Bottom-Up Meta-Policy Search

Luckeciano Melo

Technological Institute of Aeronautics
luckeciano@gmail.com

Abstract. We present Bottom-Up Meta-Policy Search (BUMPS), an algorithm that conceives keyframe optimization as a meta-learning problem in order to improve sample efficiency by using prior experience from few expert policies and exploiting the geometric relationship between subgoals. Results show BUMPS is able to generalize the performance from expert policies to unseen tasks and also improve it by using a fast adaptation strategy.

1 Bottom-Up Meta-Policy Search

Motion optimization is crucial to improve the performance of skills, especially for keyframe-based ones. When we consider the problem of passing-level control, the common solution is to optimize several keyframe models, one for each task (target distance). However, this approach is costly because we do not consider the experience from a previous optimization in the training of another task. To address this problem, we propose Bottom-Up Meta-Policy Search, a meta-learning algorithm whose objective is to improve sample efficiency in optimization procedures by considering prior experience from expert policies in two ways: first, by learning a contextual meta-policy that generalizes to unseen tasks; second, by applying a fast adaptation strategy based on policy sampling.

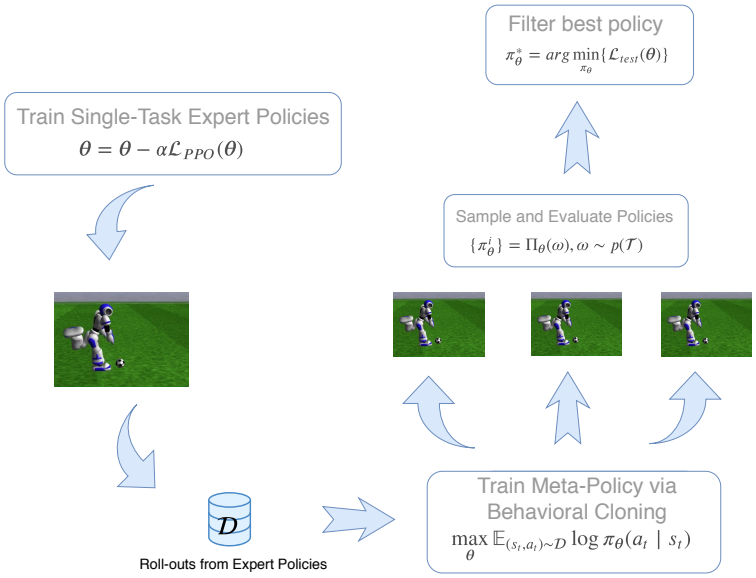

Fig. 1. Bottom-Up Meta-Policy Search.

Fig. 1 describes BUMPS. Firstly, we train single-task expert policies for meta-training tasks. We then collect roll-outs from such policies and use them to train a contextual meta-policy via Behavioral Cloning, without any new interaction between agent and environment. Finally, in order to solve a specific meta-testing task (with context ω), we apply a fast adaptation strategy where we sample contextual policies in the neighborhood of ω, evaluate them and choose the policy which minimizes mean absolute error from target.

As meta-training tasks, we chose the target kick distances from 7.0m to 18.0m, spaced by 0.5m. As meta-testing tasks, we used the same interval but with target kick distances spaced by 0.1m. To train single-task expert policies, we firstly used Imitation Learning to copy kick motion to a neural network and then applied Proximal Policy Optimization to fine-tune towards the task.

Finally, to evaluate BUMPS, we evaluate all the policies in meta-testing tasks by running their motion a hundred times and then computed the statistics presented in Table 1. As we can observe, each part of the algorithm helps to achieve lower error and effectively solve meta-testing tasks.

Table 1. Final results for BUMPS algorithm.

Model Type	**Error (m)**	
	Mean	**Std**
Single-Task Expert Policies	0.57	0.14
Meta-Policy	0.68	0.28
Meta-Policy after Fast Adaptation Strategy	0.39	0.13