Diversity-Inducing Policy Gradient: Using MMD to find a set of policies that are diverse in terms of state-visitation

Muhammad A Masood [1] **Finale Doshi-Velez** [1]

Abstract

Standard reinforcement learning methods aim to master one way of solving a task whereas there may exist multiple near-optimal policies. Being able to identify this *collection* of near-optimal policies can allow a domain expert to efficiently explore the space of reasonable solutions and identify a preferred solution. Unfortunately, existing approaches that encourage and quantify uncertainty over policies focus on measures that are not ultimatly relevant to finding policies whose behaviors are qualitatively distinct. In this work, we define the difference between policies as difference between the states they visit, rather than some internal parameters. We develop a gradient-based optimization for identifying distinct policies and demonstrate that our approach can explore the space of near-optimal policies in multi-goal tasks where existing algorithms fail to do so.

1. Introduction

Standard reinforcement learning methods aim to master one way of solving a task whereas there may exist multiple near-optimal policies that are distinct in some meaningful way. Being able to identify this *collection* of near-optimal policies can allow a domain expert to identify successful policies. For example, a clinician may appreciate knowing there exist comparably-performing policies that do and do not require sedation, or policies that trade between several visits with small procedures or a single visit with a large one. Armed with this information, they can more efficiently explore the space of reasonable treatment policies for one that might be best suited for the task at hand (given other knowledge they may have of the patient).

Unfortunately, existing approaches to find a set of diverse policies involve notions of diversity that are not aligned with the kind of efficient exploration-amongst-reasonable-options settings we described above. Liu et al. (2017) characterize the uncertainty over policies via computing a posterior over policy parameters, but differences in policy parameters may not result in qualitatively different behavior (especially in over-parameterized neural architectures). Haarnoja et al. (2017) encourage diversity via encouraging high entropy distributions over actions given states, which may result in sub-optimal behavior. Fard & Pineau (2011) seek a single non-deterministic policy that may make multiple decisions at any state, which may be overly restrictive if action choices across states must be correlated to achieve near-optimal performance.

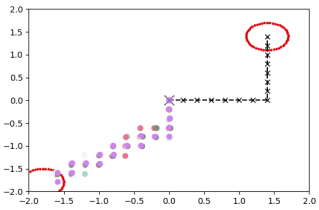

Figure 1. After specifying a path to the nearest goal (black line), we run our algorithm to find a policy that reaches the alternative goal which is further away and yields a slightly lower total reward but requires undertaking a significantly distinct path. From the space of policies that terminate in the alternate goal state, the one our algorithm finds is nearly optimal.

We believe that differences in state visits better capture the kinds of distinct behavior we are seeking, for example, does one prefer a policy that achieves wellness via a set of sedated states, or via the changes inflicted via a set of incremental procedures? We note that the stochasticity in the environment dynamics and perhaps the policy network will result in a distribution of trajectories likely under a given policy. We use the maximum mean discrepancy (MMD) to compare state visitation under different policies. Sriperumbudur et al. (2010) notes that the MMD is an integral probability metric has a closed form solution (unlike Wasserstein and Dudley metric) and exhibits good convergence behavior compared to ϕ-divergences such as KL. Figure 1 shows we can find a policy that is distinct from a specified one.

Below, we describe how policy gradient optimization can be carried out using an MMD-based diversity term. We show that unbiased gradient estimates of the MMD term can be obtained without knowledge of transition dynamics of the environment or the dependence of policy parameters on the MMD witness function.

2. Background

Reinforcement Learning Our reinforcement learning task is formulated as a policy search in a Markov decision process (MDP) defined by a continuous state space $\mathcal{S}$, a (discreete or continuous) action space $\mathcal{A}$, state transition probabilities $p_T(s, a, s') : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ as well as a discount factor γ and a reward function $r(s, a) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$. A policy $\pi(s, a)$ indicates the probability of action a in state s; together with the transition probability $p_T(s, a, s')$, it induces a distribution over state trajectories $\tau = s_0, \dots s_T \sim p_\pi(\tau)$. Traditionally, the task is to find an optimal policy; one that maximizes the long-term expected discounted sum of rewards (return) $g(\tau)$ until a terminal state is reached

$$g(\tau) = \mathbb{E}_{\tau \sim p_\pi(\tau)}\big[\sum_t \gamma^t r(s_t, a_t)\big]$$

In this work, we consider the case where there exist multiple near-optimal policies.

Policy Gradient Policy gradient methods formulate the policy search as an optimization problem where a policy $\pi_\theta(s, a)$ that is parameterized by θ is iteratively updated using gradient ascent in order to find parameters θ^* that maximize the expected return $g(\tau)$. Let $J_{\text{PG}}(\theta) = g(\tau)$. An unbiased estimate of the gradient of the objective function J_{PG} is obtained by Monte Carlo rollouts generated by the policy π_θ using the likelihood ratio trick. For a single rollout $\{s_t, a_t, r_t\}$, the gradient can be estimated as

$$\nabla_\theta J_{\text{PG}}(\theta) \approx \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(s_t, a_t) g_t$$

where $g_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ is the return over the rewards received form time t onwards. Variance reduction techniques such as substituting the reward r_t with the advantage are commonly applied to improve training. Neural networks are commonly used to parameterize the policy π_θ. In this work we augment traditional policy gradient objective function with a regularization that encourages policies that lead to different state visit distribution than previously-identified policies, and thus sequentially identify distinct policies.

Maximum Mean Discrepancy We will use the MMD metric to measure the difference between two different state visit distributions. The MMD is an integral probability metric (Gretton et al., 2007) that measures the difference between two distributions p, q using test functions h from a function space $\mathcal{H}$. The MMD is given by

$$\text{MMD}(p, q, \mathcal{H}) = \sup_{h \in \mathcal{H}} (\mathbb{E}_{x \sim p}[h(x)] - \mathbb{E}_{y \sim q}[h(y)])$$

Computing the MMD is tractable when the function space $\mathcal{H}$ is a unit-ball in a reproducing kernel hilbert space (RKHS) defined by a kernel $k(\cdot, \cdot)$ and is given by

$$\text{MMD}^2(p, q, \mathcal{H}) = \mathbb{E}[k(x, x')] - 2\mathbb{E}[k(x, y)] + \mathbb{E}[k(y, y')]$$

where x, x' i.i.d. $\sim p$ and y, y' i.i.d. $\sim q$. The expectation terms in the analytical expression for the MMD can be approximated using samples.

In Section 3, we show that when the distribution p corresponds to the state visit distribution under a policy π_θ and q corresponds to the state visit distribution under a previous policy, computing gradients with respect to the policy parameters θ is tractable. We use this gradient information to find a diverse set of policies for an MDP.

3. Diversity-Inducing Policy Gradient: DIPG

Our algorithm constructs a set of diverse policies for an MDP by iteratively finding policies that are diverse relative to an existing set of policies. First, we formulate a diversity inducing objective function that regularizes the typical policy gradient objective. Then, we show that optimizing this objective is tractable using the familiar log-derivative trick. Finally, we explain how to iteratively apply this diversity-inducing policy gradient algorithm to find a set of distinct policies that solve an MDP.

3.1. DIPG objective

We propose adding a regularization term that encourages a policy that leads to a distribution over trajectories $p_\theta(\tau)$ that are distinct from a given set of distributions over trajectories $\mathcal{Q} = \{q_m(\tau)\}_{m=1}^M$. Our diversity measure $D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q})$ is the squared MMD between the distribution of state trajectories under the current policy π_θ and the distribution $q_*(\tau)$ in $\mathcal{Q}$ that is most similar to it.

$$\begin{aligned} D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q}) &= \min_{m \in \{1,\dots,M\}} \text{MMD}^2(p_\theta(\tau), q_m(\tau)) \\ &= \text{MMD}^2(p_\theta(\tau), q_*(\tau)) \end{aligned}$$

The MMD-based diversity-inducing objective function is given by $J_{\text{MMD}}(\theta)$.

$$J_{\text{MMD}}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[R(\tau)] + \alpha D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q})$$

The first term is the expected return and the second term is proportional to the diversity measure between a policy distribution and the set of specified distributions $\mathcal{Q}$. The parameter α decides the relative importance of optimality and diversity of the policy.

In order to optimize the MMD-based diversity-inducing objective, we need to specify how gradients of the diversity term can be computed with respect to the policy parameters θ.

3.2. Optimization via gradient ascent

In order to use gradient ascent on $J_{\text{MMD}}(\theta)$, what remains to specify is the gradient with respect to θ of the diversity inducing term. Let $q_*(\tau)$ be the distribution in $\mathcal{Q}$ that minimizes the MMD between the state trajectory distribution of the policy π_θ and $q_m \in \mathcal{Q}$. Then, the gradient with respect to the policy parameters θ of the diversity term is given by

$$\begin{aligned}\nabla_\theta D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q}) &= \nabla_\theta \text{MMD}^2(p_\theta(\tau), q_*(\tau), \mathcal{H}) \\ &= \mathbb{E}[k(\tau_p, \tau_p')\nabla_\theta \log(p_\theta(\tau_p)p_\theta(\tau_p')] \\ &\quad - 2\mathbb{E}[k(\tau_p, \tau_q)\nabla_\theta \log(p_\theta(\tau_p)q_*(\tau_q)] \\ &\quad + \mathbb{E}[k(\tau_q, \tau_q')\nabla_\theta \log(q_*(\tau_q)q_*(\tau_q')] \quad \to 0\end{aligned}$$

where τ_p, τ_p' i.i.d. $\sim p_\theta(\tau)$ and τ_q, τ_q' i.i.d. $\sim q_*(\tau)$. The last term only involves the distribution $q_*(\tau) \in \mathcal{Q}$ that has no dependence on θ. The gradient term can be estimated by linear combinations of the $\nabla_\theta \log p_\theta(\tau)$ involving the kernel between sample trajectories from the policy π_θ and with a set of specified trajectories from $\mathcal{Q}$. It is well-known (Sutton et al., 2000) that the gradient of the score function of the trajectory distribution $\nabla_\theta \log p_\theta(\tau_p)$ does not require the dynamics model and can be expressed in terms of the score function of the policy network ($\nabla_\theta \log p_\theta(\tau) = \sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(a_t|s_t)$).

We now have all the machinery in place to augment any existing policy gradient method with a diversity inducing term. We will specify the basic algorithm for finding a policy that is diverse with respect to some specified state distributions and introduce an algorithm that leverages this to find a set of diverse policies.

Algorithm 1 MMD-based Diversity-Inducing Policy

Input: Known policies $\mathcal{P}_{\text{known}}$, MDP $\{\mathcal{S}, \mathcal{A}, p_s, r\}$, learning rate η
Initialize policy parameters θ
$\mathcal{Q} \leftarrow$ State distributions from following policies in $\mathcal{P}_{\text{known}}$
repeat
  Generate an episode $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$, following π_θ
  for Each step $t = 0, \ldots, T-1$ **do**
    1: Estimate $\nabla_\theta J_{\text{PG}}$ and $\nabla_\theta D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q})$
    3: Update policy parameters via gradient ascent
    $\theta \leftarrow \theta + \eta(\nabla_\theta J_{\text{PG}} + \alpha \nabla_\theta D_{\text{MMD}}(p_\theta(\tau), \mathcal{Q}))$
  end for
until convergence
Output: policy p_θ

Algorithm 2 DIPG

Input: Number of policies N, MDP $\{\mathcal{S}, \mathcal{A}, p_s, r\}$, learning rate η.
Collection of known policies $\mathcal{P}_{\text{known}} = \emptyset$
for $n = 1$ **to** N **do**
  1: Find a policy p_n that is distinct from the current set of known policies $\mathcal{P}_{\text{known}}$:
  $p_n \leftarrow$ Algorithm 1$(\mathcal{P}_{\text{known}}, \text{MDP}, \eta)$
  2: Add p_n to the set of known policies $\mathcal{P}_{\text{known}}$:
  $\mathcal{P}_{\text{known}} \leftarrow \mathcal{P}_{\text{known}} \cup p_n$
end for
Output: Set of policies $\mathcal{P}_{\text{known}}$

3.3. Finding multiple diverse policies

We begin by modifying vanilla policy gradient with the diversity inducing term (Algorithm 1) in order to get policies that are diverse with respect to a specified set. We then prescribe a method for iteratively finding a set of policies that are diverse with respect to itself (Algorithm 2).

4. Experimental Setup

Environments We consider multi-goal 2D gridworld domains where we vary the reward structure.

The multi-goal responsive reward (MG-RR) environment is the same as the one used in (Haarnoja et al., 2017) where the environment responds positively whenever the agent moves closer to one of the goals. In the multi-goal terminal reward (MG-TR) environment, we vary the reward structure such that a positive response is obtained only upon reaching one of the terminal goal regions. This type of reward structure makes the problem more challenging and realistic e.g. if you are looking for a coffee shop near you, you will not receive a positive reward until you reach the coffee shop and get your drink. Finding multiple goals in the MG-TR environment will require enhanced exploration from the agent as compared to MG-RR.

Algorithm and Baselines We compare our approach to the following other algorithms that can find multiple distinct policies.

Stein Variational Policy Gradient [**SVPG**]: Liu et al. (2017) use functional gradient descent to compute a point-based posterior distribution over policy parameter space. We use $n = 16$ agents for our point-based estimates as was done in Liu et al. (2017).

Deep Energy-Based Policy [**Soft-Q**]: Haarnoja et al. (2017) show how to find a single policy, encouraged to have high entropy on the probability of the action given state (and thus providing many options at any state).

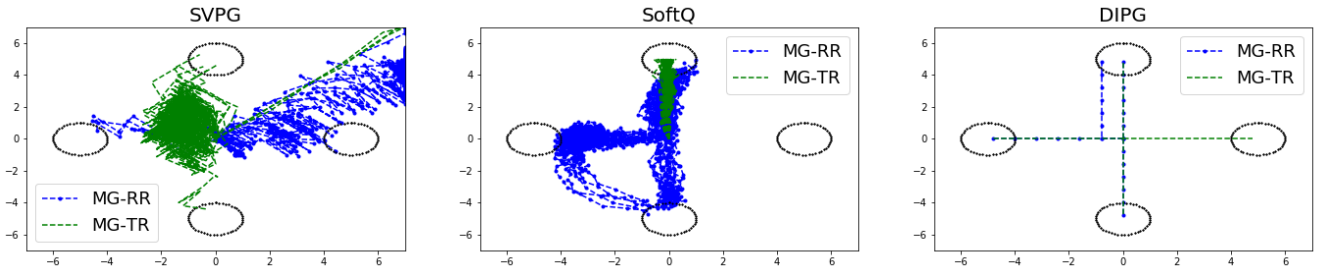

Figure 2. In the above plots, we show the policies found by SVPG (left), Soft-Q (middle) and our algorithm (right) for both variants of the multi-goal environment. MG-RR is in blue and it is an easier environment to solve than MG-TR (green). In both the MG-RR and MG-TR environments, SVPG with 16 agents fails to learn a useful policy although there are instances where it reaches the goal region (which could be due to chance). Soft-Q finds 3 of the 4 goal regions in MG-RR, however, due to the high-entropy in the action space the trajectories are sub-optimal (e.g. this policy learns to move to [0,-5] when it's near [-5,0]. In both environments running DIPG (N = 4) to find 4 policies successfully finds either all four goals or at least three. The policy follows either an optimal or nearly optimal route to the goal.

Table 1. A quantitative comparison of the performance of multi-policy algorithms in the MG-RR and MG-TR environments. Average return obtained from simulations after training is presented along with (in parenthesis), the average number of different goals regions visited by the policies found

	SVPG	Soft-Q	DIPG
MG-RR	-3265 (1)	-190 (2.6)	-347 (3)
MG-TR	-2286 (0.8)	-1971 (0.8)	811 (3.2)

5. Results

In the figure 2 we show the performance of SVPG, Soft-Q learning and our algorithm DIPG (with N = 4, Gaussian RBF kernel for the MMD with unit bandwidth and regularization strength $\alpha = 0.2$) on the two variations of the multi-goal environment. In our experiments, we set the maximum length of an episode to be 100 and trained Soft-Q and DIPG for 100 episodes. We trained SVPG for 10,000 updates which corresponds to a minimum of 100 episodes. We repeat the experiment 5 times and present the average test-time return and multi-state visitation in Table 1. In both environments DIPG has the highest average number of different goal regions visited. We note that DIPG does not have the highest return in the MG-RR environment. This is most likely due to our discretization of the action space leading to a fixed step-size in one of four cardinal directions. The environment penalizes based on the norm of the action-step, something that we did not tune our problem to.

Overall, our algorithm successfully finds multiple goals in an optimal or near-optimal manner whereas baseline approaches are either unable to reach multiple goals or do so in particularly sub-optimal manners. The poor performance of SVPG could be due to the difficulties of performing functional gradient descent over a high-dimensional parameter space. Even if those difficulties were to be overcome, the search for diversity in the space of neural network parameters does not correspond directly to any meaningful notions of diversity to us. While the Soft-Q algorithm exhibits diversity in the policies it finds, the entropy regularization in the Soft-Q algorithm gives rise to undesirable sub-optimal behavior of the policy.

6. Discussion and Conclusion

We presented an approach for identifying a collection of near-optimal policies with significantly different distributions of state visits. Being able to identify these diverse options may be useful when the agent does not have complete information about the task, and presenting a set of potentially reasonable options could help a domain expert identify the one that might work best given additional constraints unknown to the agent. In settings where there exist easy-to-use but low fidelity simulators for the task of interest, efficient algorithms for identifying multiple candidate policies from the simulator may help guide sample-efficient exploration for a near-optimal policy in the actual task.

More broadly, our approach should extend easily to other measures of distinctness, such as the joint distribution over states and actions, or the conditional distribution over actions given state. There are also opportunities for incorporating more efficient search algorithms than gradient descent (e.g. (Toussaint & Lopes, 2017)).

References

Fard, Mahdi Milani and Pineau, Joelle. Non-deterministic policies in markovian decision processes. *J. Artif. Intell. Res.(JAIR)*, 40:1–24, 2011.

Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte, Schölkopf, Bernhard, and Smola, Alex J. A kernel

method for the two-sample-problem. In *Advances in neural information processing systems*, pp. 513–520, 2007.

Haarnoja, Tuomas, Tang, Haoran, Abbeel, Pieter, and Levine, Sergey. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

Liu, Yang, Ramachandran, Prajit, Liu, Qiang, and Peng, Jian. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

Sriperumbudur, Bharath K, Fukumizu, Kenji, Gretton, Arthur, Schölkopf, Bernhard, and Lanckriet, Gert RG. Non-parametric estimation of integral probability metrics. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pp. 1428–1432. IEEE, 2010.

Sutton, Richard S, McAllester, David A, Singh, Satinder P, and Mansour, Yishay. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Toussaint, Marc and Lopes, Manuel. Multi-bound tree search for logic-geometric programming in cooperative manipulation domains. In *(ICRA 2017)*, 2017.