Reinforcement Learning Methods for Weakly Coupled MDPs

Konstantin Avrachenkov (Inria)

based on joint works with U. Ayesta, V. Borkar, F. Robledo

GDR COSMOS Workshop, Grenoble, 21/11/2023
Let us start with the classical Restless MABs and then explore possible generalizations to Weakly-coupled MDPs.
Consider \( N > 1 \) controlled Markov chains (‘arms’) \( \{X^i_n, n \geq 0\}, \ 1 \leq i \leq N \), on a finite discrete state space \( S \).
Consider $N > 1$ controlled Markov chains (‘arms’) 
$\{X_n^i, n \geq 0\}$, $1 \leq i \leq N$, on a finite discrete state space $S$.

**Control** or **action** $A_n$ is binary:

- $A_n^i = 0$ – the $i$-th arm is **passive**;
- $A_n^i = 1$ – the $i$-th arm is **active**.
Consider $N > 1$ controlled Markov chains (‘arms’) 
$\{X^i_n, n \geq 0\}, \ 1 \leq i \leq N$, on a finite discrete state space $S$.

**Control or action** $A_n$ is binary:

- $A^i_n = 0$ – the $i$-th arm is passive;
- $A^i_n = 1$ – the $i$-th arm is active.

System dynamics is defined by controlled transition kernels:

$$(k, j, a) \in S^2 \times \{0, 1\} \mapsto p^i(j|k, a) \in [0, 1].$$
Restless MABs and Whittle index

Let $R^i(x, a) : S \times A \mapsto [0, \infty)$, $a = 0$, resp. 1, denote per stage reward for passive, resp. active, mode for arm $i$. The objective is to maximize the expected discounted reward

$$V^\pi(x_1, \ldots, x_N) = \max_{\pi} E \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t R^i(X^i_t, A^i_t) | X^i_0 = x^i \right] ,$$

subject to the constraint, for prescribed $M < N$,

$$\sum_{i=1}^{N} A^i_t = M, \quad \forall t$$

I.e., at each time instant, only $M$ arms are activated.
Let $R^i(x, a) : S \times A \mapsto [0, \infty)$, $a = 0$, resp. 1, denote per stage reward for passive, resp. active, mode for arm $i$.

The objective is to maximize the expected discounted reward

$$V^{\pi^*}(x^1, \ldots, x^N) = \max_{\pi} \mathbb{E}\left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t R^i(X^i_t, A^i_t) \mid X^i_0 = x^i \right],$$

subject to the constraint, for prescribed $M < N$,

$$\sum_{i=1}^{N} A^i_t = M, \forall t$$

i.e., at each time instant, only $M$ arms are activated.
Restless MABs and Whittle index

RMAB problem is provably hard, PSPACE-complete, (Papadimitriou & Tsitsiklis, 1999).

Whittle’s ingenious observation was to replace the ‘hard constraint’ (2) by the ‘time-averaged constraint’:

\[
E \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t A^i_t \right] = \frac{M}{1 - \gamma},
\]

(3)

which renders the problem to the separable form and allows one to use the technique of Lagrange multiplier.
Restless MABs and Whittle index

$$\max_{\pi} E \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t \left( R^i (X^i_n, A^i_n) + \lambda (1 - A^i_n) \right) \right]$$

The Lagrange multiplier technique leads to the following DP equation for each arm:

$$V^i(x) = \max_{a \in \{0,1\}} \left( a(r^i(x, 1) + \gamma \sum_y p^i(y|x, 1) V^i(y)) + (4) \right)$$

$$(1 - a)(r^i(x, 0) + \lambda + \gamma \sum_y p^i(y|x, 0) V^i(y))$$,

with $V^i(x)$ the unknown variables.
Restless MABs and Whittle index

One can view the Lagrange multiplier \( \lambda \) as a ‘subsidy’ for passivity.

RMAB is said to be **indexable** if the set of passive state increases monotonically from the empty set to all of \( S \) as the subsidy is increased from \(-\infty\) to \(\infty\).

In this case, the Whittle index is defined to be the value \( \lambda^*(k) \) of \( \lambda \) for which both active and passive modes are equally preferred in the state \( k \). That is,

\[
\lambda^*(k) + r(k, 0) + \sum_y p(y|k, 0)V(y) = r(k, 1) + \sum_y p(y|k, 1)V(y).
\]
The Whittle index policy enjoys many good properties and performs very well in numerous applications.
The Whittle index policy enjoys many good properties and performs very well in numerous applications.

However, it requires the full model knowledge...
Q-learning (Watkins, 1988) is the most prominent reinforcement learning technique designed to mitigate model uncertainty.

The technique is based on stochastic approximation solution of the DP equation for Q-values:

\[
Q^i(x, a) = a \left( r^i(x, 1) + \gamma \sum_y p^i(y|x, 1) \max_{b \in \{0,1\}} Q^i(y, b) \right) \\
+ (1-a) \left( r^i(s, 0) + \lambda + \gamma \sum_y p^i(y|x, 0) \max_{b \in \{0,1\}} Q^i(y, b) \right) 
\] (6)
Whittle index based Q-learning

Fix stepsize sequence satisfying Robbins-Monro conditions:

\[ \sum_{n} \alpha(n) = \infty \text{ and } \sum_{n} \alpha(n)^2 < \infty. \]

For each \( x \in S \), \( a \in \{0, 1\} \), and the reference state \( \hat{k} \in S \), do:

\[
Q_{n+1}(x, a; \hat{k}) = Q_n(x, a; \hat{k}) + \alpha(\nu(x, a, n)) \times \\
I\{X_n = x, U_n = a\} \left( (1 - a)(r(x, 0) + \lambda_n(\hat{k})) + ur(x, 1) \right. \\
+ \left. \gamma \max_{b \in \mathcal{U}} Q_n(X_{n+1}, b; \hat{k}) - Q_n(x, a; \hat{k}) \right)
\]

(7)

where \( \lambda_n(\hat{k}) \) is an estimate of the Whittle index for state \( \hat{k} \), and where the ‘local clock’ for the pair \( (x, a) \) is given by

\[
\nu(x, a, n) = \sum_{m=0}^{n} I\{X_m = x, Z_m = a\}, \ x \in S, a \in \{0, 1\}. \]
Let us now Q-learn Whittle index!

Note that in the context of Q-learning, we need to solve (5) in the form

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

for $\lambda = \lambda(\hat{k})$. 
Whittle index based Q-learning

For the second ingredient, we can solve (8) by

\[ \lambda_{n+1}(\hat{k}) = \lambda_n(\hat{k}) + \beta(n) \left( Q_n(\hat{k}, 1; \hat{k}) - Q_n(\hat{k}, 0; \hat{k}) \right), \]  

where the stepsize sequence \( \{\beta(n)\} \) satisfies

\[ \sum_n \beta(n) = \infty, \sum_n \beta(n)^2 < \infty \]  

and \( \beta(n) = o(\alpha(n)) \).
Whittle index based Q-learning

Note that both off-policy as well as on-policy modes are possible.

In the on-policy mode, the control actions at time $n$ are defined as follows:

- with probability $(1 - \epsilon)$, we sort arms in the decreasing order of the estimated Whittle index $\lambda_n(X_n^i)$ and render the top $M$ arms active;
- the remaining arms are passive;

- with probability $\epsilon$, we render active $M$ random arms, chosen uniformly and independently;
- the remaining arms are passive.
Theorem Given that the problem satisfies the indexability condition, iterations (7) and (9) converge respectively to Q-values of the Whittle index policy, denoted by $Q_W(x, a)$, and to the Whittle indices $\lambda(x)$, i.e.,

$$
\lambda_n(x) \to \lambda(x) \quad \text{and} \quad Q_n(x, a) \to Q_W(x, a)
$$

a.s. $\forall x \in S, a \in A$ as $n \to \infty$.

Proof main ingredient: It is based on two time scale stochastic approximation and as often the case in stochastic approximation establishing the stability of the iterates is the most tricky part.
Let us illustrate the algorithm by an example.

**Example with circulant dynamics (Fu et al, 2019)**

\[
P_0 = \begin{bmatrix}
\frac{1}{2} & 0 & 0 & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}, \quad \text{and} \quad P_1 = P_0^T.
\]

The rewards do not depend on the action:
\(R(1) = -1,\) \(R(2) = 0,\) \(R(3) = 0,\) and \(R(4) = 1.\)

The exact values of the Whittle indices, calculated in (Fu et al, 2019), are as follows:
\(\lambda(1) = -\frac{1}{2},\) \(\lambda(2) = \frac{1}{2},\) \(\lambda(3) = 1,\) and \(\lambda(4) = -1.\)
Q-learning Whittle index

Let us consider a scenario with $N = 100$ identical arms, out of which $M = 20$ are active at each time. We initialize our algorithm with $\lambda_0(x) = 0$, and $Q(y, a; x) = R(y, a)$, $\forall x, y \in S$. We took $\epsilon = 0.1$.

Figure: Estimated (solid lines) and exact (dash lines) Whittle indices in the example with circulant dynamics.
What is a possible issue with Q-learning Whittle index?
What is a possible issue with Q-learning Whittle index?

Memory complexity.
Whittle index based DQ-learning

To mitigate the issue with memory complexity, we suggest to use Deep Q-learning instead of Tabular Q-learning.

Q-table is replaced with Q-network.

Specifically, we maintain the distinction between the visited state $x$ and the reference state $\hat{k}$ of the Whittle index $\lambda(\hat{k})$, so that Q-values become

$$Q^k_\theta(x) = \begin{bmatrix} Q^k_\theta(x, 0) & Q^k_\theta(x, 1) \end{bmatrix}.$$ 

This makes the two state variables $x$ and $\hat{k}$ the inputs of NN, while the outputs are the Q-values for both possible actions.
Whittle index based DQ-learning

DQN tunes the network weights by minimizing the expected Bellman error:

$$\mathcal{E}(\theta, \theta') := \mathbb{E} \left[ \left\| Q_\theta(x, a; \hat{k}) - Q_{\text{target}}(x, a; \hat{k}) \right\|^2 \right],$$

with the target network given by

$$Q_{\text{target}}(x, a; \hat{k}) = (1 - a)(r(x, 0) + \lambda_n(\hat{k})) + ar_1(x, 1) + \gamma \max_{b \in A} Q_{\theta'}^k(X_{n+1}, b; \hat{k}).$$

The target network copies ($\theta \rightarrow \theta'$) the parameter values of the main network $Q_\theta$ e.g. every 50 iterations.
For each state $\hat{k} \in S$, we update Whittle index in a similar way to the tabular implementation, i.e.:

$$\lambda_{n+1}(\hat{k}) = \lambda_n(\hat{k}) + \beta(n) \left( Q_{\theta,n}^\hat{k}(\hat{k}, 1) - Q_{\theta,n}^\hat{k}(\hat{k}, 0) \right), \quad (10)$$

where $\beta(n)$ are time steps of the slow time scale.
Let us compare the approaches on the circulant example with $N = 100$ arms, out of which $M = 70$ need to be activated.

However, we increase the number of states to 50 and with only the first and last states having non-zero rewards ($-1, +1$).
One natural generalization is to go from binary actions to more complex action spaces and non-linear costs.

This will change the constraint

$$\sum_{i=1}^{N} A_t^i = M, \quad A_t^i \in \{0, 1\},$$

to

$$\sum_{i=1}^{N} c^i(A_t^i) \leq \bar{c},$$

with $A_t^i$ belonging to a more complex space.
Weakly-coupled MDPs

If the action space is finite, one way to proceed would be to try to extend the approach of Whittle index.

And indeed, some attempts have been made:

▶ (Weber, 2007)
▶ (Glazebrook, Hodge and Kirkbride, 2011)
▶ (Hodge and Glazebrook, 2015)
▶ (Killian et al, 2021)
▶ (Niño-Mora, 2022)

However, some restrictive technical assumptions are needed and the derived Q-learning approach has shown numerical instabilities.
Weakly-coupled MDPs

(Hawkins, 2003), with refinements by (Killian et al, 2021), proposed Knapsack Lagrangian decomposition approach.

\[ V(x) = \max_{a: \sum_i c(a_i) \leq \bar{c}} \left\{ \sum_i r^i(x_i, a_i) + \gamma \sum_y p(y|x, a) V(y) \right\} \]

Use the Lagrange multiplier:

\[ V(x) = \max_a \left\{ \sum_i r^i(x_i, a_i) + \lambda \left( \bar{c} - \sum_i c^i(a_i) \right) + \gamma \sum_y p(y|x, a) V(y) \right\} \]
Weakly-coupled MDPs

Then, we can assume that

\[ V(x) = \sum_{i=1}^{N} V^i(x_i) \]

which leads to the decomposition formulation

\[ Q^i(x_i, a_i, \lambda) = r^i(x_i, a_i) - \lambda c^i(a_i) + \gamma \sum_{y_i} p(y_i|x_i, a_i) \max_{b_i} Q^i(y_i, b_i, \lambda), \]

\[ \lambda^* = \arg \min_{\lambda \geq 0} \left\{ \sum_{i=1}^{N} \max_{a_i} Q(x_i, a_i, \lambda) + \frac{\lambda \bar{c}}{1 - \gamma} \right\}. \quad (11) \]
Weakly-coupled MDPs

The decomposition formulation can be used to elaborate online reinforcement learning method:

On the fast time scale, we learn Q-values e.g. by DQN (Q-values approximated by NN with three inputs $x$, $a$ and $\lambda$).

On the slow time scale, we solve easy, 1-dim, optimization problem for $\lambda^*$. 

Finally, we force the constraint satisfaction with the Knapsack-like problem:

$$\max_a \sum_{i=1}^N Q^i(x_i(t), a_i, \lambda^*)$$

$$\sum_{i=1}^N c^i(a_i) \leq \bar{c}.$$
Let us consider numerical examples with continuous actions:

**Type A:** \( S = \{0, 1\}, \ a \in [0, 2], \ r(x, a) = x, \ c(a) = a. \)

\[
P(a) = \begin{bmatrix}
0.02a^2 - 0.09a + 0.8 & -0.02a^2 + 0.09a + 0.2 \\
0.75 \exp(-0.947a) & 1 - 0.75 \exp(-0.947a)
\end{bmatrix}
\]
Weakly-coupled MDPs

**Type B:** \( S = \{0, 1\}, \ a \in [0, 2], \ r(x, a) = x, \ c(a) = a. \)

\[
P(a) = \begin{bmatrix}
0.95 \exp(-2.235a) & 1 - 0.95 \exp(-2.235a) \\
0.3347 \exp(-1.609a) & 1 - 0.3347 \exp(-1.609a)
\end{bmatrix}
\]
Thank you!

Questions? k.avrachenkov@inria.fr
Based on the works:

Avrachenkov, K., & Borkar, V.S. Whittle Index Based Q-learning for Restless Bandits with Average Reward. *Automatica*, v.139, 110186, 2022.


Background references:


