2018 Indian Control Conference (ICC)
January 4-6, 2018. IIT Kanpur, India

A Reinforcement Learning Algorithm for Restless Bandits

Vivek S. Borkar[1] and Karan Chadha[2]

***Abstract*— We propose and analyze a reinforcement learning algorithm for learning Whittle index for a class of indexable restless bandits based on linear function approximation and illustrate its use using as an example a restless bandit problem arising in scheduling of web crawlers for ephemeral content.**

I. Introduction

Restless bandits is a popular paradigm in many resource allocation problems arising in engineering, see [7], [9], [10], [13] for a sampler. Unlike the classical multi-armed Markov bandits which remain frozen when not in use, restless bandits *do not*. Also, unlike in the classical case, a provably optimal index policy is not known for restless bandits. Instead one has an ingenious heuristic due to Whittle [18] based on a relaxation of the original problem. This heuristic is asymptotically optimal in the 'infinitely many bandits' limit [17]. More importantly, it works very well in practice, as aforementioned references demonstrate. The key property the restless bandits need to satisfy for this to be possible is the so called Whittle indexability. After introducing this notion below, we identify a large class of problems where this can be reduced to proving monotonicity of threshold in an optimal threshold policy, which can be shown to exist for many cases of interest. Nevertheless, an explicit expression for the index may be hard to come by. We propose here a simulation based reinforcement learning scheme to approximately evaluate the Whittle index when an optimal threshold policy exists. As an illustration, we apply this scheme to a problem of scheduling web crawlers for ephemeral content [2].

The next section recalls the relevant theory surrounding the Whittle index. Section 3 describes the reinforcement learning scheme, followed by numerical experiments for the aforementioned example in section 4. Section 5 concludes with some remarks.

II. Restless bandits

A. Whittle index

The restless bandit paradigm goes as follows. Consider $N > 1$ Markov chains $\{X_n^i, n \geq 0\}$ on state spaces $S^i, 1 \leq i \leq N$, resp., assumed to be intervals in $\mathbf{Z}$ or $\mathcal{R}$. We assume that each of these chains has two modes: active and passive, characterized by two (irreducible) transition kernels $p_a^i(dx'|x),\ p_b^i(dx'|x)$ and rewards $\mathcal{R}_a^i(x) > R_b^i(x), x \in S^i$, resp. We assume that the maps $x \mapsto p_a^i(dx'|x), p_b^i(dx'|x)$ are continuous as maps $S^i \mapsto$ the space $\mathcal{P}(S^i)$ of probability measures on S^i with Prohorov topology. Only $M < N$ bandits can be active at any given time. The problem is to make this choice optimally. Formally, let $\nu^i(n) := I\{i\text{th bandit is active at time } n\}$ (here $I\{\cdots\} :=$ the indicator function which is 1 if '$\cdots$' is true, 0 otherwise). Then the objective is to maximize the time-averaged reward

$$\liminf_{n\uparrow\infty} \frac{1}{n} \sum_{m=0}^{n-1} \sum_{i=1}^{N} E\Big[\nu^i(m)R_a^i(X_m^i)+(1-\nu^i(m))R_b^i(X_m^i)\Big]$$

subject to the constraint

$$\sum_{i=1}^{N} \nu^i(n) = M\ , \nu^i(n) \in \{0,1\}\ \ \forall n \geq 0. \tag{1}$$

This turns out to be a hard problem (see, e.g., [11]), so Whittle [18] proposed a relaxation of the per stage constraint (1) by the averaged constraint

$$\limsup_{n\uparrow\infty} \frac{1}{n} E\left[\sum_{m=0}^{n-1} \sum_{i=1}^{N} \nu^i(n)\right] \leq M,\ \forall n \geq 0. \tag{2}$$

Whittle introduced a parameter λ akin to the Lagrange multiplier, interpreted as a subsidy for passivity, and considered the unconstrained problem of maximizing for each i

$$\liminf_{n\uparrow\infty} \frac{1}{n} \sum_{m=0}^{n-1} \sum_{i=1}^{N} E\Big[\nu^i(m)R_a^i(X_m^i) + (1-\nu^i(m))(\lambda + R_b^i(X_m^i))\Big].$$

This is a separable problem that reduces to solving the individual problems of maximizing

$$\liminf_{n\uparrow\infty} \frac{1}{n} \sum_{m=0}^{n-1} E\Big[\nu^i(m)R_a^i(X_m^i)+(1-\nu^i(m))(\lambda+R_b^i(X_m^i))\Big].$$

The corresponding dynamic programming equation then is

$$V^i(x) + \beta = \max_{u\in\{0,1\}} \Big[uR_a^i(x) + (1-u)(\lambda + R_b^i(x))+ \int_{S^i} (up_a^i(dx'|x) + (1-u)p_b^i(dx'|x))V^i(x')\Big] \tag{3}$$

$$= \max\Big[R_a^i(x) + \int_{S^i} p_a^i(dx'|x)V^i(x'), \lambda + R_b^i(x) + \int_{S^i} p_b^i(dx'|x)V^i(x')\Big]. \tag{4}$$

Here β is the optimal reward. The problem is said to be (Whittle) indexable if the subset of S^i for which it is optimal

VSB was supported in part by a J. C. Bose Fellowship from the Government of India and CEFIPRA grant No. IFC/DST-Inria-2016-01/448 "Machine Learning for Network Analytics".

[1]Vivek S. Borkar is with Faculty of Electrical Engineering,Indian Institute of Technology Bombay, Powai, Mumbai 400076, India borkar.vs@gmail.com

[2]Karan Chadha is a with Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India karanchadhaiitb@gmail.com

to remain passive increases monotonically from ϕ to S^i as λ increases from $-\infty$ to ∞. If so, the Whittle index $\lambda^i : S^i \mapsto \mathbb{R}$ is defined as that value of λ for which both active and passive modes are equally desirable, i.e.,

$$R_a^i(x) + \int_{S^i} p_a^i(dx'|x)V^i(x') = \lambda^i(x) + R_b^i(x) + \int_{S^i} p_b^i(dx'|x)V^i(x'). \quad (5)$$

Whittle's heuristic is to arrange at time n the numbers $\{\lambda^i(X_n^i)\}$ in decreasing order (any tie being resolved arbitrarily) and render the top M according to this list active, the rest remain passive.

This is the original Whittle framework. A corresponding theory is possible for minimization problems and/or the discounted rewards/costs.

B. Threshold policies

In many problems of interest, Whittle indexability can be proved by first proving the existence of an optimal threshold policy. There is a broad template for doing this which is applicable to a large subclass of problems. Of course, one needs suitable modifications to account for problem specifics. These are usually not entirely routine. The template goes as follows. (We give a bare sketch, the details are heavily problem-dependent as already mentioned.) First one justifies the average reward dynamic programming equation (3)/(4), typically by the classical 'vanishing discount' argument [12]. Then one proves certain structural properties of V^i. Suppose $R_a^i - R_b^i$ satisfy a suitable convexity property relative to S^i. Further, suppose that the state resulting from an active transition stochastically dominates the state resulting from a passive transition in the sense that the corresponding transition probabilities satisfy the standard stochastic dominance condition [15]. Then one can prove monotone increasing property and convexity for V^i, first proving these properties for the finite horizon discounted problem simply by backward recursion, then for the infinite horizon discounted problem by letting the time horizon $\uparrow \infty$, and finally for the average reward problem above by passing through the vanishing discount limit. See [1] for how this is typically done. Since convexity implies increasing differences, one can then conclude the existence of an optimal threshold policy from (5), i.e., the existence of an $\tilde{x}_i \in S^i$ such that it is optimal to be active for $x \geq \tilde{x}_i$ and passive otherwise.

Clearly $\tilde{x}_i, \beta$ depend on λ, hence we may write $\tilde{x}_i = \tilde{x}_i(\lambda), \beta = \beta(\lambda)$. Then Whittle indexability requires that $\lambda \mapsto \tilde{x}_i(\lambda)$ be monotone increasing. This can be proved by first establishing the increasing cross-differences property defined as follows. First write $V^i(x)$ as $\breve{V}^i(\lambda, x)$ to make its λ-dependence explicit. Then this property is:

$$\lambda' > \lambda,\ y > x \Longrightarrow \breve{V}^i(\lambda', y) - \breve{V}^i(\lambda', x) \geq \breve{V}^i(\lambda, y) - \breve{V}^i(\lambda, x). \quad (6)$$

This is related to the notion of supermodularity, see Chapter 10 of [14]. Using this in conjunction with the convexity of $\lambda \mapsto \beta(\lambda)$ which is apparent from (3)-(4), one can usually work out a 'proof by contradiction' for Whittle indexability.

III. The Reinforcement Learning Scheme

Computation of Whittle index involves each bandit separately, hence we drop now the superscript i tagging the bandit and consider a single generic bandit. Given a fixed threshold policy with threshold (say) $\tilde{x}$, the fixed policy (i.e., uncontrolled) dynamic programming equation reduces to the so called Poisson equation, which is the linear system

$$V(x) = R_a(x) - \beta + \int p_a(dx'|x)V(x'),\ x \geq \tilde{x}, \quad (7)$$

$$V(x) = \lambda + R_b(x) - \beta + \int p_b(dx'|x)V(x'),\ x < \tilde{x}. \quad (8)$$

It is not always easy to justify this equation, see [1] for one approach to doing so based on an abstract coupling argument. In many cases arising in applications, one can do this successfully and in addition, show that this characterizes β uniquely as the reward of the threshold policy with threshold $\tilde{x}$, and V uniquely up to an additive scalar constant. In particular, the latter may be rendered unique by freezing its value at a prescribed state to, e.g., zero. The λ can be identified with the Whittle index for $\tilde{x}$, that is,

$$\lambda(x) = (R_a(x) + \int p_a(dx'|x)V(x')) - (R_b(x) + \int p_b(dx'|x)V(x')),$$

where $x = \tilde{x}$. This suggests computing the Whittle index for a fixed $\tilde{x}$ by minimizing the squared error

$$\Big(\lambda - (R_a(x) + \int p_a(dx'|x)\breve{V}(\lambda, x')) + (R_b(x) + \int p_b(dx'|x)\breve{V}(\lambda, x'))\Big)^2,$$

over λ, where $\breve{V}$ is a solution of the Poisson equation above for given λ, obtainable by using any linear system solver as a subroutine. (It does not matter which solution, because any constant offset will cancel out.) The foregoing is for a fixed $\tilde{x}$, so one has to consider it for all possible $\tilde{x}$. To highlight this, we write $V = \breve{V}(x, \tilde{x})$ to render explicit the $\tilde{x}$-dependence of the V above.

This linear system is usually very high dimensional, so one resorts to the so called 'linear function approximation', i.e., assume a linearly parametrized form

$$\breve{V}(x, \tilde{x}) \approx \Phi^T r = \sum_{i=1}^m r_i \phi^i(x, \tilde{x}),$$

where $\Phi^T(\cdot) = [\phi^1(\cdot) : \cdots : \phi^m(\cdot)]$, $r = [r_1, \cdots, r_m]^T$. Here $\{\phi^k\}$ can be viewed as maps $S^2 \mapsto \mathbb{R}$ and are the 'features' or 'basis functions' chosen for function approximation. These are kept fixed, with $\{r_i\}$ the corresponding weights. This reduces the search space for any learning algorithm based on this parametrization to $\mathbb{R}^m$. There are many well established algorithms for approximate solution

of the Poisson equation using linear function approximation [6], [16], [19]. Here we follow [19].

Reinforcement learning algorithms use real or simulated trajectories to learn from. We propose a simulation based method. We simulate an i.i.d. process of candidate thresholds $\{\tilde{X}_n\}$ with law κ (say) and a controlled Markov chain $\{X_n\}$ so that the joint transition kernel of the process $(\tilde{X}_n, X_n)$ is

$$I\{x \geq \tilde{x}\}p_a(dx'|x)\kappa(d\tilde{x}) + I\{x < \tilde{x}\}p_b(dx'|x)\kappa(d\tilde{x}).$$

We derive the iterative equations to update r as in [19] using LSPE(0). Define

$$g(x, \tilde{x}, \lambda) := R_a(x)I\{x \geq \tilde{x}\} + (\lambda + R_b(x))I\{x < \tilde{x}\}.$$

We denote by $\Phi(x, \tilde{x})$ the column vector $[\phi^1(x, \tilde{x}), \cdots, \phi^m(x, \tilde{x})]^T$. Consider the temporal differences $\{d_n\}$ defined as

$$\begin{aligned} d_n(m) &= g(X_m, \tilde{X}_m, \lambda) - \beta_m + \Phi(X_{m+1}, \tilde{X}_m)^T r_n - \\ &\quad \Phi(X_m, \tilde{X}_m)^T r_n, \ m \leq n. \end{aligned}$$

Define $\tilde{r}_n$ as

$$\begin{aligned} \tilde{r}_n &= \text{argmin}_{r \in \mathbb{R}^M} \sum_{k=0}^{n} (\Phi(X_k, \tilde{X}_k)^T r \\ &\quad - \Phi(X_k, \tilde{X}_k)^T r_n - d_n(k))^2. \end{aligned} \tag{9}$$

We can find a solution to the above least mean square problem by using the iteration:

$$r_{n+1} = r_n + c(n)(\tilde{r}_n - r_n), \tag{10}$$

where $\tilde{r}_n$ is the optimizer in (9) and $\{c(n)\}$ is a stepsize sequence satisfying

$$c(n) > 0, \ \sum_n c(n) = \infty, \ \sum_n c(n)^2 < \infty.$$

It is easily verified that

$$\tilde{r}_n = r_n + \tilde{B}_n^{-1}(\tilde{A}_n r_n + \tilde{b}_n),$$

where

$$\tilde{B}_n = \frac{B_n}{n+1}, \ \tilde{A}_n = \frac{A_n}{n+1}, \ \tilde{b}_n = \frac{b_n}{n+1},$$

with

$$\begin{aligned} B_n &= \sum_{k=0}^{n} \Phi(X_k, \tilde{X}_k)\Phi(X_k, \tilde{X}_k)^T, \\ A_n &= \sum_{k=0}^{n} \Phi(X_k, \tilde{X}_k)(\Phi(X_{k+1}, \tilde{X}_k)^T - \Phi(X_k, \tilde{X}_k)^T), \\ b_n &= \sum_{k=0}^{n} \Phi(X_k, \tilde{X}_k)(g(X_k, \tilde{X}_k, \lambda_k) - \beta_n), \\ \beta_n &= \frac{1}{n+1} \sum_{k=0}^{n} g(X_k, \tilde{X}_k, \lambda_k). \end{aligned}$$

These can be computed recursively [19]. The sequence $\{\lambda_n\}$ is given by a second iteration we describe next.

We now apply function approximation to the Whittle index. Let

$$\lambda(x) = \Psi^T y$$

Here $\Psi^T(\cdot) = [\psi_1(\cdot), ..., \psi_K(\cdot)]$, $\psi_i : S \mapsto \mathcal{R}$, where $\{\psi_i(\cdot)\}$ represent K features for the Whittle index. We thus want

$$\begin{aligned} \Psi^T(x)y \approx R_a(x) + \int \Phi^T(x', x) r p_a(dx'|x) - \\ R_b(x) - \int \Phi^T(x', x) r p_b(dx'|x). \end{aligned}$$

We use the following learning scheme to minimize the mean square error between the right and left hand sides:

$$\begin{aligned} y_{n+1} &= y_n - a(n)\Psi(X_n)\Big(\Psi(X_n)^T y_n - \\ &\quad (R_a(X_n) - R_b(X_n)) - (\Phi^T(X'_{n+1}, X_n) r_n \\ &\quad - \Phi^T(X''_{n+1}, X_n) r_n)\Big), \end{aligned} \tag{11}$$

where $X'_{n+1} \approx p_a(\cdot|X_n)$, $X''_{n+1} \approx p_b(\cdot|X_n)$ and $X_{n+1} = X'_{n+1}$ if $X_n \geq \tilde{X}_n$ and $= X''_{n+1}$ otherwise. Here $\{a(n)\}$ are positive stepsizes satisfying:

$$\sum_n a(n) = \infty, \ \sum_n a(n)^2 < \infty, \ \frac{a(n)}{c(n)} \stackrel{n\uparrow\infty}{\to} 0.$$

We set $\lambda_n := \Psi^T(X_n)y_n$ for use in (10).

The theoretical analysis of the coupled scheme is as follows. We introduce the following additional notation.

- $T :=$ the operator $C_b(S) \mapsto C_b(S)$ ($:=$ the space of bounded continuous functions $S \mapsto \mathcal{R}$) given by

$$x \mapsto Tx := Px + \bar{g} - \beta e \tag{12}$$

  for $P :=$ the transition operator of the chain $(X_n, \tilde{X}_n)$, $\bar{g} = \bar{g}(y) := g(x, \tilde{x}, \Psi y)$ with y treated as a fixed parameter, $\beta^* = \beta^*(y) :=$ the stationary expectation of $g(X_n, \tilde{X}_n, \Psi y)$, and $e :=$ the constant function $\equiv 1$. Thus in particular, the Poisson equation reads $V = TV$, i.e., V is a fixed point of T.
- Letting $\pi = \pi(dx, d\tilde{x})$ denote the stationary distribution of the chain $\{(X_n, \tilde{X}_n)\}$ and $\Pi :=$ the projection operator w.r.t. the weighted norm $\|z\|_D := \Big(\int \pi(dx, d\tilde{x})|z_{(x,\tilde{x})}|^2\Big)^{\frac{1}{2}}$ given by

$$\begin{aligned} (\Pi f)(x, \tilde{x}) &:= \Phi^T(x, \tilde{x}) \times \\ &\Big(\int \Phi(\breve{x}', \tilde{x}')\Phi^T(\breve{x}', \tilde{x}')\pi(d\breve{x}', d\tilde{x}')\Big)^{-1} \\ &\times \int \Phi(\breve{x}'', \tilde{x}'')f(\breve{x}'', \tilde{x}'')\pi(d\breve{x}'', d\tilde{x}''). \end{aligned}$$

The reinforcement learning scheme of [19] is designed to find a solution of $\Phi^T r = \Pi T \Phi^T r$ (in fact, for a broad class of T that subsumes the above choice). This has the interpretation of $\Phi^T r^*$ as a legitimate approximation of V if we restrict ourselves to Range(Φ^T). See [19] for details.

In view of the condition $a(n) = o(c(n))$, the coupled iteration is a *two time scale* stochastic approximation algorithm analyzed in [4] and Chapter 6 of [5]. Using the analysis therein, we treat the second iteration as quasi-static

while analyzing the first iteration. That is, view $y_n \approx$ a constant y. Then (10) is simply the scheme of [19]. As in [19], one can prove that:

Lemma 1 For $y_n \equiv y$, $r_n \to r^*$ where r^* is the unique solution to $\Phi^T r^* = \Pi T \Phi^T r^*$.

Write $r^* = r^*(y)$ to make its y-dependence explicit. In the 'two time scale' framework, this leads to [4]:

Corollary 1 Almost surely, $r_n - r^*(y_n) \to 0$.

In turn, this allows us to analyze (11) with $r^*(y_n)$ replacing r_n with vanishing error. Let $\pi_1(dx) := \pi(dx, S)$ and define $F : \mathcal{R}^m \mapsto \mathcal{R}$ by:

$$F(z,r) := \int \pi_1(dx)\Big(\Psi^T(x)z - (R_a(x) - \ R_b(x) + \int p_a(dx'|x)\Phi^T(x',x)r - \int p_b(dx'|x)\Phi^T(x',x)r)\Big)^2,$$

for $z \in \mathcal{R}^m$. Then standard theory surrounding stochastic approximation schemes (see, e.g., Section 10.2 of [5]) leads to:

Lemma 2 If $\{y_n\}$ converge a.s., they converge to the unique minimizer of F.

Proof Treating $\{X_n\}$ as 'Markov noise' as in [5], Chapter 6, (11) has the o.d.e. limit

$$\begin{aligned}\dot{w}(t) \ = \ & - \int \pi_1(dx)\Psi(x)\Big(\Psi(x)w(t) - (R_a(x) - \ R_b(x) \\ & + \int p_a(dx'|x)\Phi^T(x',x)r^*(w(t)) \\ & - \int p_b(dx'|x)\Phi^T(x',x)r^*(w(t)))\Big).\end{aligned}$$

The equiibrium point of this o.d.e. is precisely the unique minimizer of F. The claim follows from standard facts about stochastic approximation algorithms and their o.d.e. limits (see, e.g., section 10.2, [5]). □

Note that we do not claim a.s. convergence. This is because the above scheme is not an exact stochastic gradient scheme by virtue of having ignored the y-dependence of r^*, which turns out to be extremely messy to incorporate. Such simplifications are common in reinforcement learning literature and our numerical experiments so far justify them as they do show convergent behavior.

It should be underscored that this learning scheme is 'raw' insofar as the threshold variable is concerned. This is because it assumes no a priori knowledge about this variable. This is why the threshold variable is sampled i.i.d. In many cases, there is some prior knowledge to suggest the most likely range of the threshold and this can be used to advantage in reducing the search space and speeding up the computation. We shall see one example of this in the next section where an explicitly known index for the simpler deterministic (i.e., averaged) model carries useful information. Another instance is when deterministic models arrived at by taking fluid limits are amenable to explicit index computation [8]. The only difference in the foregoing analysis made by thus incorporating prior information is that the law $\kappa(dx')$ of IID random variables $\{\tilde{X}_n\}$ gets replaced by a conditional probability kernel $\kappa(dx'|x) :=$ the conditional law of $\tilde{X}_n$ given $X_n = x$ for any $n \geq 0$. As long as the combined Markov chain $\{(X_n, \tilde{X}_n)\}$ remains positive recurrent, the above analysis goes through. Note that $\{X_n\}$ remains a Markov chain in its own right. Also, it is possible that the range of $\tilde{X}_n$ may no longer be the whole of S, but that does not cause a problem, one can simply restrict the combined state space of $(X_n, \tilde{X}_n)$ to a suitable subset of S^2.

IV. Example: crawlers for ephemeral content

A. The algorithm

Ephemeral content is web content of short-lived interest. One standard model for the interest dynamics (based on click rate data and validated) is described as follows [2]. Let $X^i(n) :=$ interest in the 'web content' at location i at time n, $1 \leq i \leq N$, $\alpha^i \in (0,1)$ decay rate of 'interest', and $u^i :=$ mean arrival rate of 'content' per epoch. Then the dynamics is:

$$X^i(n+1) = \alpha^i X^i(n) + u^i \text{ if not crawled,} \quad (13)$$

$$X^i(n+1) = u^i \text{ if crawled.} \quad (14)$$

The control process is $\{v^i(t)\}$ where $v^i(t) = 1$ if ith site crawled, 0 otherwise. This is to be chosen so as to maximize the average reward

$$\limsup_{t\uparrow\infty} \sum_{i=1}^{N} \frac{1}{t} \sum_{m=0}^{t} X^i(m)v^i(m)$$

subject to

$$\lim_{t\uparrow\infty} \sum_{i=1}^{N} \frac{1}{t} \sum_{m=0}^{t} v^i(m) = M.$$

The latter constraint is the Whittle relaxation of the per stage constraint

$$\sum_{i=1}^{N} v^i(n) = M \quad \forall n.$$

One can establish Whittle indexability via the existence of an optimal threshold policy that is monotone increasing in λ [2]. Let

$$\zeta^i(x) := \left\lceil \log^+_{\alpha^i}\left(\frac{u^i - (1-\alpha^i)x}{u^i}\right)\right\rceil.$$

Then the Whittle index is

$$\begin{aligned}\lambda^i(x) \ := \ & \zeta^i(x)((1-\alpha^i)x - u^i) + \\ & \Big[(\alpha^i)^{\zeta^i(x)} + \Big(\frac{1-(\alpha^i)^{\zeta^i(x)}}{1-\alpha^i}\Big)\Big]u^i.\end{aligned}$$

In the fully stochastic case, we replace u^i in (13)-(14) by i.i.d. random variables $\{U^i_n\}$ with law (say) φ. A concrete scenario about how these come by is described in the next

section. One can again establish Whittle indexability via the existence of an optimal threshold policy that is monotone increasing in λ [2]. However, the explicit expression for the Whittle index proves elusive. The Poisson equation (7)-(8) for this case becomes

$$\begin{aligned} V(x,\tilde{x})+\beta &= I\{x<\tilde{x}\}\left(\lambda+\int V(\alpha x+x',\tilde{x})\varphi(dx')\right) \\ &\quad + I\{x\geq\tilde{x}\}\left(x+\int V(x',\tilde{x})\varphi(dx')\right). \end{aligned}$$

As before, we set $V \approx \Phi r$ and $\lambda \approx \Psi^T y$. Then the reinforcement learning sheme for $\{r_n\}$ is exactly as above (section III) with $R_a(x)=x$ and $R_b(x)\equiv 0$. For the Whittle index, the learning scheme takes the specific form

$$\begin{aligned} y_{n+1} &= y_n - a(n)\Big(\Psi(X_n)^T y_n - X_n - V_n(U_{n+1},X_n) \\ &\quad + V_n(\alpha X_n + U_{n+1},X_n)\Big)\Psi(X_n) \end{aligned}$$

where $V_n := \Phi^T r_n$ and U_{n+1} is the new utility observed.

B. Numerical experiments

Let us demonstrate the theoretical results by a small numerical example. We use the model as specified in [2]. Without loss of generality, let the discrete time instants for crawling be $nT,\ n\geq 0$, for some $T>0$. The content at source $i\in\{1,\ldots,N\}$ is published at times τ_n^i with an initial utility ($\approx$ interest level) ξ_n^i. The $\{\xi_n^i\}$ are assumed to be i.i.d. exponential with parameter θ^i. The utility decreases exponentially over time with a deterministic rate μ_i. Further, it is assumed that new content at source i arrives according to a time homogeneous Poisson process with rate Λ_i. At the end of the interval $[nT,(n+1)T]$, the new utility seen is

$$U_{n+1} := \sum_{\{n:\tau_n\in(nT,(n+1)T]\}} \xi_n^i e^{-\mu_i((n+1)T-\tau_n)}.$$

For our numerical experiments, we have taken $N=4$. We run the system with parameters specified in Table 1.

TABLE I
DATA FOR NUMERICAL EXAMPLES

i	1	2	3	4
θ_i	1	0.7	0.2	0.08
μ_i	0.7	0.35	0.7	0.21
Λ_i	25	25	25	25

We first check the convergence of our algorithm provided for the stochastic case. For the value function, we use the following basis functions for function approximation:

$$\Phi = [1\ \ x\ \ \tilde{x}\ \ x^2\ \ \tilde{x}^2\ \ x\tilde{x}]$$

where x represents the state and $\tilde{x}$ the threshold. In [2], good performance for the crawler was reported when the deterministic index was used in the stochastic case. Motivated by this we choose our function approximation for the Whittle index to be a perturbation around the deterministic Whittle index, the closed form expression for which has been derived in [2] and is given above. Let $\gamma(x)$ represent the stochastic index and $\gamma^*(x)$ represent the deterministic index. The following is the approximation used:

$$\gamma(x)=\gamma^*(x)+\Psi^T(x)y,\ \ arg(x)=\frac{\pi(x-u)(1-\alpha)}{\alpha u}$$
$$\Psi_i = k\gamma^*(x)[cos(i*arg(x))]$$

where i=1,2,...6, and u is the expected utility per epoch as before. Next, we apply a projection on the perturbation term $\Psi^T y$ in order to restrict it to be within a pre-specified fraction p away from the deterministic value. The update equation is given by:

$$\begin{aligned} \tilde{y}_{n+1} &= y_n - a(n)(\gamma^*(x)+\Psi(X_n)'y - X_n - \\ &\quad V_n(U_{n+1},X_n)+V_n(\alpha X_n+U_{n+1},X_n))\Psi(X_n), \\ y_{n+1} &= \Gamma\tilde{y}_{n+1}. \end{aligned}$$

Here Γ projects the updated y value so that $|\Psi' y|\leq|p\gamma^*(x)|$.

To implement this projection the following quadratic program is used:

$$\min_y ||y-\tilde{y}||_2 \ \text{ s.t. } \ |\Psi' y|\leq|p\gamma^*(x)|.$$

The plots shown in Fig. 1 and Fig. 2 show the convergence for weights of agents with and without the above projection respectively. In both the plots, a burn-in period of 34000 iterations has been removed. In both of them, different colours indicate the different co-ordinates of the weight vector. Also, the projection operator was used only of the order 10^2 times out of the 10^5 iterates shown. Once projected down, the iterates have a tendency to remain in the projection region for a long time before a jump occurs due to randomness in the arrival process. We observe that in Fig. 1, jumps are like spikes and the iterate values are immediately projected back whereas, in Fig. 2, the effect of a jump dies down smoothly due to absence of projection.

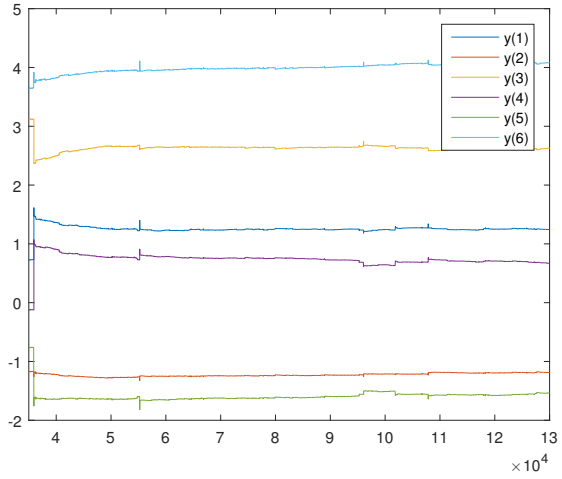

Fig. 1. Convergence of weights of Function Approximation for the Policy with Projection

The results obtained are shown in Table II. Five policies were tested for two different cases averaged over 10000

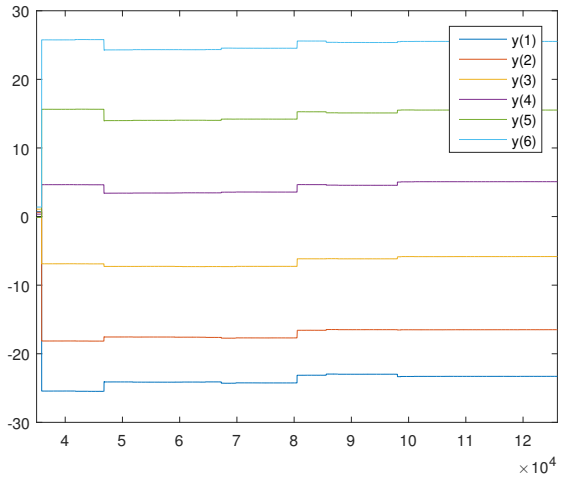

Fig. 2. Convergence of weights of Function Approximation for the Policy without Projection

iterations for comparison. The cases were distinguished by number of agents allowed to be active in each epoch. The different policies used include the Round Robin and the Greedy Policy. The greedy policy used was same as the one mentioned in [2], which chooses the sources maximising

$$\frac{\Lambda_i \bar{\xi}_i}{\mu_i}(1 - exp(-\mu_i \times t_i))$$

where $\bar{\xi}_i = E\left[\xi_n^i\right]$ and t_i = the lapsed time since the last crawl for source i. For the Deterministic, Stochastic without Projection and Stochastic with Projection policies, the index is used as derived in the above theory. From Table

TABLE II

Policy	β (M = 1)	β (M = 2)
Round Robin	208.13	281.53
Greedy	232.47	322.72
Deterministic	259.61	328.44
Stochastic without Projection	241.77	307.05
Stochastic with Projection	258.42	333.36

II, we observe that in the case of only one agent being active per epoch, the deterministic policy outperforms all others but the stochastic policy with projection is very close. All other policies show significant difference in the average reward. When we increase the number of agents active per epoch to 2, we see that the stochastic policy with projection performs better than the deterministic policy. Again there is a significant difference in the stochastic policy with and without projection showing the importance of projection. That said, the gain is small, showing that the exact index for the deterministic approximation was indeed a good policy even for the stochastic case. In future, we plan to test the scheme on other examples.

V. Conclusion

We have proposed and analyzed a reinforcement learning algorithm using linear function approximation for learning the Whittle indices associated with a class of indexable restless bandits. The scheme was successfully applied to an example of restless bandits arising from the problem of scheduling web crawlers for ephemeral content where an exact expression for the indices is known for a deterministic approximation, but not for the full stochastic problem. We use this information to advantage to reduce the search space. This is illustrative of how the methodology could be used in a variety of restless bandit problems.

References

[1] M. Agarwal, V. S. Borkar and A. Karandikar, "Structural properties of optimal transmission policies over a randomly varying channel", *I*EEE Trans. Automatic Control 53(6), 2008, 1476-1491.

[2] K. E. Avrachenkov and V. S. Borkar, "Whittle index policy for crawling ephemeral content", *I*EEE Trans. Control of Network Systems, to appear, 2017. (available at http://ieeexplore.ieee.org/abstract/document/7593334/

[3] D. P. Bertsekas, *Dynamic Programming and Optimal Control Vol. 2* (4th ed.), Athena Scientific, Belmont, Mass., 2012.

[4] V. S. Borkar, "Stochastic approximation with two time scales", *Systems & Control Letters* 29(5), 1997, 291-294.

[5] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*, Hindustan Publishing Agency, New Delhi, and Cambridge Uni. Press, Cambridge, UK, 2008.

[6] V. S. Borkar, "Reinforcement learning: a bridge between numerical methods and Monte Carlo", in *Perspectives in Mathematical Science I: Probability and Statistics* (N. S. N. Sastry, T. S. S. R. K. Rao, M. Delampady, B. Rajeev, eds.), World Scientific, Singapore, 2009, 71-91.

[7] P. Jacko, *Dynamic Priority Allocation in Restless Bandit Models*, Lambert Academic Publishing, 2010.

[8] M. Larrañaga, U. Ayesta and I. M. Verloop, "Index policies for a multi-class queue with convex holding cost and abandonments", In *Proc. ACM SIGMETRICS Performance Evaluation Review* 42(1), 2014, 125-137.

[9] K. Liu and Q. Zhao, "Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access", *IEEE Trans. Info. Theory* 56(11), 2010, 5547-5567.

[10] J. Nino-Mora and S. S. Villar, "Sensor scheduling for hunting elusive hiding targets via Whittle's restless bandit index policy", *Proc. NetGCoop 2011*, 12-14 Oct., 2011, Paris, 1-8.

[11] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queuing network control", *Math. Op. Research* 24(2), (1999), 293-305.

[12] M. I. Puterman, *Markov Decision Processes*, John Wiley and Sons, Hoboken, New Jersey, 1994.

[13] V. Raghunathan, V. S. Borkar, M. Cao and P. R. Kumar, "Index policies for real-time multicast scheduling for wireless broadcast systems", *Proc. IEEE INFOCOM 2008* 13-18 April 2008, Phoenix, 2243-2251.

[14] R. K. Sundaram, *A First Course in Optimization Theory*, Cambridge Uni. Press, Cambridge, UK, 1996.

[15] M. Shaked and J. G. Shanthikumar, *Stochastic Orders and Their Applications*, Academic Press, Boston, 1994.

[16] J. N. Tsitsiklis and B. Van Roy, "Average cost temporal-difference learning", *Automatica* 35(11), 1999, 1799-1808.

[17] R. R. Weber and G. Weiss, "On an index policy for restless bandits", *J. of App. Prob.* 27(3), 1990, 637-648.

[18] P. Whittle, "Restless bandits: activity allocation in a changing world", *J. Appl. Prob.* 25.A, 1988, 287-298.

[19] H. Yu and D. P. Bertsekas, "Convergence results for some temporal difference methods based on least squares", *IEEE Trans. Automatic Control* 54(7), 2009, 1515-1531.