REINFORCEMENT LEARNING WITH PROBABILISTICALLY COMPLETE EXPLORATION

Philippe Morere*
University of Sydney,

Gilad Francis*
University of Sydney,

Tom Blau*
University of Sydney,

Fabio Ramos
University of Sydney, Nvidia

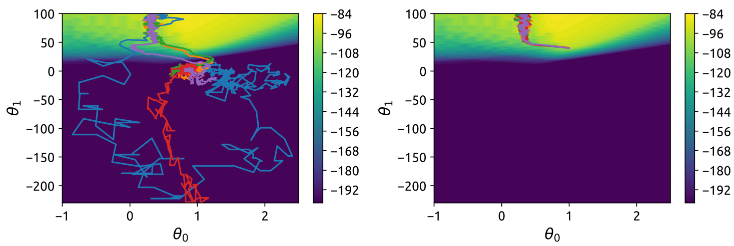
Abstract

Balancing exploration and exploitation remains a key challenge in reinforcement learning (RL). State-of-the-art RL algorithms suffer from high sample complexity, particularly in the sparse reward case, where they can do no better than to explore in all directions until the first positive rewards are found. To mitigate this, we propose Rapidly Randomly-exploring Reinforcement Learning (R3L). We formulate exploration as a search problem and leverage widely-used planning algorithms such as Rapidly-exploring Random Tree (RRT) to find initial solutions. These solutions are used as demonstrations to initialize a policy, then refined by a generic RL algorithm, leading to faster and more stable convergence. We provide theoretical guarantees of R3L exploration finding successful solutions, as well as bounds for its sampling complexity. We experimentally demonstrate the method outperforms classic and intrinsic exploration techniques, requiring only a fraction of exploration samples and achieving better asymptotic performance.

1 Introduction

Reinforcement Learning (RL) studies how agents can learn a desired behaviour by simply using interactions with an environment and a reinforcement signal. Central to RL is the long-standing problem of balancing exploration and exploitation. Agents must first sufficiently explore the environment to identify high-reward behaviours, before this knowledge can be exploited and refined to maximize long-term rewards. Many recent RL successes have been obtained by relying on well-formed reward signals, that provide rich gradient information to guide policy learning. However, designing such informative rewards is challenging, and rewards are often highly specific to the particular task being solved. Sparse rewards, which carry little or no information besides binary success or failure, are much easier to design. This simplicity comes at a cost; most rewards are identical, so that there is little gradient information to guide policy learning. In this setting, the sample complexity of simple exploration strategies was shown to grow exponentially with state dimension in some cases (Osband et al., 2016b). Intuition behind this phenomenon can be gained by inspecting Figure 1a: exploration in regions where the return surface is flat leads to a random walk type search. This inefficient search continues until non-zero gradients are found, which can then be followed to a local optimum. Planning algorithms can achieve much better exploration performance than random walk by taking search history into account (Lavalle, 1998). These techniques are also often guaranteed to find a solution in finite time if one exists (Karaman Frazzoli, 2011). In order to leverage the advantages of these methods, we formulate RL exploration as a planning problem in the state space. Solutions found by search algorithms are then used as demonstrations for RL algorithms, initializing them in regions of policy parameter space where the return surface is not flat. Figure 1b shows the importance of such good initialization; surface gradients can be followed, which greatly facilitates learning. This paper brings the following contributions. We first formulate RL exploration as a planning problem. This yields a simple and effective method for automatically generating demonstrations without the need for an external expert, solving the planning problem by adapting the classic

Figure 1: Expected returns achieved by linear policy with 2 parameters on Sparse MountainCar domain (background). Gradient is 0 in the dark blue area. Trajectories show the evolution of policy



(a) θ_0 (b) θ_0

parameters over 1000 iterations of TRPO, with 5 random seeds. Same colors indicate the same random seeds. 1a Random-walk type behaviour observed when parameters are initialized using Glorot initialization (Glorot & Bengio, 2010). 1b Convergence observed when parameters are initialized in a region with gradients (1, 40). Rapidly-exploring Random Tree algorithm (RRT) (Kuffner LaValle, 2000). The demonstrations are then used to initialize an RL policy, which can be refined with a classic RL method such as TRPO (Schulman et al., 2015). We call the proposed method Rapidly Randomly-exploring Reinforcement Learning (R3L)1 , provide theoretical guarantees for finding successful solutions and derive bounds for its sampling complexity. Experimentally, we demonstrate R3L improves exploration and outperforms classic and recent exploration techniques, and requires only a fraction of the samples while achieving better asymptotic performance. Lastly, we show that R3L lowers the variance of policy gradient methods such as TRPO, and verify that initializing policies in regions with rich gradient information makes them less sensitive to initial conditions and random seed. The paper is structured as follows: Section 2 analyzes the limitations of classic RL exploration. Section 3 describes R3L and provides theoretical exploration guarantees. Related work is discussed in Section 4, followed by experimental results and comments in Section 5. Finally, Section 6 concludes and gives directions for future work.

2 SPARSE-REWARD RL AS RANDOM WALK

Many recent RL methods are based on a policy gradient optimization scheme. This approach optimizes policy parameters θ with gradient descent, using a loss function $\mathcal{L}(\theta)$ (e.g. expected return) and gradient $g(\theta) \equiv \nabla_\theta Z(\theta)$,. Since computing $\mathcal{L}(\theta)$ exactly is intractable, it is common to use unbiased empirical estimators $\hat{L}(\theta)$ and $\hat{g}(\theta)$, estimated from samples acquired by executing the policy. Optimization of then follows the common *stochastic gradient descent* t (SGD) update-rule (Bottou, 2010; Robbins & Monro, 1951): $\theta_{n+1} = \theta_n - \epsilon \hat{g}\,(\theta_n)$ where E is the learning rate

The SGD update rule defines a discrete-time stochastic process (Mandt et al., 2017). Note that $\hat{g}$ is the mean of $n_m b$ b i.i.d. samples. Following the central limit theorem, the distribution over $\hat{g}$ is approximately

$$\hat{g}(\theta) \sim \mathcal{N}(g(\theta), \frac{1}{n_{ma}} C(\theta))$$

meaning $\hat{g}$ is an unbiased estimator of g with covariance $\frac{1}{n_{ma}} C(\theta))$ Consequently, the update rule can be rewritten as (Mandt et al., 2017):

$$\theta_{n+1} = \theta_n - \epsilon g(\theta_n) + \frac{\epsilon}{n_{mb}} B \Delta W, \qquad \Delta W \sim \mathcal{N}(0, \bar{1}).$$

Here we assume that $C(\theta) = C$, i.e. approximately constant w.r.t. θ. and factorizes as $C = BB^T$.
SGD is efficient for high-dimensional problems as it offers almost dimension independent convergence rates (Nesterov, 2018). However, SGD requires non-zero gradients to guide the search towards the optimum θ^* , i.e. $|g(\theta)| > \epsilon_\alpha, \forall \theta \neq \theta^* \; \epsilon_{\mathcal{I}} \in R$ In the case of sparse-reward RL problems, such as in Figure l, much of the loss surfaceis flat. This leadsto inefficient exploration of parameter space $\boldsymbol{\Theta}(0)$ as the drift component in Eq. (1) $g \approx 0$. turning the SGD to a random walk in Θ: $\Delta \theta = \frac{e}{n_{mb}} B \Delta W$ Random walk is guaranteed to wander to infinity when dimensionality $d\sigma \geq 3$ (Polya, 1921; Kakutani, l944). However, the probability of it reaching a desired region in $\boldsymbol{\Lambda}(\alpha)$ e.g. where $a \neq 0$, depends

Table1: impact of learning local policy π_l and biasing search towards $\mathcal{F}_{goal}$ with probability p_g on R3L exploration. Results show the mean and standard deviation of successful trajectory length $|\tau|$ and number of timesteps required., computed over 2O runs

		Goal bias ($p_g = 0.05$)		Unbiased ($p_g = 0$)	
		Learned π_l	Random π_l	Learned π_l	Random π_l
MountainCar	$\|\tau\|$	**84.75 ± 5.47**	131.90 ± 17.91	86.75 ± 11.82	139.85 ± 18.79
	timesteps	**895.65 ± 190.70**	4303.80 ± 681.60	928.90 ± 204.0	4447.55 ± 417.10
Pendulum	$\|\tau\|$	**73.10 ± 12.86**	75.35 ± 14.50	67.05 ± 15.30	77.90 ± 12.62
	timesteps	**1108.65 ± 155.29**	2171.95 ± 381.20	1221.35 ± 216.14	2349.20 ± 249.26
Acrobot	$\|\tau\|$	177.55 ± 20.66	**163.95 ± 19.19**	173.5 ± 24.22	169.05 ± 17.07
	timesteps	15422.00 ± 2624.16	**12675.20 ± 2652.39**	15792.65 ± 3182.77	13133.55 ± 2060.51
Cartpole Swingup	$\|\tau\|$	**217.35 ± 53.09**	319.20 ± 58.78	235.70 ± 70.06	348.35 ± 80.09
	timesteps	**17502.75 ± 13923.82**	27186.70 ± 12246.32	23456.25 ± 16792.11	34482.20 ± 12034.27
Reacher	$\|\tau\|$	32.70 ± 13.55	**22.05 ± 8.81**	32.25 ± 12.05	25.30 ± 11.02
	timesteps	1445.80 ± 1314.48	**838.85 ± 846.94**	1423.00 ± 1030.07	1034.55 ± 1208.67

Accelerating RL by learning from demonstration was investigated in Niekum et al. (2015); Bojarski et al. (2016); Torabi et al. (2018). However, these techniques rely on user-generated demonstrations or a-priori knowledge of environment parameters. In contrast, R3L automatically generates demonstrations, with no need of an external expert.

5 EXPERIMENTS

In this section, we investigate (i) how learning a local policy π_l and biasing search towards F_{goal} with probability pg affects R3L exploration, (ii) whether separating exploration from policy refinement is a viable and robust methodology in RL, (iii) whether R3L reduces the number of exploration samples needed to find good policies, compared with methods using classic and intrinsic exploration, and (iv) how R3L exploration can reduce the variance associated with policy gradient methods. All experiments make use of the Garage (Duan et al., 2016) and Gym (Brockman et al., 2016) frameworks. The experimental setup features the following tasks with sparse rewards: Cartpole SWingup ($S \subseteq R^4, \mathcal{A} \subseteq R$) MountainCar ($S \subseteq R^2, \mathcal{A} \subseteq R$) Acrobot ($S \subseteq R^4, \mathcal{A} \subseteq R$) Pendulun (S GR,A C R) Reacherc ($S \subseteq R^6, \mathcal{A} \subseteq R^2$) Fetch Reach c $S \subseteq R^{13}, \mathcal{A} \subseteq R^4$. , and Hand Reacl ($S \subseteq R^{78}, A \subseteq R^{20}$) The exact environment and reward definitions are described in Appendix S3 **R3L exploration analysis** We first analyze the exploration performance of R3L in a limited set of RL environments, to determine the impact that learning policy l has on exploration speed. We also investigate whether R3L exploration is viable in environments where no goal information is available. Table 1 shows the results of this analysis. Learning l seems to greatly decrease the number of exploration timesteps needed on most environments. However, it significantly increases the number of timesteps on the acrobot and reacher environments. Results also suggest that learning l helps R3L to find shorter trajectories on the same environments, which is a desirable property in many RL problems. Biasing R3L exploration towards the goal set Fgoal helps finding successful trajectories faster, as well as reducing their length. However, R3L exploration without goal bias is still viable in all cases. Although goal information is not given in the classic MDP framework, it is often available in real-world problems and can be easily utilized by R3L. Lastly, successful trajectory lengths have low variance, which suggests R3L finds consistent solutions. Comparison to classic and intrinsic exploration on RL benchmarks We examine the rates at which R3L learns to solve several RL benchmarks, and compare them with state-of-the-art RL algorithms. Performance is measured in terms of undiscounted returns and aggregated over 10 random seeds, sampled at random for each environment. We focus on domains with sparse rewards, which are notoriously difficult to explore for traditional RL methods. Our experiments focus on the widely-used methods TRPO (Schulman et al., 2015) and DDPG (Lillicrap et al., 2015). R3L-TRPO and R3L-DDPG are compared to the baseline algorithms with Gaussian action noise. As an additional baseline we include VIME-TRPO (Houthooft et al., 2016). VIME is an exploration strategy based on maximizing information gain about the agent's belief of the environment dynamics. It is included to show that R3L can improve on state-of-the-art exploration methods as well as naive ones, even though the return surface for VIME-TRPO is no longer flat, unlike Figure 1. The exact experimental setup is described in Appendix S3.2. The R3L exploration phase is first run to generate training trajectories for all environments. The number of environment interactions during this phase is accounted for in the results, displayed as an offset with a vertical dashed black line. The average performance achieved 7 outperforms classic and intrinsic exploration techniques, requiring only a fraction of exploration samples and achieving better asymptotic performance. As future work, R3L could be extended to real-world problems by leveraging recent advances on bridging the gap between simulation and reality (Peng et al., 2018). Respecting Assumption 2, a policy would first be trained on a simulator and then transferred to the real-world. Exploration in high-dimensional tasks is also challenging as stated in Theorem 3 and confirmed experimentally by increased R3L exploration timesteps. Exploiting external prior knowledge and/or the structure of the problem can benefit exploration in high-dimensional tasks, and help make R3L practical for problems such as Atari games. Lastly, recent advances in RRT (Chiang et al., 2019) and learning from demonstration (Torabi et al., 2018) could also improve R3L.

References

R. Arratia and L. Gordon. Tutorial on large deviations for the binomial distribution. Bulletin of Mathematical Biology, 1989.

M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In Advances in Neural Information Processing Systems, 2016.

M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Ziebaand. End to end learning for self-driving cars. NIPS Deep Learning Symposium, 2016. L.N. Bottou. Large-scale machine learning with stochastic gradient descent. In International

Conference on Computational Statistics, 2010.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym, 2016.

B. Chen, B. Dai, and L. Song. Learning to plan via neural exploration-exploitation trees. arXiv:1903.00070, 2019.

H. Lewis Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust. RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies. Robotics and Automation Letters, 4, 2019.

Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In International Conference on Machine Learning, 2016.

A. Dvoretzky and P. Erdos. Some problems on random walk in space. In ˝ Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1951.

A. Ecoffet, J Huizinga, J Lehman, K. O. Stanley, and J. Clune. Go-explore: a new approach for hard-exploration problems. arXiv:1901.10995, 2019.

A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson. PRM-RL: Longrange robotic navigation tasks by combining reinforcement learning and sampling-based planning. In International Conference on Robotics and Automation, 2018.

C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel. Reverse curriculum generation for reinforcement learning. In Conference on Robot Learning, 2017.

L. Fox, L. Choshen, and Y. Loewenstein. DORA the explorer: Directed outreaching reinforcement action-selection. In International Conference on Learning Representations, 2018.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In International Conference on Artificial Intelligence and Statistics, 2010.

R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. In Advances in Neural Information Processing Systems, 2016.

D. Hsu, J.C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In International Conference on Robotics and Automation, 1997.

D. Hsu, R. Kindel, J.C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. The International Journal of Robotics Research, 2002.

S. Kakutani. On brownian motions in n-space. Proceedings of the Imperial Academy, 1944.

S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 2011.

L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Transactions on Robotics and Automation, 1996.

M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin. Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation. Robotics and Automation Letters, 2019.

J. J. Kuffner and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. In International Conference on Robotics and Automation, 2000.

S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Department of Computer Science. Iowa State University., 1998.

S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. The International Journal of Robotics Research, 2001.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, et al. Continuous control with deep reinforcement learning. arXiv:1509.02971, 2015.

M. Lopes, T. Lang, M. Toussaint, and P.Y. Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In Advances in Neural Information Processing Systems, 2012.

S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate bayesian inference. The Journal of Machine Learning Research, 18, 2017.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 2015.

P. Morere and F. Ramos. Bayesian RL for goal-only rewards. In Conference on Robot Learning, 2018.

A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In International Conference on Robotics and Automation, 2018.

Y. Nesterov. Lectures on Convex Optimization. Springer, 2018.

S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, et al. Learning grounded finite-state representations from unstructured demonstrations. The International Journal of Robotics Research, 34, 2015.

I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped DQN. In Advances in Neural Information Processing Systems, 2016a.

I. Osband, B. Van Roy, and Z. Wen. Generalization and exploration via randomized value functions. In International Conference on Machine Learning, 2016b.

P. Y. Oudeyer and F. Kaplan. How can we define intrinsic motivation? In International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 2008.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In International Conference on Machine Learning, 2017.

X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In International Conference on Robotics and Automation, 2018.

M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz. Parameter space noise for exploration. In International Conference on Learning Representations, 2018.

G. Pólya. Über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im straßennetz. Mathematische Annalen, 1921.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems, 2008.

H. Robbins and S. Monro. A stochastic approximation method. The annals of mathematical statistics, pp. 400–407, 1951.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In International Conference on Machine Learning, 2015.

B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. arXiv:1507.00814, 2015.

I. Szita and A. Lorincz. The many faces of optimism: a unifying approach. In ˝ International Conference on Machine learning, 2008.

F. Torabi, G. Warnell, and P. Stone. Behavioral cloning from observation. In International Joint Conference on Artificial Intelligence, 2018.

C. Urmson and R. Simmons. Approaches for heuristically biasing RRT growth. In International Conference on Intelligent Robots and Systems, 2003.

N. A. Vien, H. Zimmermann, and M. Toussaint. Bayesian functional optimization. In AAAI Conference on Artificial Intelligence, 2018.

A. Wilson, A. Fern, and P. Tadepalli. Using trajectory data to improve bayesian optimization for reinforcement learning. The Journal of Machine Learning Research, 2014.