Full Gradient Deep Reinforcement Learning for Average-Reward Criterion

Tejas Pagare TEJASPAGARE2002@GMAIL.COM
Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India.

Vivek Borkar BORKAR.VS@GMAIL.COM
Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India.

Konstantin Avrachenkov K.AVRACHENKOV@INRIA.FR
INRIA Sophia Antipolis, 2004, Route des Lucioles, B.P.93, 06902, Valbonne, France.

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

We extend the provably convergent Full Gradient DQN algorithm for discounted reward Markov decision processes from Avrachenkov et al. (2021) to average reward problems. We experimentally compare widely used RVI Q-learning with recently proposed Differential Q-learning in the neural function approximation setting with Full Gradient DQN and DQN. We also extend this to learn Whittle indices for Markovian restless multi-armed bandits. We observe a better convergence rate of the proposed Full Gradient variant across different tasks.[1]

Keywords: average reward Markov decision processes, Full Gradient DQN algorithm, restless bandits, Whittle index

1. Introduction

Average reward Markov Decision Processes (MDPs) are popular stochastic models for the applications when the transient behaviour can be ignored and only the long term behaviour matters. Average reward MDPs find numerous applications in communication networks Altman (2002), medical treatment Schaefer et al. (2005) and machine maintenance Ross (2013). They are also a good approximation for discounted reward problems with discounts close to one and offer the simplicity of independence from the initial condition in the irreducible MDP case Bertsekas (2007). For these reasons, there is already a substantial body of work on reinforcement learning for this class of problems, e.g., Mahadevan (2005); Dewanto et al. (2020). In this work, we present function approximation algorithms for average reward Q-learning based on RVI Q-learning Abounadi et al. (2002) and Differential Q-learning Wan et al. (2021) with DQN Mnih et al. (2013) and Full Gradient DQN which is a variant of DQN proposed and proven to be convergent with the ODE-based analysis in Avrachenkov et al. (2021). Well-studied semi-gradient methods for Q-learning with function approximation lack theoretical guarantees and are shown to diverge Baird (1995). Then, the deadly triad of function approximation, bootstrapping, and off-policy training formalizes the divergence issue in RL algorithms Sutton and Barto (2018). For deterministic problems, the full gradient version, which tries to minimize the Bellman error using gradient descent, namely the residual algorithm, is shown to converge in Baird (1995). However, for stochastic problems, the

[1] Additional results and insights can be found in the arXiv version.

proposed residual algorithm considers two independently drawn samples from the simulator for a single update iteration. However, this so-called double-sampling is not possible in many learning scenario, typically in robotics and video games. Double-sampling is needed due to the product of expectations as opposed to the expectation of the product being required by the Bellman error in the update step. Full Gradient DQN, on the other hand, uses the novel experience replay to perform the first expectation approximately ahead of time. We further extend this to the problem of learning Whittle indices for Restless Multi-Armed Bandits (RMABs) with Q-learning with the average reward criterion, first studied for the tabular case in Avrachenkov and Borkar (2022). The Whittle index approach allows us to deal efficiently with many scheduling and resource allocation problems described in Whittle (1988); Papadimitriou and Tsitsiklis (1999); Gittins et al. (2011).

The paper is organized as follows. First, we consider a preliminary introduction to average reward Q-learning and propose our Full Gradient DQN (FGDQN) variant. We then propose a variant of Differential Q-learning for FGDQN. Applications to restless bandits are then discussed. Lastly, we show experimental results and conclude with some future directions.

Throughout, for a finite set S, we denote by $\mathcal{P}(S)$ the simplex of probability vectors indexed by the elements of S.

2. Preliminaries

Consider a controlled Markov chain $\{X_n, U_n\}, n \geq 0$, on a finite state space $\mathcal{S}, |\mathcal{S}| = s$, controlled by a process $U_n, n \geq 0$, taking values in a finite action space $\mathcal{A}, |\mathcal{A}| = \ell$, with transition probabilities $p(j|i,u), i,j \in \mathcal{S}, u \in \mathcal{A}$, satisfying $p(j|i,u) \in [0,1]\ \forall\, i,j,u$, and $\sum_j p(j|i,u) = 1\ \forall\, i,u$. Its time evolution is described by

$$\Pr(X_{n+1} = j | X_m, U_m, m \leq n) = p(j|X_n, U_n). \tag{1}$$

Given a per stage reward function $r : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{R}$, the average reward MDP seeks to maximize the time averaged reward

$$\liminf_{N\uparrow\infty} \mathbb{E}\left[\frac{1}{N}\sum_{n=0}^{N-1} r(X_n, U_n)\right]. \tag{2}$$

Special classes of interest are stationary policies, each associated with a map $v : \mathcal{S} \mapsto \mathcal{A}$ such that $U_n = v(X_n)\ \forall n$, and stationary randomized policies, each associated with a map $i \in \mathcal{S} \mapsto \varphi(\cdot|i) \in \mathcal{P}(\mathcal{A})$ such that $P(U_n = u | X_m, U_{m-1}, m \leq n) = \varphi(u|X_n)\ \forall n$. In particular, a stationary randomized policy with $\varphi(v(i)|i) = 1$ for $v : \mathcal{S} \mapsto \mathcal{A}$ corresponds to a stationary policy v. It is clear that $\{X_n\}$ is a Markov chain with transition probabilities $p(j|i,v(i))$, resp. $\sum_u p(j|i,u)\varphi(u|i)$ under a stationary policy v, resp. stationary randomized policy φ. We assume that this chain is irreducible under any v, resp. φ. Then it has a unique stationary distribution π_v, resp. π_φ in $\mathcal{P}(\mathcal{S})$ and (2) a.s. equals $\beta_v := \sum_i \pi_v(i) r(i,v(i))$, resp. $\beta_\varphi := \sum_{i,u} \pi_\varphi(i)\varphi(u|i) r(i,u)$. The objective is to maximize (2) over all admissible $\{U_n\}$, i.e., $\{U_n\}$ for which (1) holds. It is known that an optimal stationary policy exists (Puterman (1994), Chapter 8) and is characterized as the maximizer on the right hand side of the dynamic programming equation

$$V(i) = \max_u \left[r(i,u) - \beta + \sum_j p(j|i,u) V(j)\right],\ i \in \mathcal{S}. \tag{3}$$

This is an equation in unknowns $V(i), i \in \mathcal{S}$ and β. The β is characterized uniquely as the optimal reward β^* and $V(\cdot)$ is uniquely characterized modulo an additive constant.

We define the Q-values as the expression in the square brackets on the RHS of (3), i.e.,

$$Q(i,u) := r(i,u) - \beta + \sum_j p(j|i,u)V(j),\ i \in \mathcal{S}, u \in \mathcal{A}.$$

Then these satisfy the equation

$$Q(i,u) = r(i,u) - \beta + \sum_j p(j|i,u) \max_v Q(j,v),\ i \in \mathcal{S}, u \in \mathcal{A}. \tag{4}$$

Again, β is characterized uniquely as the optimal reward β^* and $Q(\cdot,\cdot)$ is uniquely characterized modulo an additive constant. If we solve this equation, the optimal stationary policy is given by $V(i) = \arg\max Q(i,\cdot)$, breaking ties arbitrarily if non-unique. Furthermore, this does not require the knowledge of the transition probabilities. Also, unlike (3), the nonlinearity, i.e., the 'max' operator is now inside the conditional expectation, which facilitates a stochastic approximation version (see, e.g., Borkar (2020)). This is the motivation for using the Q-values as a basis for reinforcement learning algorithms. We do this next.

3. Full Gradient DQN for Average Reward

A reinforcement learning algorithm to solve (4) called 'RVI Q-learning', based on the classical Relative Value Iteration for solving (3), was proposed and analyzed in Abounadi et al. (2002) for the tabular case. Here we study a variant based on function approximation using neural networks, that seeks to minimize the so-called 'Bellman error'. Thus, with some abuse of notation, let $Q(i,u;\theta)$ denote a parametrized family of approximating functions with parameter $\theta \in \mathcal{R}^d$ learned such that

$$Q(i,u;\theta) \approx r(i,u) - \beta + \sum_j p(j|i,u) \max_v Q(j,v;\theta),\ i \in \mathcal{S}, u \in \mathcal{A}. \tag{5}$$

This suggests minimizing the mean square error between the right and left hand sides of (5), i.e., minimizing the following Bellman error

$$\mathcal{E}(\theta) := \mathbb{E}\Big[\Big(r(X_n,U_n) - f(Q;\theta_n) + \sum_j p(j|i,u) \max_v Q(j,v;\theta) - Q(X_n,U_n;\theta)\Big)^2\Big]. \tag{6}$$

Here $f(Q;\theta_n)$ is the offset that works as a surrogate for β as in the RVI Q-learning of Abounadi et al. (2002). Since β is unknown, we use a surrogate $f(Q;\theta_n)$ by analogy with the original relative value iteration algorithm (see, e.g., Puterman (1994)). With a suitable choice, this can be shown to converge to β^* a.s. Some popular choices are $f(Q) = Q(i_0,u_0)$, $\max_u Q(i_0,u)$, $\max_{i,u} Q(i,u)$, $\frac{1}{s\ell}\sum_{i,u} Q(i,u)$, for some fixed $i_0 \in \mathcal{S}, u_0 \in \mathcal{A}$ etc. See Abounadi et al. (2002) for a characterization of admissible $f(\cdot)$.

This suggests the naive stochastic (sub)gradient scheme

$$\begin{aligned}\theta_{n+1} &= \theta_n + a(n)\Big(r(X_n,U_n) - f(Q;\theta_n) + \max_v Q(X_{n+1},v;\theta_n) - Q(X_n,U_n;\theta_n)\Big)\\ &\quad \times \left(\nabla_\theta f(Q;\theta_n) - \nabla_\theta Q(X_{n+1},v_{n+1};\theta_n) + \nabla_\theta Q(X_n,U_n;\theta_n)\right).\end{aligned} \tag{7}$$

Here $X_{n+1} \sim p(\cdot|X_n, U_n)$, $\{a(n)\}$ is a positive stepsize sequence satisfying the Robbins-Monro conditions $\sum_n a(n) = \infty$, $\sum_n a(n)^2 < \infty$, ∇_θ denotes the gradient with respect to the parameter vector θ and $v_{n+1} \in \text{Arg max}\, Q(X_{n+1}, \cdot; \theta_n)$[2] which renders the term $\nabla_\theta Q(X_{n+1}, v_{n+1}; \theta_n)$ a legitimate subgradient by Danskin's theorem Danskin (1966).

The catch with this is that if we apply the standard stochastic approximation theory, one sees the right hand side averaged with respect to the conditional distribution, i.e., the transition probability. But then we have a conditional expectation of the product of the two brackets as opposed to a product of their conditional expectations, as suggested by the gradient of (6). One way to avoid this was already proposed in the pioneering work of Baird (1995), which is to simulate another random variable $\widetilde{X}_{n+1}$ with exactly the same conditional distribution as X_{n+1} (i.e., $p(\cdot|X_n, U_n)$) and conditionally independent of it given $X_m, U_m, m \leq n$, then replace X_{n+1} in the second bracket by $\widetilde{X}_{n+1}$. But this is an awkward exercise and adds an additional overhead in the simulation, where the Markov chain simulator is often separately available and one does not want to 'dig into it'. Hence we use an alternative strategy based on experience replay, by replacing (7) by

$$\begin{aligned} \theta_{n+1} &= \theta_n + a(n)\overline{\Big(r(X_n, U_n) - f(Q;\theta_n) + \max_v Q(X_{n+1}, v; \theta_n) - Q(X_n, U_n; \theta_n)\Big)} \\ &\quad \times \Big(\nabla_\theta f(Q;\theta_n) - \nabla_\theta Q(X_{n+1}, v_{n+1}; \theta_n) + \nabla_\theta Q(X_n, U_n; \theta_n)\Big), \end{aligned} \tag{8}$$

where the overline denotes empirical average over triplets $(X_m, U_m, X_{m+1}), m < n$, for which $\{X_m = X_n \;\&\; U_m = U_n\}$. This sets it apart from the classical experience replay (i.e., sampling random transitions), but the advantage is that it achieves the approximate conditional expectation as desired for the first bracket of (7). That of the second bracket is then taken care of by the averaging effect of stochastic approximation. For deterministic problems, this is equivalent to the expression without the overline, thereby saving the extra computation needed to compute the first bracket.

We contrast this with what would be the natural extension of the DQN algorithm (Mnih et al. (2013)) for average reward, viz.,

$$\theta_{n+1} = \theta_n + \frac{a(n)}{B} \times \sum_{b=1}^{B} \Big((Z_{n(b)} - Q(X_{n(b)}, U_{n(b)}; \theta_n))\nabla_\theta Q(X_{n(b)}, U_{n(b)}; \theta_n)\Big), \tag{9}$$

where $(X_{n(b)}, U_{n(b)}, X_{n(b)+1}), 1 \leq b \leq B$ are randomly selected triplets from the experience replay, and

$$Z_{n(b)} = r(X_{n(b)}, U_{n(b)}) + \max_v Q(X_{n(b)+1}, v; \tilde{\theta}_n) - f(Q; \tilde{\theta}_n) \tag{10}$$

is the 'target' being chased by $Q(X_{n(b)}, U_{n(b)})$ and $\tilde{\theta}_n$ is updated on a slower time scale by setting it equal to θ_n periodically and left unaltered in between. As pointed out in Avrachenkov et al. (2021), there are several theoretical issues about DQN, but it remains a popular scheme because of good empirical behavior. We shall numerically compare our scheme (8) above with (10).

We conclude this section with some comments on the convergence analysis, which we do not pursue in detail here because it goes more or less along standard lines. If the averaging due to experience replay is exact, we have an exact stochastic subgradient scheme which under reasonable 'richness' condition on the noise, is known to converge a.s. to a local minimum (or to a connected set

[2] Arg max is used to denote the possible non-uniqueness of the arg max operation.

thereof if they are not isolated). The condition on noise requires that it be adequate in all directions, which is easily ensured, e.g., as in Avrachenkov et al. (2021), by adding a small extraneous zero mean noise. But this is rarely required in practice. The errors due to inexact averaging by experience replay etc., if small, will lead to a weaker claim, viz., convergence to a small neighborhood of a local minimum. These claims are standard in stochastic approximation theory, see, e.g., Chapter 11 of Borkar (2020).

4. Differential Q-learning

We modify the Differential Q-learning algorithm described in Wan et al. (2021) for off-policy control in tabular setting to neural function approximation. The idea here is to maintain a scalar proxy $\bar{R}$ similar to $f(Q)$ in RVI Q-learning which is updated based on the temporal difference error. In the following we consider a variant based on Full Gradient DQN. Due to the θ dependence of $\bar{R}$, we maintain similar iteration for $\nabla_\theta \bar{R}$ denoted as Y_n which is used in the full gradient of Q-iteration as follows:

$$\theta_{n+1} = \theta_n - a(n)\overline{\Big(r(X_n, U_n) + \max_{a'} Q(X_{n+1}, a'; \theta_n) - \bar{R}_n - Q(X_n, U_n; \theta_n)\Big)} \times \Big(\nabla_\theta Q(X_{n+1}, v_{n+1}; \theta_n) - Y_n - \nabla_\theta Q(X_n, U_n; \theta_n)\Big) \tag{11}$$

$$\bar{R}_{n+1} = \bar{R}_n + \frac{\eta a(n)}{|\mathcal{S}||\mathcal{A}|} \times \sum_{s\in\mathcal{S}, a\in\mathcal{A}} \Big(r(s,a) + \max_{a'} Q(s', a'; \theta_n) - \bar{R}_n - Q(s, a; \theta_n)\Big) \tag{12}$$

$$Y_{n+1} = \nabla_\theta \bar{R}_{n+1} = Y_n + \frac{\eta a(n)}{|\mathcal{S}||\mathcal{A}|} \times \sum_{s\in\mathcal{S}, a\in\mathcal{A}} \Big(\nabla_\theta Q(s', v; \theta_n) - Y_n - \nabla_\theta Q(s, a; \theta_n))\Big) \tag{13}$$

where $s' \sim p(\cdot|s,a)$, $v \in \operatorname{Arg\,max} Q(s', \cdot; \theta_n)$ and η is a positive constant which can be thought of as a parameter which controls the speed of the $\bar{R}$ update w.r.t. Q update. Similarly, for the DQN variant, we can use the semi-gradient in the Q-iteration, and we won't need Y_n, i.e., $Y_n = 0\ \forall n$.

5. Application to Restless Bandits

The problem of Markovian restless bandits is as follows. One has $N > 1$ Markov chains $\{X_n^i, n \geq 0\}, 1 \leq i \leq N$, taking values in discrete state spaces S^i resp. Each has two 'modes' of operation, active and passive, with corresponding transition probabilities and per stage rewards given by $p_a^i(j|k), r_a^i(k), j, k \in S^i$, for the active mode and $p_b^i(j|k), r_b^i(k), j, k \in S^i$, for the passive mode, respectively. The problem is to maximize the average reward

$$\liminf_{n\uparrow\infty} \mathbb{E}\left[\frac{1}{n}\sum_{m=0}^{n-1}\sum_{i=1}^{N}(\nu_m^i r_a^i(X_m^i) + (1-\nu_m^i) r_b^i(X_m^i))\right], \tag{14}$$

where $\nu_m^i = 1$ if the ith chain is in the active mode and zero, otherwise. This is to be done subject to the constraint

$$\sum_{i=1}^{N} \nu_n^i = M\ \forall n, \tag{15}$$

for a prescribed $M < N$. In a classic article, Whittle (1988) approached this problem, now known to be PSPACE-hard Papadimitriou and Tsitsiklis (1999), by first relaxing the per stage constraint (15) to the average constraint

$$\liminf_{n\uparrow\infty} \mathbb{E}\left[\frac{1}{n}\sum_{m=0}^{n-1}\sum_{i=1}^{N}\nu_m^i\right] = M, \tag{16}$$

which makes it a standard constrained MDP with average reward. Using the Lagrange multiplier approach (which can be rigorously justified), it becomes an unconstrained average reward MDP with separable reward and separable constraint function. The Lagrange multiplier then decouples it into separate uncoupled MDPs. Using this as a motivation, Whittle first introduced a subsidy λ for passivity added to the reward for being passive, and called the problem (Whittle) indexable if the states in which it is optimal to be passive under subsidy λ, increases from the empty space to the entire state space as λ increases from $-\infty$ to ∞. If so, assign to each state k, the (Whittle) index $\lambda^*(k)$ defined as that value of the subsidy λ for which both active and passive modes are equally desirable. The (Whittle) index policy then at time n is to sort $\lambda^*(X_n)$ in decreasing order (any ties being resolved according to some pre-specified rule) and then render the top M chains active, the rest passive. This is a heuristic that is known to work well in practice and is asymptotically optimal in a certain sense when $N \uparrow \infty$ Weber and Weiss (1990); Gittins et al. (2011).

The chains are now coupled only through the index policy and the definition of the index for a chain depends on the chain alone. Hence, we drop the superscript i henceforth and look at a single individual chain controlled by a $\{0,1\}$-valued control process $\{\nu_n\}$ so as to maximize the average reward

$$\liminf_{n\uparrow\infty} \mathbb{E}\left[\frac{1}{n}\sum_{m=0}^{n-1}(\nu_m^i r_a^i(X_m) + (1-\nu_m) r_b^i(X_m))\right]. \tag{17}$$

Let $Q(i,u), i \in \mathcal{S}, u \in \{0,1\}$, be the Q-values for this constrained problem. It is clear that the condition for the threshold for a switch from passive to active or vice versa at state $\hat{k}$ is given by the equation

$$Q(\hat{k},1) = Q(\hat{k},0),$$

where the dependence of $Q(\hat{k},\cdot)$ on λ is implicit. Thus, to obtain the Whittle index for state $\hat{k}$, one needs to solve this simultaneous equation in $Q(\cdot,\cdot)$ and λ. The solution perforce will depend on the choice of $\hat{k}$. We parameterize λ by a parameterized family $(\hat{k},\sigma) \to \lambda(\hat{k};\sigma)$ and Q by parameterized family $(x,u,\lambda(\hat{k};\sigma),\theta) \to Q(x,u,\lambda(\hat{k};\sigma);\theta)$ where $x \in \mathcal{S}$ and $u \in \{0,1\}$. This is done to render the implicit dependence of Q on $\lambda(\hat{k})$ for each $\hat{k}$ separately.
The above equation suggests minimizing the following error

$$\bar{\mathcal{E}}(\sigma) = \mathbb{E}\Big[\Big(Q(\hat{k},1,\lambda(\hat{k};\sigma);\theta) - Q(\hat{k},0,\lambda(\hat{k};\sigma);\theta)\Big)^2\Big]. \tag{18}$$

The update iteration for σ based on the mini-batch stochastic gradient descent is as follows

$$\sigma_{n+1} = \sigma_n - \frac{b(n)}{B} \times \sum_{b=1}^{B} \nabla_\sigma \Big(Q(X_{n(b)},1,\lambda(X_{n(b)};\sigma_n);\theta_n) - Q(X_{n(b)},0,\lambda(X_{n(b)};\sigma_n);\theta_n)\Big)^2 \tag{19}$$

where the stepsizes $\{b(n)\}$ satisfy

$$b(n) > 0,\ \sum_n b(n) = \infty,\ \sum_n b(n)^2 < \infty,\ b(n) = o(a(n)),$$

and the $Q(j,u,\lambda(\hat{k};\sigma);\theta_n), j \in \mathcal{S}, u \in \{0,1\}$ values are updated *for each* $\hat{k} \in \mathcal{S}$ *separately* with modified reward as

$$\tilde{r}(X_n, U_n; \hat{k}) = (1-U_n)(r_b(X_n) + \lambda(\hat{k};\sigma)) + U_n r_a(X_n)$$

using (8) for the FGDQN algorithm and (10) for the DQN algorithm. Overall, the algorithm can be viewed as two time scale stochastic approximation where the Whittle index is updated on a slower timescale and Q-values are updated on a faster time scale. In general one would maintain different Q and λ neural networks for different arms, which can be shared for statistically identical arms reducing the search space for Q-learning from $(2|\mathcal{S}|)^N$ to $2|\mathcal{S}|^2 + |\mathcal{S}|$.

6. Numerical Results

In our experiments, we test RVI Q-learning and Differential Q-learning with function approximation based on FGDQN and DQN. We consider infinite-horizon problems with varying difficulty : Forest Management Chizat and Bach (2018), Access Control Queuing Sutton and Barto (2018) and Catcher Tasfi (2016). We further modify the FGDQN iteration (8) to incorporate the benefit of mini-batch as in DQN to reduce variance as follows:

$$\begin{aligned}\theta_{n+1} \quad = \quad & \theta_n - \frac{a(n)}{B} \times \sum_{b=1}^{B} \Bigg(\overline{\Big(r(X_{n(b)}, U_{n(b)}) + \max_v Q(X_{n(b)+1}, v; \theta_n) - f(Q;\theta_n) } \\ & \overline{- Q(X_{n(b)}, U_{n(b)}; \theta_n) \Big)} \times \Big(\nabla_\theta Q(X_{n(b)+1}, v_{n(b)+1}; \theta_n) - \nabla_\theta f(Q;\theta_n) - \\ & \nabla_\theta Q(X_{n(b)}, U_{n(b)}; \theta_n) \Big) \Bigg), \quad n \geq 0, \end{aligned} \tag{20}$$

where $(X_{n(b)}, U_{n(b)}, X_{n(b)+1}), 1 \leq b \leq B$ are randomly selected triplets from the experience replay, and $v_{n(b)+1} \in \text{Arg max}\, Q(X_{n(b)+1}, \cdot; \theta)$.
For FGDQN, we use $f(Q) = Q(s_0, a_0)$ for fixed $s_0 \in \mathcal{S}$ and $a_0 \in \mathcal{A}$, which can be obtained by selecting the most frequently occurring state-action pair in the replay buffer filled by a fixed stationary randomized policy at the start of the training. Fig. 1[3] shows evaluation average reward calculated by running the Q-value maximizing policy for 1000 time steps. In Fig. 2 we plot

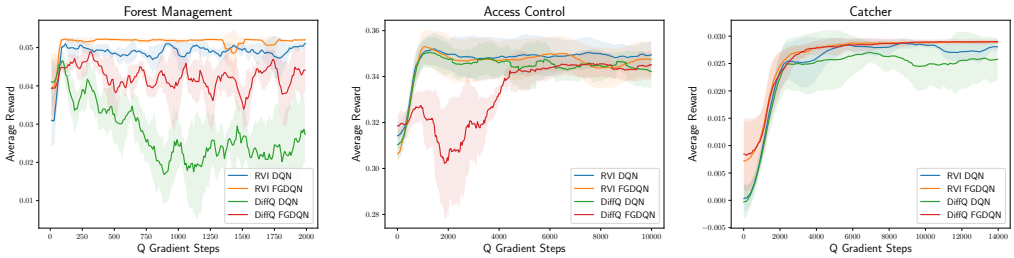

Figure 1: Evaluation average reward

[3] All the figures show data on five different randomly chosen seeds. Further, we show the line plots with moving average (dark) over a defined window for better analysis and 95% confidence interval (light).

$Q(s_0, a_0; \tilde{\theta})$[4], $Q(s_0, a_0; \theta)$ for RVI DQN and RVI FGDQN resp. and $\bar{R}$ for Differential Q-learning (DiffQ). Asymptotically we want these values to converge to the true average reward β^*. This convergence can be observed for the RVI DQN, RVI FGDQN and DiffQ FGDQN, but not for DiffQ DQN, for which $\bar{R}$ remains close to 0 and hence attributes to relatively poor performance of DiffQ DQN in the Fig. 1. Note in Access Control, DiffQ FGDQN converges when $\bar{R}$ converges to the true value at about 4000 Q-gradient steps. Overall, this shows the direct correlation between the convergence of $\bar{R}$ and the algorithm's performance. In all the experiments, we set $\bar{R} = 0$ i.e. $\min r(\cdot, \cdot)$ during initialization which plays a significant role in the convergence rate of the DiffQ algorithm.

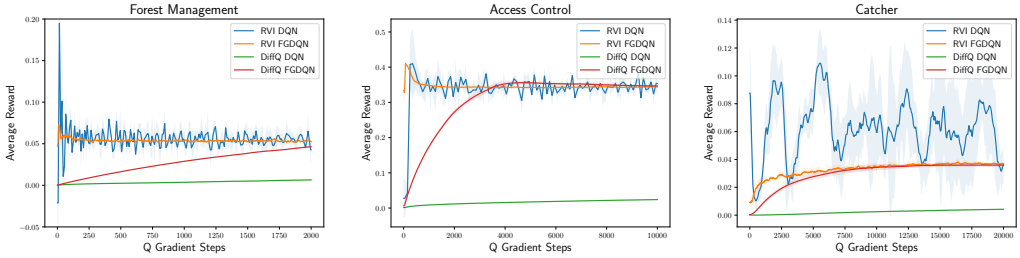

Figure 2: Average reward proxy

6.1. Restless Bandits

For Restless Multi-Armed Bandits, we consider the problems of Circulant dynamics Fu et al. (2019), Restart problem Avrachenkov and Borkar (2022) and Deadline scheduling Yu et al. (2018). We observe unstable learning and thus poor performance of Differential Q-learning hence we don't show them here.

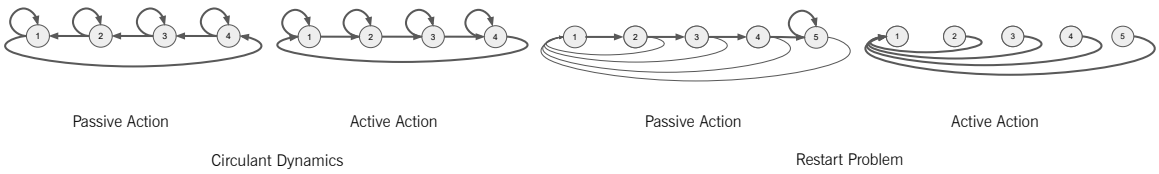

Figure 3: Markov Chains

Fig. 3 shows the Markov chains for active and passive action for both Circulant dynamics and Restart problem. In circulant dynamics problem the state increments (decrements) or remains same with same probability for active (passive) action. Reward function is given as $r(1, a) = -1$, $r(4, a) = 1$ and $r(s, a) = 0$ for others, independent of the action. Restart problem, restarts the MDP to state 1 for active action and for passive action, it increments (for $s = 5$ it stays at 5) or restarts to state 1 with probability 0.9 and 0.1 resp. The reward function $r(s, 1) = 0.9^s$, and $r(s, 0) = 0$ for state $s \in \{1, 2, 3, 4, 5\}$.

Deadline Scheduling: The problem considers scheduling of jobs depending upon the job arrival, workload, deadline for completion, and the processing cost. One such application can be scheduling of Electrical Vehicles (EV) on a charging station. The charging cost depends on the cost

[4]We observe the zigzag effect in RVI DQN since the target network $\tilde{\theta}$ is updated periodically by setting it to θ.

of electricity at the time of charging, and a penalty is imposed when the service provider is unable to fulfill the request. We consider the problem of EV charging station similar to Nakhleh et al. (2021), consisting of N charging stations which can charge M vehicles at any instant of time. For any instant, an EV arrives which declares the amount of charging needed b units and the hard deadline d. Action is to charge or not charge the vehicle at every instant which are active and passive actions resp. based upon the deadline $D \in [0, d]$ and load $B \in [0, b]$. Reward is $1 - c$ for every unit of charge where c is the cost of electricity. At the deadline if the vehicle is not fully charged, a penalty is received depending upon the amount of charge remaining. After the deadline, new vehicle arrives with probability 0.3. The goal of the station is to maximize the infinite horizon average reward. We characterize a problem with maximum load $B_{\max}$ and deadline $D_{\max}$ possible, which leads to problem with state space of size $(B_{\max} + 1) \times (D_{\max} + 1)$. We consider two variants, first with $D_{\max} = 12, B_{\max} = 9$ and $D_{\max} = 50, B_{\max} = 45$ with 130 and 2346 number of states resp.

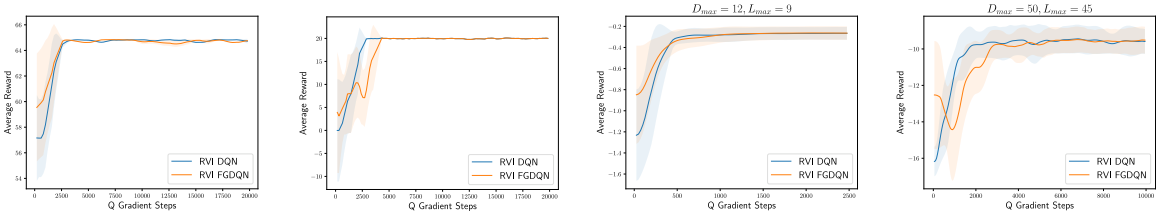

Figure 4: Evaluation average reward

For all the problems we consider the restless bandit scenerio with $N = 100, M = 20$ for circular and restart problems and $N = 4, M = 1$ for deadline scheduling problem. Here we consider the shared architecture, which implies using the same neural network for Q-values and Whittle index λ for all arms. This is done since all the arms are statistically identical (i.e. have same transition matrix and reward function). Fig. 4 shows average evaluation reward calculated by running the index policy for 5000 time steps for deadline scheduling with $D_{\max} = 50, B_{\max} = 45$ and 1000 time steps for others. We consider $f(Q) = Q(s_0, u_0)$ for circular and restart problems and $f(Q) = \max_u Q(s_0, u)$ for deadline scheduling where s_0, u_0 are fixed state and action, which we found to be stable in the respective problems. It can be observed that overall FGDQN starts off quickly compared to DQN and has better convergence rate than DQN for 2 out of 4 problems.

7. Conclusion

We observe that the introduced full gradient variant of DQN (FGDQN) for average reward criterion, at the expense of extra computation, outperformed DQN on some tasks for both RVI Q-learning and Differential Q-learning algorithms. Further, we presented an elegant way of solving the problem of restless bandits based on Q-learning in case of complex problems. There are promising research directions where one could consider risk-sensitive control with Q-learning Borkar (2002), which has applications in finance for portfolio optimization Bielecki and Pliska (1999). On the theoretical side, finite-time Chandak et al. (2022) and regret analysis of these average reward algorithms can also be explored.

8. Appendix - Pseudocode

$$\theta_{n+1} = \theta_n - \frac{a(n)}{\mathcal{B}} \times \sum_{b=1}^{\mathcal{B}} \left(E_b \times \left(\nabla_\theta Z_b - \nabla_\theta Q(x_{n(b)}, u_{n(b)}; \theta_n) \right) \right), \quad n \geq 0, \tag{21}$$

Problems where state space is huge, iterating over all $\hat{k} \in \mathcal{S}$ is time consuming, hence, we instead sample a batch of reference states to update Q-values and Whittle index λ.

Algorithm 1 Whittle Indices with FGDQN (Statistically Identical Arms)

Input: replay memory $\mathcal{D}$, batch size $\mathcal{B}$, exploration probability ϵ, total gradient steps T.
Initialise the weights θ & σ randomly for the Q-network and Whittle-network. Consider RMAB problem with N statistically identical chains such that at every time step M of them are active. Denote state of the system at time n as $X(n) = (x_1(n), \ldots, x_N(n))$ where $x_i(n) \in \{1, \ldots, d\}$ is a state of chain for $i \in \{1, \ldots, N\}$. Similarly, denote reward and action vectors as $R(n)$ and $U(n)$.
for n = 1 *to* T **do**
  if *Uni[0,1]* $< \epsilon$ **then**
    Select $U(n) = (u_1(n), \ldots, u_N(n))$ at random such that $\sum_{i=1}^{N} u_i(n) = M$
  else
    Select $u_i(n) = 1$ for largest M Whittle indices $\lambda(u_i(n); \sigma)$
  end
  Add transitions $\{x_i(n), u_i(n), r_i(n), x_i'(n)\}\ \forall i \in \{1, \ldots, N\}$ in $\mathcal{D}$.
  Sample random batch of size $\mathcal{B}$ transitions from $\mathcal{D}$.
  Sample $\mathcal{B}$ number of reference states from $\mathcal{D}$.
  for $b = 1$ *to* $\mathcal{B}$ **do**
    Sample transitions with fixed state-action pair as x_b, u_b.
    Set $E_b = \overline{r_b + (1-u_b)\lambda(\hat{k}_b;\theta) + \max_v Q(x_b', v, \lambda(\hat{k}_b);\theta) - f(Q(\lambda(\hat{k}_b;\theta)))}$
    $\overline{-Q(x_b, u_b, \lambda(\hat{k}_b);\theta)}$
    Set $Z_b = r_b + (1-u_b)\lambda(\hat{k}_b;\theta) + \max_v Q(x_b', v, \lambda(\hat{k}_b);\theta) - f(Q(\lambda(\hat{k}_b;\theta)))$
  end
  Compute gradient for the average loss over the batch $\mathcal{B}$ and using Eq. (21) update parameters θ.

  On slower time-scale **do**
  Sample $\mathcal{B}$ number of reference states from $\mathcal{D}$.
  Perform mini-batch stochastic gradient descent for following mean-squared loss to update σ using Eq. (19)

$$\bar{\mathcal{E}}(\sigma) = \|Q(\hat{k}, 1, \lambda(\hat{k};\sigma);\theta) - Q(\hat{k}, 0, \lambda(\hat{k};\sigma);\theta)\|^2$$

end

For continuous state-space, our modified experience replay can be deployed by retrieving from the memory buffer state-control-next state triplets that are sufficiently representative according to proximity in some a priori chosen feature space. The feature space may also be learned, but that calls for a separate algorithm for the purpose as a pre-processing step.

Acknowledgments

Computing resources were provided by Compute Canada. We would like to acknowledge the support of Cefipra-Inria project "Learning In Operations and Networks (LION)". VB's work was supported in part by a S. S. Bhatnagar Fellowship from the Government of India. We thank Abhishek Naik for a few useful suggestions about the experiments.

References

Jinane Abounadi, Dimitri Bertsekas, and Vivek S. Borkar. Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2002. ISSN 0363-0129. doi: 10.1137/S0363012999361974. Copyright: Copyright 2008 Elsevier B.V., All rights reserved.

Eitan Altman. *Applications of Markov Decision Processes in Communication Networks*, pages 489–536. Springer US, Boston, MA, 2002. ISBN 978-1-4615-0805-2. doi: 10.1007/978-1-4615-0805-2_16. URL https://doi.org/10.1007/978-1-4615-0805-2_16.

Konstantin E. Avrachenkov and Vivek S. Borkar. Whittle index based Q-learning for restless bandits with average reward. *Automatica*, 139(C), may 2022. ISSN 0005-1098. doi: 10.1016/j.automatica.2022.110186. URL https://doi.org/10.1016/j.automatica.2022.110186.

Konstantin E. Avrachenkov, Vivek S. Borkar, Harsh P. Dolhare, and Kishor Patil. Full gradient DQN reinforcement learning: A provably convergent scheme. In Alexey Piunovskiy and Yi Zhang, editors, *Modern Trends in Controlled Stochastic Processes:*, pages 192–220, Cham, 2021. Springer International Publishing. ISBN 978-3-030-76928-4.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 30–37. Morgan Kaufmann, San Francisco (CA), 1995. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50013-X. URL https://www.sciencedirect.com/science/article/pii/B978155860377650013X.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 3rd edition, 2007. ISBN 1886529302.

Tomasz R. Bielecki and Stanley R. Pliska. Risk-sensitive dynamic asset management. *Applied Mathematics and Optimization*, 39(3):337–360, Jun 1999. ISSN 1432-0606. doi: 10.1007/s002459900110. URL https://doi.org/10.1007/s002459900110.

Vivek S. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2): 294–311, 2002. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3690589.

Vivek S. Borkar. *Stochastic Approximation. A Dynamical Systems Viewpoint (2nd edition)*. Hindustan Book Agency, New Delhi, and Springer Nature, New Delhi, 2020.

Siddharth Chandak, Vivek S. Borkar, and Parth Dodhia. Concentration of contractive stochastic approximation and reinforcement learning. *Stochastic Systems*, 12(4):411–430, 2022.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3040–3050, Red Hook, NY, USA, 2018. Curran Associates Inc.

John M. Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966. ISSN 00361399. URL http://www.jstor.org/stable/2946123.

Vektor Dewanto, George Dunn, Ali Eshragh, Marcus Gallagher, and Fred Roosta. Average-reward model-free reinforcement learning: a systematic review and literature mapping, 2020. URL https://arxiv.org/abs/2010.08920.

Jing Fu, Yoni Nazarathy, Sarat Moka, and Peter G. Taylor. Towards q-learning the whittle index for restless bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*, pages 249–254, 2019. doi: 10.1109/ANZCC47194.2019.8945748.

John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22:159–195, 2005.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing Atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Khaled Nakhleh, Santosh Ganji, Ping-Chun Hsieh, I-Hong Hou, and Srinivas Shakkottai. NeurWIN: Neural Whittle index network for restless bandits via deep RL. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 828–839. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/0768281a05da9f27df178b5c39a51263-Paper.pdf.

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3690486.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Sheldon M Ross. *Applied probability models with optimization applications*. Courier Corporation, 2013.

Andrew J Schaefer, Matthew D Bailey, Steven M Shechter, and Mark S Roberts. Modeling medical treatment using Markov decision processes. In *Operations research and health care*, pages 593–612. Springer, 2005.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Norman Tasfi. Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment, 2016.

Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward markov decision processes. In *International Conference on Machine Learning*, pages 10653–10662. PMLR, 2021.

Richard R. Weber and Gideon Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990. ISSN 00219002. URL http://www.jstor.org/stable/3214547.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988. ISSN 00219002. URL http://www.jstor.org/stable/3214163.

Zhe Yu, Yunjian Xu, and Lang Tong. Deadline scheduling as restless bandits. *IEEE Transactions on Automatic Control*, 63(8):2343–2358, 2018. doi: 10.1109/TAC.2018.2807924.