Variance Reduction for Reinforcement Learning in Input-Driven Environments

Stefan Werner

***Abstract*– Reducing variance of estimated policy gradients remains among the key challenges to unlocking the power behind the popular family of policy gradient methods (PGMs). In their work, Mao et al. [2] demonstrate that standard state-value baselines are insufficient when agents are confronted with input-driven environments whose dynamics are partially determined by exogenous input processes. Accordingly, they derive an explicit notion of an optimal input-dependent baseline for input-driven Markov Decision Processes (MDPs) and further propose two approaches for their heuristic estimation. Extensive experimental results verify significant improvements over standard state-value baselines for a broad field of study including networking or locomotion related tasks.**

1 Introduction

In recent years, deep reinforcement learning (DRL) has been effectively applied to various real-world applications, while outperforming prior state-of-the-art approaches in numerous fields. As of now, PGMs remain among the most effective and thereby important RL algorithms. Reducing the variance of estimated policy gradients, however, continues to prevail among the key challenges to unlocking the power behind PGMs. Subtracting a state-dependent value function from policy gradient estimates is a commonly applied variance reduction technique, which Mao et al. demonstrate to be ineffective in input-driven environments. Here, rewards along with internal state-transitions are not merely the product of the policy's taken actions but also affected by an exogenous stochastic process – the *input process*. Related applications include traffic control, queuing systems, robotic control with disturbances, resource allocation and job scheduling tasks. As originally demonstrated by Mao et al., this seminar work particularly demonstrates their proposed variance reduction techniques within the scope of job scheduling, i.e. distributing incoming requests of varying size via a load balancing frontend (dispatcher) among several servers.

Consider, for instance, optimizing the dispatcher's scheduling policy regarding the average job completion time of incoming requests for the setting illustrated in Figure 1. We may model the former task as a discrete MDP whose observed state representation s_t comprises information on each server's load as well as the size of an incoming job at each timestep. It is apparent that the environment's dynamics are not completely dictated at hand of its previous state and the agent's action, but also depend on the rate of incoming jobs along with their respective job size, thus are subject to the exogenous input process's behavior. Reinforcing an agent's valuable actions properly is particularly challenging in Mao et al.'s notion of an *input-driven environment* if the input-behavior is not specifically accounted for. Again, refer to the scheduling task depicted in Figure 1 (left). Assigning the incoming job (green) to the network's shortest queue is clearly optimal in the sense of the original objective, i.e. improving average job completion time [5]. If, by chance, a significant increase of large-sized requests ensues our scheduling decision (right) the average job completion time degrades and respective returns diminish. In contrast, suboptimal scheduling decisions, such as assigning an incoming job to the network's largest queue, may improve upon the former action's received rewards under more favorable input sequences. The stochas-

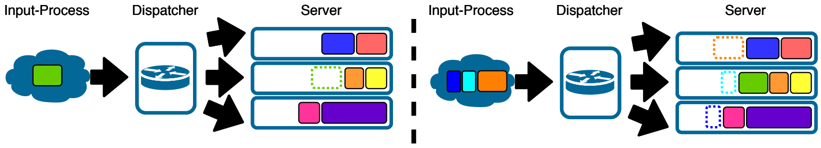

Figure 1: Load balancing of incoming jobs for two sequential timesteps via a dispatcher.

tic nature of the exogenous input process therefore renders an appropriate reward attribution to the policy's taken actions as challenging, hence causing further variance in its reward signal which ultimately prohibits sample efficient learning. In their experiments, Mao et al. indeed find policy gradient estimates of high variance to prohibit the popular PGM A2C from learning former optimal policy when job arrival adheres to an exogenous Poisson process. They further argue that former observation is not the exception but the rule, as state-dependent baselines completely disregard the effects of varying input-behavior.

2 Input-Dependent Baselines

In order to properly account for the proposed setting's problem characteristics, Mao et al. augment the definition of a MDP by an exogenous input process $\mathcal{Z}$ that emits multidimensional (k-dimensional) input values $z_t \in \mathbb{R}^k$ (e.g. job size, priority, ...) at each timestep t. In this context, state transition probabilities $\mathbb{P}_s(s_{t+1}|s_t, a_t, z_t)$ are not only determined by the environment's current state s_t and taken action a_t, but also partially affected by the current input z_t. In contrast, the input process's dynamics $\mathbb{P}_z(z_{t+1}|z_{0:t})$ are determined by the entire history of emitted inputs, hence the entire input sequence $z_{0:t}$ until timestep t. The authors identify and subsequently address two fundamentally different input-driven environments. First, environments where an agent observes the emitted input z_t at each timestep t and $\mathcal{Z}$ is Markov, as in the prior load balancing example ($\mathcal{Z}$ is a Poisson process and thus Markov). Here, the Markov property assures the future's independence from its past given the present. Hence, we may assess by what means the observed input z_t influences our believe on the state transition probabilities P_s and how to select an agent's action a_t separate from prior inputs $z_{0:t-1}$. Mao et al. further show that an input-driven MDP under the former assumptions corresponds to a fully-observable MDP with the augmented state representation $w_t = (s_t, z_t)$.

Second, settings where an exogenous stochastic (not necessarily Markov) process influences the environment's state-dynamics, but no input behavior is observed. Here, the agent is not provided any additional information concerning the current input z_t and therefore selects an action a_t oblivious to it. For instance, Mao et al. relate former assumptions of the *input-driven MDP* to an example where the agent is tasked to navigate over several floating tiles subject to different damping and friction properties determined by the input process [1]. Mao et al. prove that this setting relates to a partially-observable MDP (POMDP) where the (augmented) state w_t is given by $(s_t, z_{0:t})$, while $z_{0:t}$ remains unknown and solely s_t is observed. In general, this case is significantly more challenging since agents do not directly observe input-behavior but rather their implications on state transitions. Also, input processes are defined more broadly and not assumed to be Markov. Hence, an agent must account for all prior inputs $z_{0:t}$ as opposed to only z_t. The assumption of an input process that is independent from previously taken actions is, however, critical to both examined cases.

PGMs generally build upon the policy gradient theorem [4] which suggests reinforcing actions that favor high future rewards, for instance, by adjusting a policy network's weights according to received Monte Carlo returns as an estimate of respective state-values. Here, respective policy gradient estimates are subject to high variance, as Monte Carlo returns at timestep t sum all future rewards $r_t, \dots, r_T$ until termination and hence depend on numerous stochastic variables. Subtracting a state-dependent baseline $b(s_t)$ (e.g. $V^\pi(s_t)$) compensates for the reward signal's dependence on experienced states, as taken actions are now assessed based on whether they were advantageous. Intuitively, baselines reduce variance by relating rewards to whether they should be attributed to selected actions or were due to the agent being in a favorable states.

In input-driven environments, however, standard state-dependent baselines turn unreliable. In fact, Mao et al. show that their estimate's variance under $b(s_t)$ is unbounded for input-driven MDPs. Accordingly, they introduce input-dependent baselines $b(w_t, z_{t:\infty})$ which adjust state-value estimates for w_t by future inputs $z_{t:\infty}$ to explicitly account for their implications on subsequent dynamics and rewards. Updating the policy-network's parameterization subject to former input-dependent baseline $b(w_t, z_{t:\infty})$ hence relies on the trajectory's future inputs $z_{t+1:\infty}$. These must either be provided beforehand, as is the case when using simulators and recorded input traces, or be logged during training.

Since we assume an agent's taken actions not to affect the exogenous input process's dynamics, Mao et al. initially prove that given an observation w_t, the input sequence $z_{t:\infty}$ and action a_t are conditionally independent, thus form a Markov chain. On this basis, they formulate an adjusted policy gradient theorem in the scope of input-driven MDPs and further derive that, similar to standard state-value baselines for MDPs, input-dependent baselines are bias-free. Mao et al. also contribute a theoretical notion of an optimal input-dependent baselines $b^*(w, z)$ in the sense of minimizing the covariance matrix's trace for estimated policy gradients of model-free PGMs. While computing the former optimal input-dependent baseline b^* generally demands convoluted estimations, Mao et al. opt for a less sophisticated alternative that approximates b^* in expectation, similar to the way that $V^\pi(s_t)$ is commonly applied as a practical but suboptimal baseline for $b(s_t)$. In detail, they suggest $V^\pi(w_t, z_{t:\infty})$ which corrects the expected return starting from s_t by a trajectory's future input sequence $z_{t:\infty}$.

3 Heuristic Estimation

Mao et al. specifically note that their proposed input-dependent baseline $V^\pi(w_t, z_{t:\infty})$ can effectively be learned by heuristic estimations in input-repeatable environments such as within aforementioned simulations. Intuitively, input-repeatability allows an agent to distinguish whether observed state changes and received rewards were due to its taken actions or subject to the input sequence, since former simulations provide multiple trajectories to estimate $V^\pi(w_t, z_{t:\infty})$ under the fixed input $z_{t:\infty}$.

3.1 Multi-value-network approach

First, they propose to learn separate input-dependent baselines $b(w_t, z_{t:\infty})$ for each of N provided input-sequences $\{z^1, \dots, z^N\} \subseteq \mathcal{Z}$ via independent value-networks of parameters $\theta_1, \dots \theta_N$. During training, a previously recorded input sequence z^n is first sampled at random and subsequently applied to simulate trajectory τ under the acting policy π within the input-driven environment. Policy gradients ∇J are subsequently estimated according to τ while variance is reduced subject to the input-dependent baseline $V^\pi_{\theta_n}(w, z^n)$. Note, however, that although separate value-networks with parameters $\theta_1, \dots \theta_N$ are used to estimate state-values for the input-dependent baseline $b(w, z)$, policy π (represented by a policy-network of parameterization ψ) is mutually shared. Each iteration consequently updates ψ subject to policy gradient estimates $\nabla_\psi J$ along with the sampled input sequence's value-network θ_n.

3.2 Meta-learning approach

While their proposed multi-value network heuristic effectively reduces variance when confronted with few input sequences, the approach is not particularly scalable, as training numerous value-networks is severely resource demanding. Issues are further exacerbated by the fact that former procedure applies updates in a wasteful, sample inefficient manner. Consider, for instance, that each iteration solely updates the value-network's parameters θ_n of the sampled input sequence z^n, even though all input-dependent baselines $b(w, z)$ share significant similarities with regard to their joint state space as well as potential resemblance of respective inputs. Faster convergence and superior generalization could thus be accomplished by a heuristic that embraces the setting's similarities and leverages them in a scalable manner.

Consequently, Mao et al. leverage the meta-learning framework, which opts to generalize experience beyond single tasks to enable an efficient adaptation of knowledge to previously unseen tasks (learning to learn). They specifically aim to derive a joint meta state-value network with parameterization θ_V that integrates experience broadly applicable to various input sequences through Model-Agnostic Meta-Learning (MAML) [3]. Here, the key underlying idea is to train the meta state-value network as to prioritize parameterizations θ_V sensitive to changes in the input, such that few sampled trajectories suffice to adapt the network to an input sequence (few-shot reinforcement learning). Network gradients are accordingly updated as to improve the approximation quality of $V^{\pi}_{\theta_V}(w, z)$ across an entire sequence distribution $\mathcal{Z}$ rather than single sequences z. Instead of training θ_n anew for any recorded input sequence z^n, MAML rapidly adapts the broadly applicable parameterization θ_V as to approximate $V^{\pi}_{\theta_n}(w, z^n)$ with significantly less updates. Intuitively, their meta-learning based methodology enables learning state-values for several input sequences simultaneously from trajectories that solely relate to single sequences, which ultimately encourages faster convergence and improves scalability.

4 Results & Implementation

Mao et al. study the effectiveness of their proposed heuristics in several input-driven simulations related to robotic locomotion as well as networking and find that standard input-oblivious baselines consistently perform poorly across all examined tasks. Their meta-learning based methodology, in contrast, effectively reduces variance and ultimately outperforms their multi-value-network approach due to superior generalization capabilities and improved scalability. All experiments along with their implementation for the standard, multi-value-network and meta-learning based baseline are provided on GitHub[1].

5 Conclusion & Future Work

In their work, Mao et al. demonstrate consistent improvement of their proposed heuristics over standard state-value baselines when confronted with input-driven environments in various application domains. Although a dedicated treatment of external input processes generally increases the (input-driven) MDP's complexity and potentially exacerbates issues related to large state spaces, experimental results show arising benefits to outweigh their drawbacks.

Future work could explore an application of meta-learning beyond scaling input-dependent baselines to numerous input sequences. In particular, one-shot RL could generalize scheduling policies as to respect changes in the underlying network's topology which may manifest in joining or leaving compute notes. While prior studies primarily develop policies for the idealized assumption of static networks, meta-learning might enable a dynamic adaptation of policies which do not require expensive retraining. Leveraging meta-learning could consequently improve robustness against node failure or facilitate intelligent resource allocation within rapidly changing network topologies (e.g. peer-to-peer networks) where retraining is generally infeasible.

[1] `https://github.com/hongzimao/input_driven_rl_example`

References

[1] I. Clavera, A. Nagabandi, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. "Learning to adapt: Meta-learning for model-based control". In: *arXiv preprint arXiv:1803.11347* 3 (2018).

[2] H. Mao, S. B. Venkatakrishnan, M. Schwarzkopf, and M. Alizadeh. "Variance reduction for reinforcement learning in input-driven environments". In: *arXiv preprint arXiv:1807.02264* (2018).

[3] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[4] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[5] D. Daley. "Certain optimality properties of the first-come first-served discipline for G/G/s queues". In: *Stochastic Processes and their Applications* 25 (1987), pp. 301–308.