Whittle Index-based Q-learning using FGDQN

Tejas Pagare

March 10, 2022

Derivation

As we have seen, whittle index is equivalent to solving for $\lambda(\hat{k})$ the following equation

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

Substituting $Q(\hat{k},0)$ we get

$$Q(\hat{k},1) = r(\hat{k},0) + \lambda(\hat{k}) - \rho + \sum_y p(y|\hat{k},0) \max_{v\in\{0,1\}} Q(y,v)$$

which is equivalent to

$$\lambda(\hat{k}) = Q(\hat{k},1) - r(\hat{k},0) + \rho - \sum_y p(y|\hat{k},0) \max_{v\in\{0,1\}} Q(y,v)$$

Now, using stochastic approximation we remove the conditional expectation by a real random variable $\zeta(\hat{k},0)$ with the law $p(\cdot|\hat{k},0)$ and make increment based on our current estimate, which gives us the following λ iteration

$$\lambda_{n+1}(\hat{k}) = (1-b(n))\lambda_n(\hat{k}) + b(n)\Big(Q(\hat{k},1) - r(\hat{k},0) + \rho - \max_{v\in\{0,1\}} Q(\zeta_{n+1}(\hat{k},0),v)\Big)$$

The algorithm follows as below:
At each time instant n, we observe the state X_n of the controlled Markov chain and accordingly update the X_n^{th} component of $\lambda(\cdot)$. We consider a single run $\{X = X_n, U = 0\}$ of the controlled Markov chain which gives X_{n+1}^0 (^{0} is to indicate that we only consider action 0 to get the next state from the state X_n) with the same conditional law of $p(\cdot|X_n,0)$ and hence replaces $\zeta_{n+1}(\hat{k},0)$ for $X_n = \hat{k}$.
$\therefore$ We now have the following iteration

$$\lambda_{n+1}(\hat{k}) = \lambda_n(\hat{k}) + b(n)I\{X_n = \hat{k}\}\Big(Q(\hat{k},1) - r(\hat{k},0) + \rho - \max_{v\in\{0,1\}} Q(X_{n+1}^0,v) - \lambda_n(\hat{k})\Big)$$

$$\sigma_{n+1} = \sigma_n + b(n) \times \Bigg(\Big(\sum_y p(y|X_n,0)(Q(X_n,1) - r(X_n,0) + f(Q) - \max_{v\in\{0,1\}} Q(X_{n+1}^0,v) - \lambda(X_n))\Big) \times \nabla_\sigma \lambda(X_n)\Bigg)$$

$$\sigma_{n+1} = \sigma_n + b(n) \times \Bigg(\overline{\Big(Q(X_n,1) - r(X_n,0) + f(Q) - \max_{v\in\{0,1\}} Q(X_{n+1}^0,v) - \lambda(X_n)\Big)} \times \nabla_\sigma \lambda(X_n)\Bigg)$$

Whittle Iteration:

$$\sigma_{n+1} = \sigma_n + b(n) \times \left(\overline{\Big(Q(X_n, 1) - r(X_n, 0) + f(Q) - \max_{v \in \{0,1\}} Q(X^0_{n+1}, v) - \lambda(X_n) \Big)} \times \nabla_\sigma \lambda(X_n) \right) \tag{1}$$

Q Iteration:

$$\theta_{n+1} = \theta_n - a(n) \Bigg(\overline{(1 - U_n)(r(X_n, 0) + \lambda_n(\hat{k})) + U_n r(X_n, 1) + \max_{v \in \{0,1\}} Q(X_{n+1}, v; \theta_n, \hat{k}) - f(Q(\hat{k}; \theta)) - Q(X_n, U_n; \theta_n, \hat{k})} \times$$
$$\Big(\nabla_\theta Q(X_{n+1}, v_n; \theta_n, \hat{k}) - \nabla_\theta f(Q(\hat{k}; \theta)) - \nabla_\theta Q(X_n, U_n; \theta_n, \hat{k}) \Big) + \xi_{n+1} \Bigg) \tag{2}$$

Algorithm 1: Whittle Indices with FGDQN

Input: replay memory $\mathcal{D}$ of size M, minibatch size B for Q iteration and C for λ iteration, whittle index λ, T number of iterations.
Initialise the weights θ & σ randomly for the Q-Network and Whittle-Network.
Consider RMABP with I projects such that at every time step K of them are active.
Denote state of the system at time n as $S(n) = (s_1(n), \cdots, s_i(n), \cdots, s_I(n))$ where $s_i(n)$ is a state of project $i \in \{1, 2, \cdots, I\}$
for $n = 1$ *to* T **do**
    for $s = 1$ *to* d **do**
        $S(n) = (s, \cdots, s)$
        Select actions $U(n) = (u_1(n), \cdots, u_i(n), \cdots, u_I(n)))$ at random such that $\sum_{i=1}^{I} u_i(n) = K$.
        Execute actions and take a step.
        Observe the rewards $R(n) = (r_1(n), \cdots, r_I(n))$ and obtain next state of the system $S(n+1)$.
        Store all the tuples $(S(n), U(n), R(n), S(n+1))$ in $\mathcal{D}$
    end
    for $s = 1$ *to* d **do**
        for $a = \{0, 1\}$ **do**
            Sample all tuples (X_j, U_j, R_j, X_{j+1}) of size B with a fix state-action pair $(X_j = s, U_j = a)$ from $\mathcal{D}$

$$Set\, Z_j = (1 - U_j)(R_j + \lambda(X_j)) + U_j R_j + \max_v Q(X_{j+1}, v; \theta) - f(Q)$$

            Compute gradients and using Eq. (2) update parameters θ.
        end
    end

    On slower time-scale **do**
    for $s = 1$ *to* d **do**
        Sample all tuples (X_k, U_k, R_k, X_{k+1}) of size C with a fix state-action pair $(X_j = s, U_j = 0)$ from $\mathcal{D}$

$$Set\, Z_k = Q(X_k, 1; \theta_n) - r(X_k, 0) + f(Q) - \max_{v \in \{0,1\}} Q(X_{k+1}, v; \theta_n)$$

        Compute gradients and using Eq. (1) update parameters σ.
    end
end

Consider parametrized families $\lambda(k;\sigma,\omega)$ and $Q(i,u;\theta,\omega)$. Q values are not directly related to whittle index but depends on some parameters on which the whittle index depends.

$$
\begin{aligned}
\theta_{n+1} \ &= \ \theta_n \ - \ a(n)\Big(\nabla_\theta Q(X_{n+1}, v_n; \theta_n, \omega_n) \ - \ \nabla_\theta f(Q(\theta, \omega_n))\Big|_{\theta=\theta_n} \\
&\quad - \nabla_\theta Q(X_n, U_n; \theta_n, \omega_n) \ \Big) \times \\
&\quad \overline{\Big((1-U_n)(r(X_n,0)+\lambda(X_n;\sigma_n,\omega_n)) + U_n r_n(X_n,1) + \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n)} \\
&\quad \overline{- \ f(Q(\theta_n,\omega_n)) - Q(X_n, U_n; \theta_n, \omega_n)\Big)} + a(n)\xi_{n+1}, \\
\sigma_{n+1} &= \sigma_n - b(n)\Big(\overline{Q(X_n,1;\theta_n,\omega_n) - r(X_n,0) + \ f(Q(\theta_n,\omega_n))} \\
&\quad \overline{- \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n)} - \lambda(X_n;\sigma_n,\omega_n)\Big) \times \\
&\quad \Big(- \nabla_\sigma \lambda(X_n;\sigma_n,\omega_n)\Big) \\
\omega_{n+1} &= \omega_n - a(n)\Big(\nabla_\omega Q(X_{n+1}, v_n; \theta_n, \omega_n) - \nabla_\omega f(Q(\theta_n,\omega))\Big|_{\omega=\omega_n} \\
&\quad - \nabla_\omega Q(X_n, U_n; \theta_n, \omega_n)\Big) \times \\
&\quad \overline{\Big((1-U_n)(r(X_n,0)+\lambda(X_n;\sigma_n,\omega_n)) + U_n r_n(X_n,1) + \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n)} \\
&\quad \overline{- \ f(Q(\theta_n,\omega_n)) - Q(X_n, U_n; \theta_n, \omega_n)\Big)} + a(n)\xi_{n+1} \\
&\quad - b(n)\Big(\overline{Q(X_n,1;\theta_n,\omega_n) - r(X_n,0) + \ f(Q(\theta_n,\omega_n))} \\
&\quad \overline{- \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n)} - \lambda(X_n;\sigma_n,\omega_n)\Big) \times \\
&\quad \Big(\nabla_\omega Q(X_{n+1}, v_n; \theta_n, \omega_n) \ - \ \nabla_\omega f(Q(\theta_n,\omega))\Big|_{\omega=\omega_n} \\
&\quad - \ \nabla_\omega Q(X_n, U_n; \theta_n, \omega_n) - \nabla_\omega \lambda(X_n;\sigma_n,\omega_n)\Big)
\end{aligned}
$$

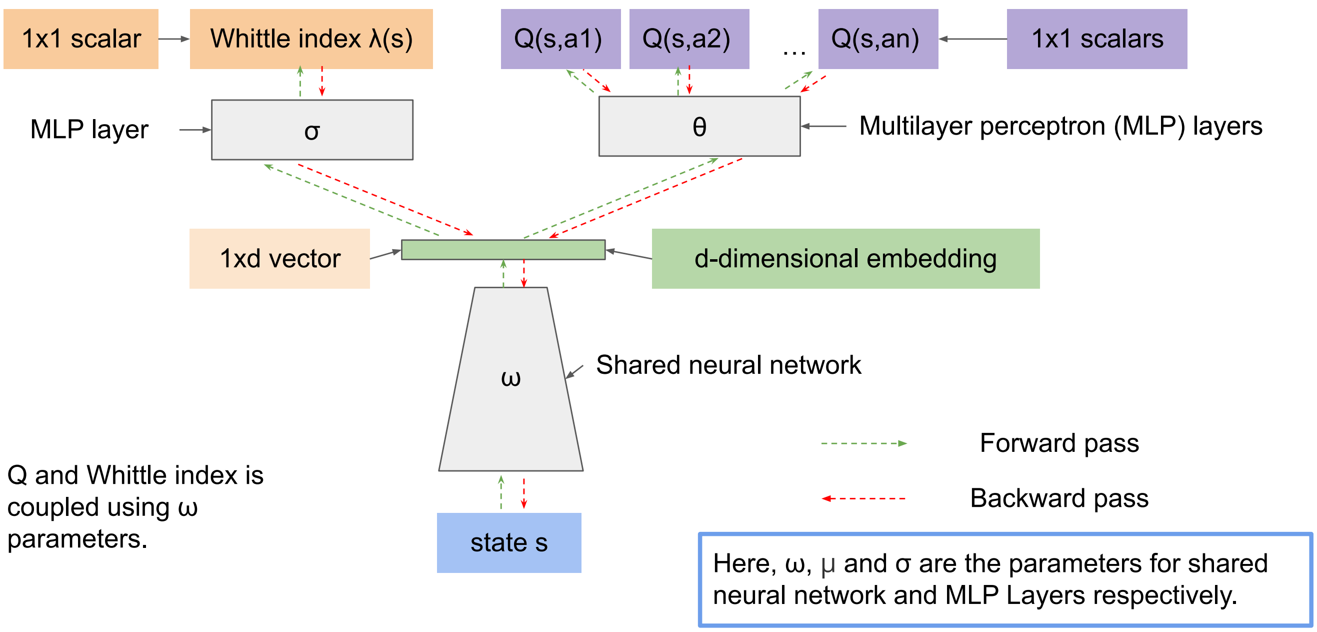
The θ_n iteration is the SGD for the mean square error

$$
\begin{aligned}
\mathcal{E}_1 := E\Big[\Big\|(1-U_n)(r(X_n,0)+\lambda(X_n;\sigma_n,\omega_n)) + U_n r_n(X_n,1) \\
+ \ \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n) - \ f(Q(\theta_n,\omega_n)) - Q(X_n, U_n; \theta_n, \omega_n)\Big\|^2\Big].
\end{aligned}
$$

The σ_n iteration is the SGD to minimize the mean square error

$$
\begin{aligned}
\mathcal{E}_2 := E\Big[\Big\|Q(X_n,1;\theta_n,\omega_n) - r(X_n,0) + \ f(Q(X_n,0;\theta_n,\omega_n)) \\
- \max_{v\in\{0,1\}} Q(X_{n+1}, v; \theta_n, \omega_n) - \lambda(X_n;\sigma_n,\omega_n)\Big\|^2\Big].
\end{aligned}
$$

Term with the Red overline denotes averaging over all the past transitions $(X_k, U_k, R_k, X_{k+1}),\ k \le n,$ for which $X_k = X_n,\ U_k = U_n$ i.e. with fixed state-action pair.
Term with the Blue overline denotes averaging over all the past transitions $(X_k, U_k, R_k, X_{k+1}),\ k \le n,$ for which $X_k = X_n,\ U_k = 0$ (this comes from the derivation described above).



Consider parametrized families $\lambda(k;\theta')$ and $Q(i,u;\theta)$ where $\theta' = \mu + \omega$ and $\theta = \sigma + \omega$.
Here we render explicit the implicit dependence of Q on λ and therefore ω.

Figure 1: Architecture Design

Results

Circulant Dynamics

We consider a simple RMAB problem of Circulant Dynamics from [2], where we have I projects out of which we can do only K at a particular instant on priority basis. Each project has a underlying Markov chain for both Active ($u = 1$) and Passive action ($u = 0$). These two problems are taken from [1] as they serve as a basis for most of the other types of RMAB problems.

(a) Action Action (b) Passive Action

Figure 2: Underlying Markov chains of the Circulant Dynamics Problem

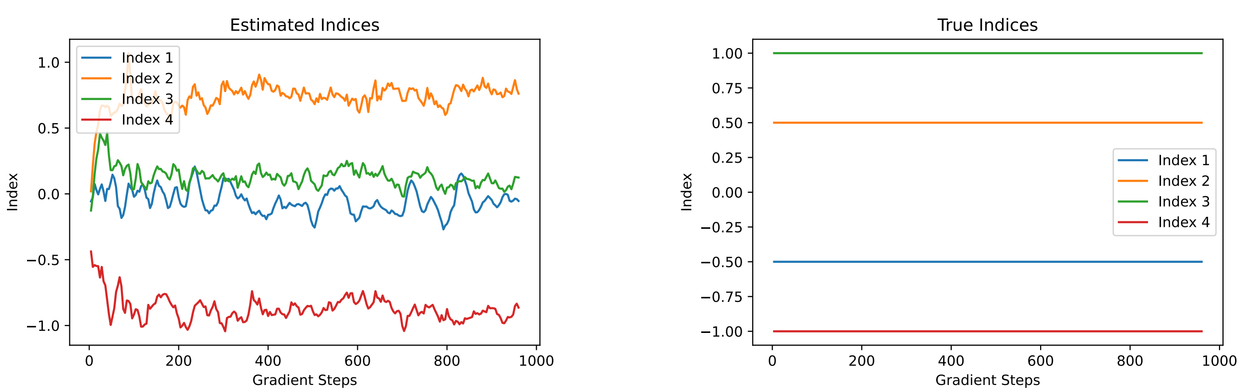
Consider the transition probability matrix $P_0 = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$, and $P_1 = P_0^T$, for passive and active action respectively.

The rewards here do not depend on action and are given by $R(1, 0) = R(1, 1) = -1$, $R(2, 0) = R(2, 1) = 0$, $R(3, 0) = R(3, 1) = 0$, and $R(4, 0) = R(4, 1) = 1$. Where $R(s, u)$ denotes the reward in state s after taking action u. Intuitively, there is a preference to activate an arm when the arm is in state 3.

For experimentation, we consider a scenario with $N = 100$ arms, out of which $M = 20$ are active at each time. The exact whittle indices for this problem as calculated in [1] are $\lambda(1) = -1/2$, $\lambda(2) = 1/2$, $\lambda(3) = 1$, and $\lambda(4) = -1$, which give priority to state 3.

For experimentation, Whittle Network is updated every 320 gradient steps of Q-Network update. Each Q-network update corresponds to updating Q-value for a particular state-action pair and each Whittle network update corresponds to updating Whittle Index for a particular state.

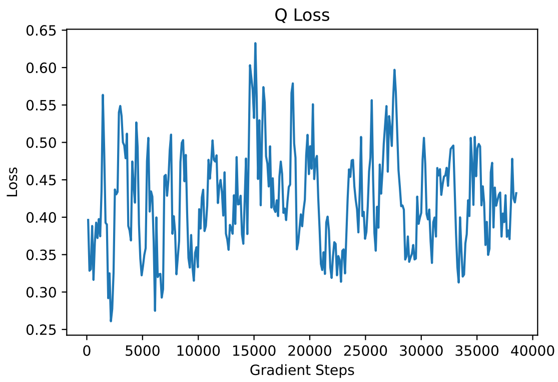
This problem is difficult due to very high stochasticity in the environment, hence as observed the Q-values are high enough throughout the training process which implies the Q-Network struggles to learn the exact Q-values. In view of this stochasticity the number 320 for gradient steps is chosen.



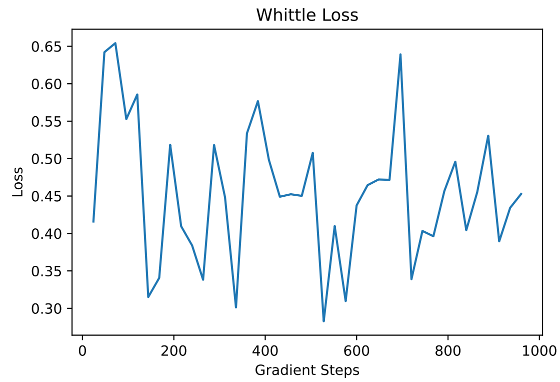
(a) Estimated Indices (b) Exact Indices

Figure 3: Whittle Indices

As seen from the figure, the estimated ordering is $\lambda(4) < \lambda(1) < \lambda(3) < \lambda(2)$ whereas the correct ordering calculated in [1] is $\lambda(4) < \lambda(1) < \lambda(2) < \lambda(3)$



(a) Q Loss



(b) Whittle Loss

Figure 4: Loss

The loss remains high, the reason I believe is mostly due to high stochasticity of the problem.

0.0.1 Circulant Dynamics with Restart

Now we consider an example where the active action forces an arm to restart from some state. We consider an example with 5 states, where in the passive mode ($u = 0$) an arm has tendency to go up the state space, i.e.,

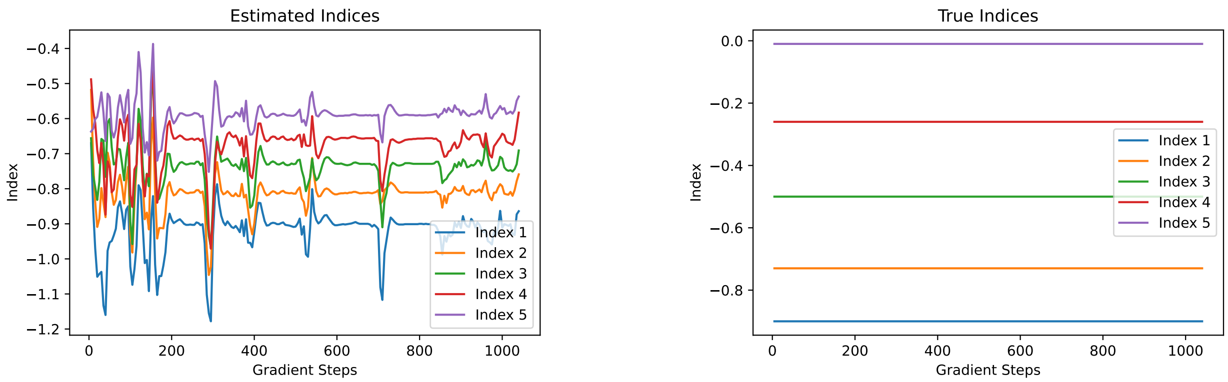
$$P_0 = \begin{bmatrix} 1/10 & 9/10 & 0 & 0 & 0 \\ 1/10 & 0 & 9/10 & 0 & 0 \\ 1/10 & 0 & 0 & 9/10 & 0 \\ 1/10 & 0 & 0 & 0 & 9/10 \\ 1/10 & 0 & 0 & 0 & 9/10 \end{bmatrix},$$

whereas in the active mode ($u = 1$) the arm restarts from state 1 with probability 1, i.e.,

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The rewards in the passive mode are given by $R(k, 0) = \alpha^k$ (α is taken to be 0.9) and the rewards in the active mode are all zero.

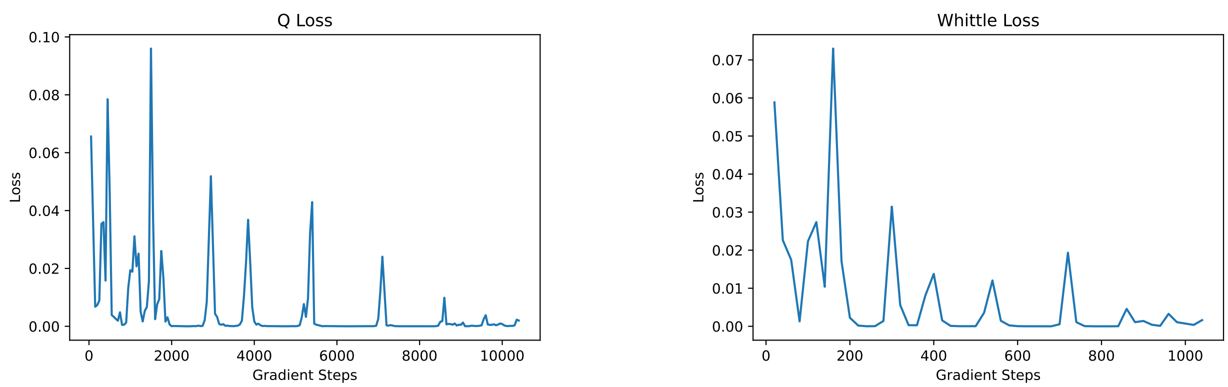
For experimentation, Whittle Network is updated every 40 gradient steps of Q-Network update.



(a) Estimated Indices (b) Exact Indices

Figure 5: Whittle Indices

Here, as seen from the figure, the ordering matches with the calculated ordering from [1] which is $\lambda(1) < \lambda(2) < \lambda(3) < \lambda(4) < \lambda(5)$. This environment was simple due to less stochastic nature.



(a) Q Loss (b) Whittle Loss

Figure 6: Loss

As seen from the above figure, Q and Whittle Loss both goes to zero as the training progresses.

References

[1] Konstantin E. Avrachenkov and Vivek S. Borkar. Whittle index based q-learning for restless bandits with average reward, 2021.

[2] 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.