Multiscale Q-learning with Function Approximation and an Application in Wireless Sensor Networks

Shalabh Bhatnagar

Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560 012, India.
shalabh@csa.iisc.ernet.in

September 08, 2014

Outline

1. An introduction to reinforcement learning
2. Markov decision processes
3. Stochastic approximation
4. Q-learning
5. Multiscale Q-learning
6. Finding optimal sleep-wake schedules in sensor networks

An Introduction to Reinforcement Learning

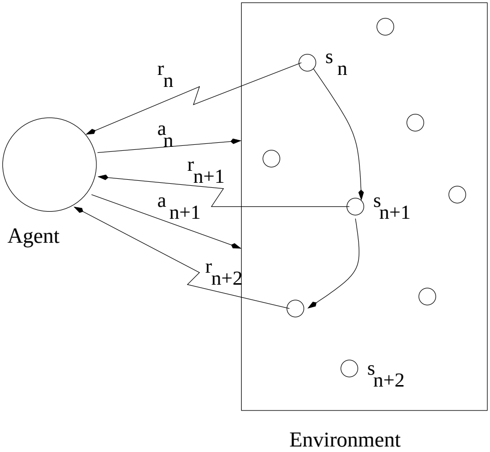


Figure: Agent-Environment Interaction

Markov Decision Processes

- A Markov Decision Process (MDP) is a controlled random process $\{s_t\}$ that depends on a control-valued sequence $\{a_t\}$ with state transitions governed according to controlled transition probabilities $P^{a_t}_{s_t,s_{t+1}}$
- Let S denote the state space and A the action space. Assume S and A are finite sets
- In general, when state is $i \in S$, feasible action space is $A(i)$. Here $A = \cup_{i \in S} A(i)$
- Let $k(s_t, a_t, s_{t+1})$ be the cost incurred when state at time t is s_t, action chosen is a_t and the next state is s_{t+1}

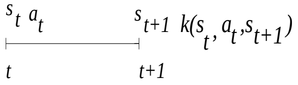


Figure: State, Action and Single-Stage Cost

The Infinite Horizon Discounted Cost Problem

- The aim is to find $\{a_t^*\}$ of actions such that for any state i,

$$V^*(i) \stackrel{\triangle}{=} V_{a_t^*}(i) = \min_{\{a_t\}} E\left[\sum_{j=0}^{\infty} \gamma^j k(s_j, a_j, s_{j+1}) \mid s_0 = i\right]$$

- It is often more convenient to work with policies rather than state-action sequences
- An admissible policy π is a sequence of functions $\pi = \{\mu_0, \mu_1, \ldots, \}$ such that each $\mu_n : S \to A$ and $\mu_n(j) \in A(j)$, $\forall j \in S$. At instant n, actions under π are selected according to μ_n
- Let Π be the set of all admissible policies

The Objective

- Objective: Find a π^* that minimizes over all $\pi \in \Pi$, the cost-to-go or the value function

$$V_\pi(i) = E\left[\sum_{j=0}^{\infty} \gamma^j k(X_j, \mu_j(X_j), X_{j+1}) \mid X_0 = i\right]$$

- Let $V^*(i) = \min_{\pi \in \Pi} V_\pi(i) = V_{\pi^*}(i)$
- A stationary deterministic policy (SDP) π is one for which $\mu_i \equiv \mu$ for all $i = 0, 1, 2, \ldots$. Many times we just call μ an SDP
- A stationary randomized policy ϕ is characterized by probability distributions $\phi(i) = (\phi(i,a), a \in A(i)), i \in S$
- It can be shown that the optimal policy (i.e., the one that attains the minimum) is an SDP and so also an SRP

The Bellman Equation

- The Bellman equation The optimal cost function V^* satisfies

$$V^*(i) = \min_{a \in A(i)} \sum_j P_{ij}^a (k(i,a,j) + \gamma V^*(j)), \quad i \in S.$$

  Further, V^* is the unique solution of this equation within the class of bounded functions

- The Bellman Equation for a Given SDP For every stationary policy μ, the associated cost function V_μ satisfies

$$V_\mu(i) = \sum_j P_{ij}^{\mu(i)} (k(i,\mu(i),j) + \gamma V_\mu(j)), \quad i \in S.$$

  Further, V_μ is the unique solution of this equation within the class of bounded functions

Limitations of Numerical Methods for Exact Schemes

- For solving Bellman optimality equations (in various cases) using numerical methods, one requires complete knowledge of transition probabilities (or *model information*) P_{ij}^{a}, $i, j \in S$, $a \in A(i)$ and the single-stage cost function
- The amount of computation required to solve Bellman equation grows exponentially in the cardinality of the state and action spaces (*the curse of dimensionality*)
- Hence, one resorts to approaches that involve a combination of "simulation" and "feature-based approximations"

- Objective: Let $F : \mathcal{R}^d \rightarrow \mathcal{R}^d$. Solve the equation $F(\theta) = 0$ when analytical form of F is not known, however, noisy measurements $F(\theta(n)) + M_{n+1}$ can be obtained, where $\theta(n), n \geq 0$ are the input parameters and $M_{n+1}, n \geq 0$ are i.i.d and zero mean

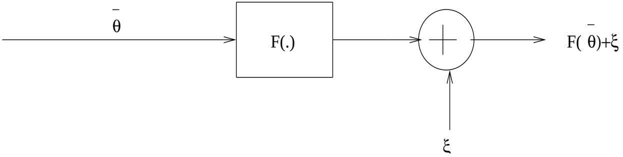


Figure: Noisy System with $E[\xi] = 0$

- $M_{n+1}, n \geq 0$ could be more general, not necessarily i.i.d.

The Stochastic Approximation Algorithm[1] [2]

- Algorithm Start with an initial $\theta(0)$ and perform the recursion

$$\theta(n+1) = \theta(n) + a(n)(F(\theta(n)) + M_{n+1}),$$

with $a(n), n \geq 0$ satisfying

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

- Let F be Lipschitz continuous
- $M_{n+1}, n \geq 0$ is a martingale difference sequence w.r.t. the filtration $\mathcal{F}_n = \sigma(\theta(m), M_m, m \leq n), n \geq 1$. Further, $E[\| \theta(n) \|^2 | \mathcal{F}_n] \leq K_1(1+ \| \theta(n) \|^2)$, for some $K_1 > 0$

[1] Originally due to Robbins and Monro [1951]

[2] The setting considered here is same as in Borkar [2008]

Analyzing the Stochastic Recursion

- In addition to foregoing, either assume or prove

$$\sup_n \parallel \theta(n) \parallel < \infty,$$

i.e., the iterates are stable[3]

- Consider the ODE

$$\dot{\theta}(t) = F(\theta(t)),$$

with A as its set of asymptotically stable equilibria

- One then shows that the algorithm's 'trajectory' asymptotically converges almost surely to A

[3] Borkar [2008], Kushner and Yin [1996]

A More General Case

- Consider the recursion

$$\theta(n+1) = \theta(n) + a(n)(F(\theta(n), Y_n) + M_{n+1}),$$

  where $Y_n, n \geq 0$ is a parameterized Markov process (with transition kernel $p_{\theta(n)}(y, dy'))$ assumed ergodic when $\theta(n) \equiv \theta$
- Let

$$G(\theta) = \int F(\theta, y)\nu_\theta(dy),$$

  where $\nu_\theta(dy)$ is the stationary distribution of $\{Y_n\}$, given θ
- Consider the ODE

$$\dot{\theta}(t) = G(\theta(t)),$$

  with B as its set of asymptotically stable equilibria
- It can be shown[4] that $\theta(n) \to B$ almost surely

[4] Borkar [2008], Benveniste, Metivier and Priouret [1991]

The Q-Bellman Equation

- Recall the Bellman equation:

$$V^*(i) = \min_{a \in A(i)} \sum_j P_{ij}^a (k(i,a,j) + \gamma V^*(j)), \quad i \in S$$

- Let

$$Q^*(i,a) = \sum_j P_{ij}^a [k(i,a,j) + \gamma V^*(j)]$$

Then, one obtains the following (Q-Bellman equation)

$$Q^*(i,a) = \sum_j P_{ij}^a [k(i,a,j) + \gamma \min_b Q^*(j,b)]$$

- Note: Q-Bellman is amenable to stochastic approximation

Regular Q-learning

- This algorithm aims to solve Q-Bellman equation using SA
- Let $\eta_n(i,a)$, $n \geq 0$ be independent random variables (simulation samples) having the common distribution $P^a_{i\cdot}$
- Let $c(n)$, $n \geq 0$ satisfy

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

- The QL-FS Algorithm: For every feasible state-action tuple (i,a), iterate

$$\begin{aligned} Q_{n+1}(i,a) = Q_n(i,a) + c(n)(k(i,a,\eta_n(i,a)) \\ +\gamma \min_v Q_n(\eta_n(i,a),v) - Q_n(i,a)) \end{aligned} \tag{1}$$

Function Approximation

- Let $Q(i,a) \approx \theta^T \phi_{i,a}$, where
  - $\phi_{i,a} = (\phi_{i,a}(1), \ldots, \phi_{i,a}(d))^T$ is a d-dimensional feature vector corresponding to (i,a), with $d << |S \times A(S)| \stackrel{\triangle}{=} M$
  - θ is a tunable d-dimensional parameter
- Let $\Phi = [[\phi_{i,a}]]$ be an $M \times d$ (feature) matrix
- Let $\Phi(k) = (\phi_{i,a}(k), (i,a) \in S \times A(S))^T$ be the kth column of Φ.

Q-learning with Function Approximation

- Q-learning with FA: Let $\{s_n\}$ denote a sample online trajectory of states of the MDP with $\{a_n\}$ as the associated action sequence. Then,

$$\theta_{n+1} = \theta_n + c(n)\phi_{s_n,a_n}(k(s_n, a_n, s_{n+1}) + \gamma \min_v \theta_n^T \phi_{s_{n+1},v} - \theta_n^T \phi_{s_n,a_n})$$

- This algorithm has been widely used in applications even though it does not empirically exhibit convergence in many cases
- There are no valid proofs of convergence available

Two-Timescale Q-learning - Key Idea [5]

- Work with parameterized SRP rather than SDP
- The exact minimization is then replaced with a gradient search in the parameterized SRP space
- The above operation is performed on a faster timescale
- Given the parameter and hence the policy update, update Q-value estimates along a slower timescale

[5] In Bhatnagar and Babu [2008], a similar idea has been used for the case of full state-action representations

Two-Timescale Q-learning

- Let $\pi_w = (\pi_w(i), i \in S)^T$ represent a class of SRP parameterized by $w \stackrel{\triangle}{=} (w_1, \ldots, w_N)^T \in C \subset \mathcal{R}^N$
- Let $\theta \in D \subset \mathcal{R}^d$ be the Q-value function parameter as before
- Assumptions
  1. The Markov process $\{X_n\}$ under any SRP π_w is aperiodic and irreducible
  2. The probabilities $\pi_w(i, a)$, $i \in S$, $a \in A(i)$ are continuously differentiable in the parameter $w \in C$. Further, $\pi_w(i, a) > 0$ $\forall (i, a) \in S \times A(S)$, $w \in C$
  3. The basis functions $\Phi(k), k = 1, \ldots, d$ are linearly independent

Fast and Slow Schedules

- Example of parameterized SRP: Boltzmann policies

$$\pi_w(i,a) = \frac{\exp(w^T \phi_{i,a})}{\sum_{b \in A(i)} \exp(w^T \phi_{i,b})}$$

- Let $\{a(n)\}$ and $\{b(n)\}$ be two step-size sequences. The following properties are satisfied:

$$\sum_n a(n) = \sum_n b(n) = \infty,$$

$$\sum_n (a(n)^2 + b(n)^2) < \infty,$$

$$\lim_{n \to \infty} \frac{b(n)}{a(n)} = 0.$$

- Note: $b(n) \to 0$ faster than $a(n)$. Thus, recursions governed by $b(n)$ are slower than those governed by $a(n)$.

The Algorithm

- For all $n \geq 0$,

$$\theta_{n+1} = \Gamma_1 \left(\theta_n + b(n)\phi_{s_n,a_n} \left(g(s_n,a_n) + \gamma\theta_n^T \phi_{s_{n+1},a_{n+1}} - \theta_n^T \phi_{s_n,a_n} \right) \right), \tag{2}$$

$$w_{n+1} = \Gamma_2 \left(w_n - a(n) \left(\frac{\theta_n^T \phi_{s_n,a_n}}{\delta} \right) (\Delta_n)^{-1} \right). \tag{3}$$

- In the above, $\Gamma_1(\cdot), \Gamma_2(\cdot)$ are suitable projection operators. Further, a_n are selected using the parameters $\Gamma_2(w_n + \delta\Delta_n)$, with Δ_n obtained using a Hadamard matrix based construction.

Hadamard Matrices

- Let H_{2^k}, $k \geq 1$ be matrices of order $2^k \times 2^k$ that are recursively obtained as:

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } H_{2^k} = \begin{pmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{pmatrix}, \; k > 1.$$

- Such matrices are called normalized Hadamard matrices[6]

[6]Bhatnagar, S., Fu, M.C., Marcus, S.I. and Wang, I.-J. [2003], Bhatnagar, S., Prasad, H.L. and Prashanth, L.A. [2013]

Hadamard Matrix Based Perturbations

- Let $P = 2^{\lceil \log_2 d \rceil}$. (Note that $P \geq d$.) Consider now the matrix H_P (with P chosen as above). Let $h(1), \ldots, h(d)$, be any d columns of H_P. In case $P = d$, then $h(1), \ldots, h(d)$, will correspond to all d columns of H_P.
- Form a matrix H'_P of order $P \times d$ that has $h(1), \ldots, h(d)$ as its columns. Let $e(p), p = 1, \ldots, P$, be the P rows of H'_P. Now set $\Delta(n)^T = e(n \text{ mod } P + 1)$, $\forall n \geq 0$. The perturbations are thus generated by cycling through the rows of H'_P with $\Delta(0)^T = e(1), \Delta(1)^T = e(2), \ldots, \Delta(P-1)^T = e(P)$, $\Delta(P)^T = e(1)$, etc.

Convergence Results for Faster Recursion

- Let

$$R(\theta, w) \triangleq \sum_{i \in S, a \in A(i)} f_w(i,a)\theta^T \phi_{i,a}$$

denote the stationary average Q-value under the parameters θ and w, respectively.

- Lemma The partial derivatives of $R(\theta, w)$ with respect to any $\theta \in D$ and $w \in C$ exist and are continuous.
- The following ODE is associated with (3):

$$\dot{w}(t) = \hat{\Gamma}_2 \left(-\nabla_w R(\theta, w(t))\right). \quad (4)$$

- Let $w(\theta)$ denote the set of asymptotically stable equilibria of (4) and $w(\theta)^\epsilon$ its ϵ-neighborhood
- Theorem Given $\epsilon > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, $w_n \to w(\theta)^\epsilon$ as $n \to \infty$ with probability one.

Convergence Results for Slower Recursion

- Proposition $w(\theta)$ is a compact subset of $\mathcal{R}^N$ for any θ.
- One may now consider the following stochastic recursive inclusion in place of (2):

$$\theta_{n+1} = \Gamma_1(\theta_n + b(n)(y_n + Y_{n+1})), \tag{5}$$

where

$$y_n = \sum_{(i,a)} f_{w_n}(i,a)\big(g(i,a) + \gamma {\theta_n}^T \sum_{(j,b)} p_{w_n}(i,a;j,b)\phi_{j,b} - {\theta_n}^T \phi_{i,a}\big)\phi_{i,a},$$

with $w_n \in w(\theta_n)^\epsilon$, $\forall n$.

- Let $h(\theta) \stackrel{\triangle}{=} \Big\{ \sum_{(i,a)} f_w(i,a)(g(i,a)$

$$+\gamma\theta^T \sum_{(j,b)} p_w(i,a;j,b)\phi_{j,b} - \theta^T \phi_{i,a})\phi_{i,a} \mid w \in w(\theta)^\epsilon \Big\}$$

- Let

$$\hat{\Gamma}_\theta(h(\theta)) \triangleq \bigcap_{\epsilon>0} \bar{co}\left(\bigcup_{\|\beta-\theta\|<\epsilon} \{\gamma_1(\beta; y+Y) \mid y \in h(\beta), Y \in A(\beta)\} \right)$$

- Proposition $h(\theta)$ satisfies the following properties:
  (i) $\hat{\Gamma}_\theta(h(\theta))$ is a convex and compact set for any $\theta \in D$.
  (ii) For all $\theta \in D$,

$$\sup_{\beta \in \hat{\Gamma}_\theta(h(\theta))} \| \beta \| < K(1 + \| \theta \|)$$

  for some $K > 0$.
  (iii) $\hat{\Gamma}_\theta(h(\theta))$ is upper-semicontinuous, i.e., if $\theta_n \to \theta$ and $\beta_n \to \beta$ with $\beta_n \in \hat{\Gamma}_{\theta_n}(h(\theta_n))\ \forall n$, then $\beta \in \hat{\Gamma}_\theta(h(\theta))$.

Convergence Results for Slower Recursion (Contd.)

- Consider now the following differential inclusion (DI):

$$\dot{\theta}(t) \in \hat{\Gamma}_{\theta}(h(\theta(t))). \tag{6}$$

- Let $\bar{\theta}(\cdot)$ be defined according to $\bar{\theta}(t(n)) = \theta_n$, $n \geq 0$, with linear interpolation on each interval $[t(n), t(n+1)]$.
- Let $G = \bigcap_{t \geq 0} \overline{\{\bar{\theta}(t+s) : s \geq 0\}}$.
- Main Theorem $\theta_n, n \geq 0$ of the QW-FA algorithm converge to G almost surely. Further, the set G is a closed connected internally chain transitive invariant set of (6).

Two-timescale Q-learning for the Average Cost Problem

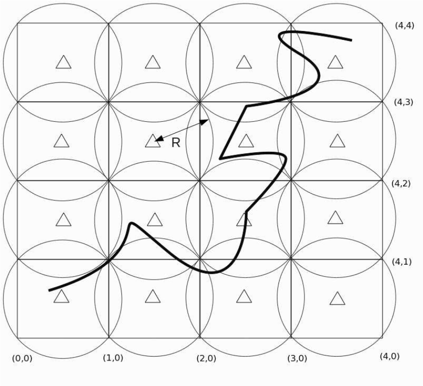
$$\theta_{n+1} = \Gamma_1\left(\theta_n + b(n)\sigma_{s_n,a_n}(g(s_n,a_n) - \hat{J}_{n+1} + \theta_n^T \sigma_{s_{n+1},a_{n+1}} - \theta_n^T \sigma_{s_n,a_n})\right),$$

$$\hat{J}_{n+1} = \hat{J}_n + c(n)\left(g(s_n,a_n) - \hat{J}_n\right),$$

$$w_{n+1} = \Gamma_2\left(w_n - a(n)\frac{\theta_n^T \sigma_{s_n,a_n}}{\delta}\Delta_n^{-1}\right)$$

- Here $a(n), b(n)$ are as before. Also, $c(n) = ka(n)$ for some $k > 0$

Application to Optimal Sleep-Wake Control in Sensors[7]



[7] Prashanth, Chatterjee and Bhatnagar [2014]

Sleep-wake Control

- In an intrusion detection application, the goal is to
  - minimize the energy consumption of the sensors, while
  - keeping tracking error to a minimum
- Setting involves partially observed Markov decision processes (POMDP) under the long-run average cost objective

The Setting

- Sensors can be either awake or sleep
- sleep time $\in \{0, \ldots, \Lambda\}$
- Object movement evolves as a Markov chain, with transition probability matrix $\mathbf{P} = [P_{ij}]_{(N+1)\times(N+1)}$
- $\mathcal{T}$: exterior of the network
- Objective:
  - Make sensors sleep to save energy
  - Keep minimum sensors awake to have good tracking accuracy
  - Find “good trade-off” between the above two conflicting objectives

Sleep–Wake Control POMDP

- State: $s_k = (l_k, r_k)$
  - l_k - intruder's location at instant k
  - $r_k(i)$ denotes the remaining sleep time of the i^{th} sensor, $i = 1, \ldots, N$ and evolves as

$$r_{k+1}(i) = (r_k(i) - 1)\mathcal{I}_{\{r_k(i)>0\}} + a_k(i)\mathcal{I}_{\{r_k(i)=0\}}$$

- Action: a_k at instant k is the vector of chosen sleep times of the sensors

Sleep–Wake Control POMDP - II

- Single-stage cost

$$g(s_k, a_k) = \mathcal{I}_{\{l_k \neq \mathcal{T}\}} \left(\sum_{\{i : r_k(i) = 0\}} c + \mathcal{I}_{\{r_k(l_k) > 0\}} \mathcal{K} \right)$$

- The states, actions and costs constitute an MDP. However, there is a problem of observability.

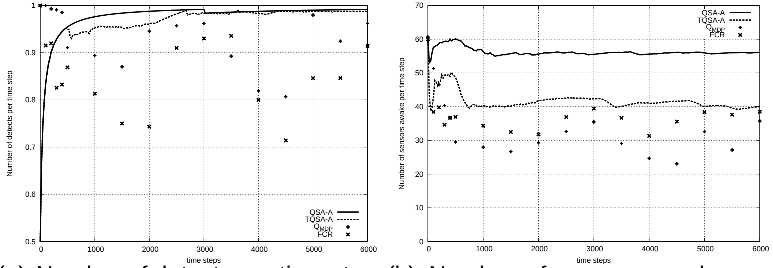
Sleep–Wake Control POMDP - III

- Note: It is not always possible to track the object (l_k)
- Hence use the sufficient statistic – $p_k = (p_k(1), ..., p_k(N), p_k(\mathcal{T}))$ - the distribution of the intruder's location - that evolves as

$$p_{k+1} = e_{l_{k+1}} \mathcal{I}_{\{r_{k+1}(l_{k+1})=0\}} + p_k P \mathcal{I}_{\{r_{k+1}(l_{k+1})>0\}}$$

- Our algorithms work with p_k and find a good enough sleeping policy

Results on a 2-d network



(a) Number of detects per time step (b) Number of sensors awake per time step

Figure: Tradeoff characteristics

- TQSA-A requires significantly less number of sensors to be awake while giving nearly the same accuracy as QSA-A
- FCR and QMDP do not show good results