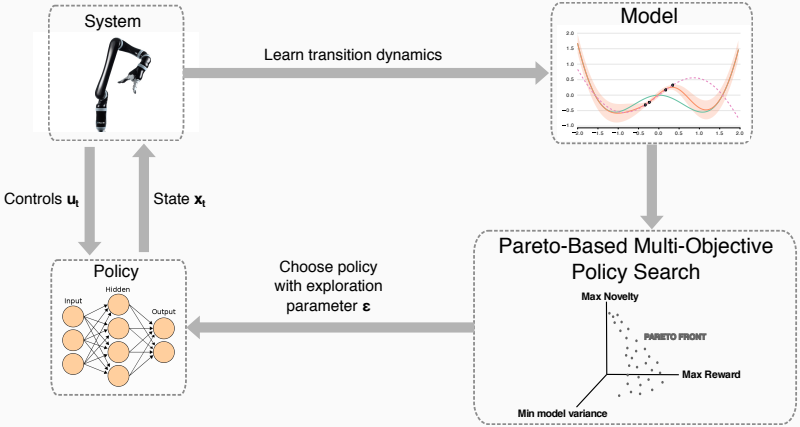
We define the cumulative model-variance for a policy $\pi_{\boldsymbol{\theta}}$ as the negative mean of the step-by-step model prediction variances:

$$\hat{J}_{\sigma^2}(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T}(||\sigma_{\boldsymbol{x_t}}||^2) \tag{5}$$

where $\boldsymbol{x_t}$ is given by applying the policy $\pi_{\boldsymbol{\theta}}$ on the model

Multi-DEX Approach: Multi-Objective Policy Search

We optimize the policy for the objectives $\hat{J}_r(\boldsymbol{\theta}), \hat{J}_n(\boldsymbol{\theta}), \hat{J}_{\sigma^2}(\boldsymbol{\theta})$ using a Pareto based multi-objective optimization algorithm NSGA-II.



Experiments

We compare to several state-of-the-art approaches in a sequential goal reaching task and in a drawer opening task:

1. **Black-DROPS**, a model-based policy search algorithm
2. **TRPO**[10], a model-free policy gradient approach
3. **TRPO** with the **VIME**[11] exploration strategy
4. **CMA-ES**[12], a black-box optimizer effective for direct policy search, and
5. **GEP-PG**[13], an curiosity-driven model-free approach.

[10] John Schulman et. al., Trust region policy optimization. In Proc. of ICML, 2015.

[11] Rein Houthooft et. al., Vime: Variational information maximizing exploration. In Proc. of NIPS, 2016.

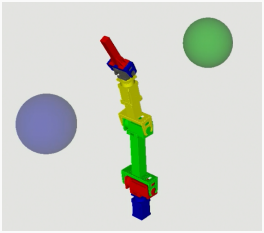
[12] Nikolaus Hansen. The CMA Evolution Strategy: A Comparing Review. Springer, 2006.

[13] Cedric Colas et. al., GEP-PG: Decoupling Exploration and Exploitation in Deep Reinforcement Learning Algorithms. In Proc. of ICML, 2018.

Sequential Goal Reaching with a 2-DOF Robotic Arm

- **Goal**: To reach the green goal while first passing through the blue region.
- **Episode length**: 4 seconds
- **Control frequency**: 10Hz.
- **Policy**: Feed-forward neural network
- **Reward function**: A positive reward is given only when end effector passes through the blue region and is within 0.1m radius of the green goal's centre.

Sequential Goal Reaching with a 2-DOF Robotic Arm



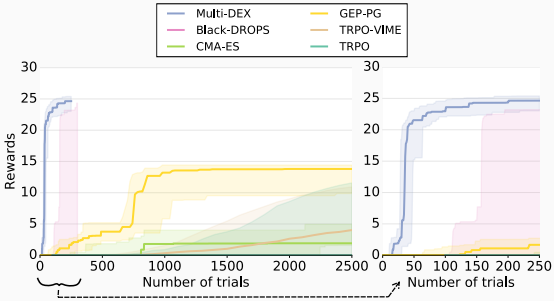
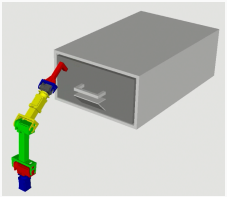


Figure 1: Best reward found per trial (20 replicates). Multi-DEX finds working policies in about only 7 minutes of interaction (around 100 episodes/trials).

Drawer Opening Task with 2-DOF Robotic Arm

- **Goal**: To open a drawer with a 2-DOF robotic arm and to go back to the up-right position.
- **Episode length**: 4 seconds
- **Control frequency**: 10Hz.
- **Policy**: Feed-forward neural network
- **Reward function**: The total reward is composed of two rewards
  1. A small positive reward is given proportional to the drawer displacement
  2. Another reward is given which is inversely proportional current state and target state of the arm if the drawer is already open.

Drawer Opening Task with 2-DOF Robotic Arm



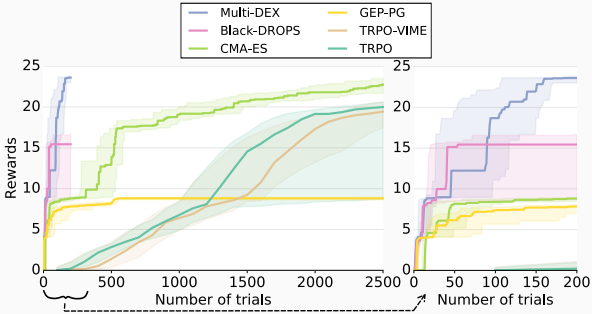


Figure 2: Best reward found per trial (20 replicates). Multi-DEX finds working policies in about only 14 minutes of interaction (200 trials).

Deceptive pendulum swing-up task

- Pendulum powered by an underpowered torque-controlled actuator.
- **Goal**: To swing the pendulum to the upright position applying torques as small as possible (*i.e.*, using minimum power) to the actuator and hold it in that position.
- **Episode length**: 4 seconds
- **Control frequency**: 10Hz.
- **Policy**: Feed-forward neural network
- **Reward function**:
  - A constant positive reward of $+10$ every time-step if the pendulum is in upright position
  - Gets a negative reward proportional to the square of torque for every time step.

Deceptive pendulum swing-up task

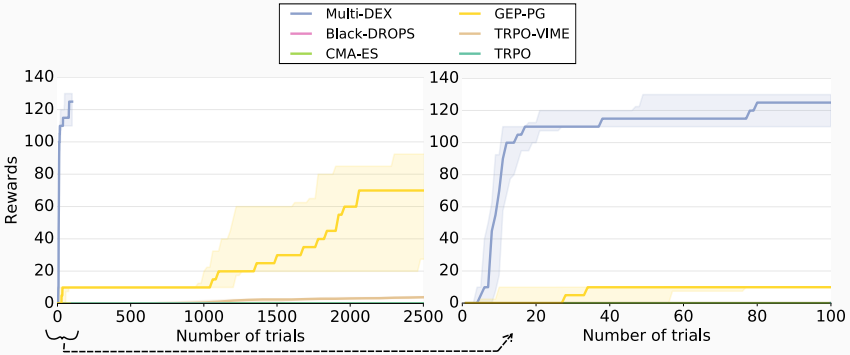


Figure 3: Best reward found vs number of trials plot for Pendulum Swing-up Task. The plot clearly outperforms all the competing approaches and achieves very high reward (balancing the pendulum in upright position) in just 100 trials (approx 6.6 minutes of total interaction)

Short video

Additional Experiments

High dimensional state-space

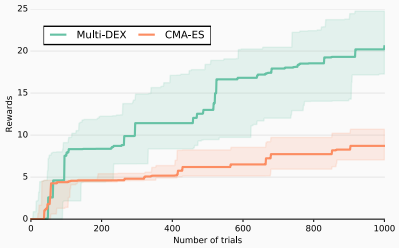


Figure 4: Drawer opening task with 4-DOF arm

Non-sparse reward problem

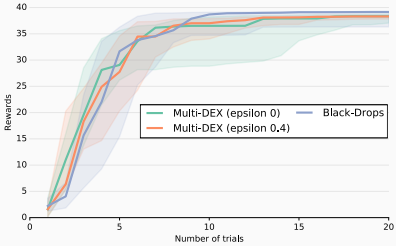


Figure 5: Single goal reaching task with 4-DOF arm. The reward is a function of the distance between the end effector and the goal. Multi-DEX is competitive to Black-DROPS in non-spare reward scenarios.

Future work

- Multi-DEX with priors from simulator so that it can be used in complex robots with high dimensional state-space.
- Multi-DEX for fast damage recovery in robotics.

Thanks

If you want to know more about this topic, we have a paper that will be presented at Conference on Robot Learning (CoRL) 2018 soon.

Paper: R. Kaushik, K. Chatzilygeroudis, and J.-B. Mouret, "Multi-objective Model-based Policy Search for Data-efficient Learning with Sparse Rewards", *Conference on Robot Learning (CoRL)*, 2018.

Arxiv `https://arxiv.org/pdf/1806.09351.pdf`

Questions or Comments?