Q-learning Algorithms for Optimal Stopping Based on Least Squares

H. Yu[1] D. P. Bertsekas[2]

[1] Department of Computer Science
University of Helsinki

[2] Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

European Control Conference, Kos, Greece, 2007

Outline

Basic Problem and Bellman Equation

- An irreducible Markov chain with n states and transition matrix P
  Action: stop or continue
  Cost at state i: $c(i)$ if stop; $g(i)$ if continue
  Minimize the expected discounted total cost till stop

- Bellman equations in vector notation[1]

$$J^* = \min\{c,\, g + \alpha P J^*\}, \qquad Q^* = g + \alpha P \min\{c,\, Q^*\}$$

  Optimal policy: stop as soon as the state hits the set

$$\mathcal{D} = \{i \mid c(i) \leq Q^*(i)\}$$

- Applications:
  search, sequential hypothesis testing, finance

- Focus of this paper: Q-learning with linear function approximation[2]

---

[1] α: discount factor, J^*: optimal cost, Q^*: Q-factor for the continuation action (the cost of continuing for the first stage and using an optimal stopping policy in the remaining stages)

[2] Q-learning aims to find the Q-factor for each action-state pair, i.e., the vector Q^* (the Q-factor vector for the stop action is c).

Q-Learning with Function Approximation

(Tsitsiklis and Van Roy 1999)

Subspace Approximation[3]

$$[\Phi]_{n\times s} = \begin{bmatrix} \cdots \\ \phi(i)' \\ \cdots \end{bmatrix}, \qquad Q = \Phi r \quad \text{or,} \quad Q(i,r) = \phi(i)'r$$

Weighted Euclidean Projection

$$\Pi Q = \arg\min_{r\in\Re^s} \|Q - \Phi r\|_\pi, \qquad \pi = (\pi(1),\ldots,\pi(n)) : \text{invariant distribution of } P$$

Key Fact: DP mapping F is $\|\cdot\|_\pi$-contraction and so is ΠF, where

$$FQ \stackrel{def}{=} g + \alpha P \min\{c, Q\}$$

Temporal Difference (TD) Learning solves *Projected Bellman Equation*:

$$\Phi r^* = \Pi F(\Phi r^*)$$

Suboptimal policy μ: stop as soon as the state hits the set $\{i \mid c(i) \le \phi(i)'r^*\}$[4]

$$\sum_{i=1}^{n} \pi(i)\big(J_\mu(i) - J^*(i)\big) \le \frac{2}{(1-\alpha)\sqrt{1-\alpha^2}} \|\Pi Q^* - Q^*\|_\pi$$

[3] Assume that Φ has linearly independent columns.

[4] Denote by J_μ the cost of this policy.

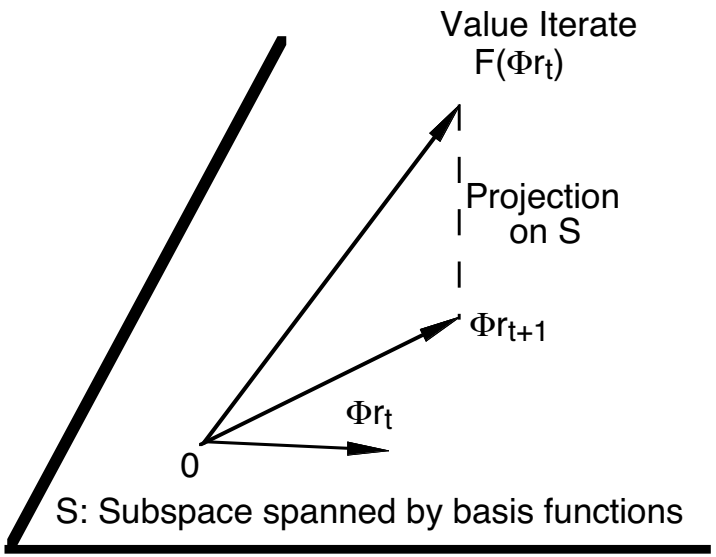
Basis of Least Squares Methods I

Projected Value Iteration

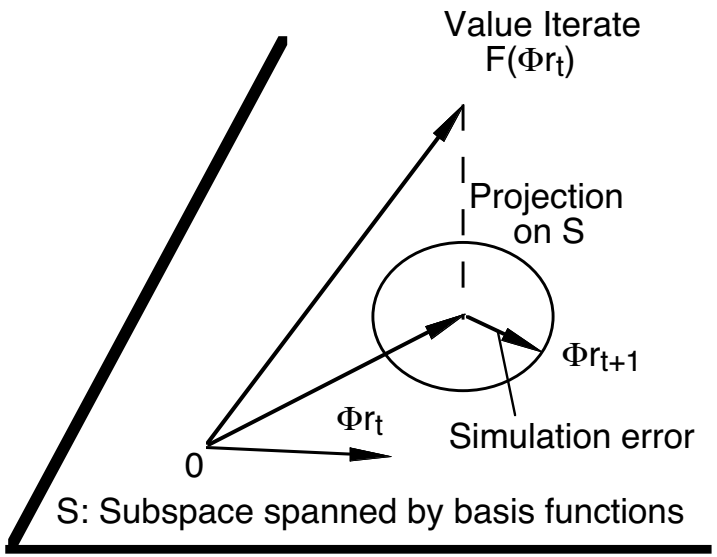
Simulation: $(x_0, x_1, \ldots)$ unstopped state process; implicitly approximate ΠF with increasing accuracy

Projected Value Iteration and LSPE (Bertsekas and Ioffe 1996):[5]

$$\Phi r_{t+1} = \Pi F(\Phi r_t), \qquad \Phi r_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi r_t) = \Pi F(\Phi r_t) + \epsilon_t$$



Projected Value Iteration



Least Squares Policy Evaluation (LSPE)

[5] Roughly speaking, $\widehat{\Pi}_t \widehat{F}_t \to \Pi F$, $\epsilon_t \to 0$ as $t \to \infty$.

Basis of Least Squares Methods II

Solving Approximate Projected Bellman Equation

LSTD (Bradtke and Barto 1996, Boyan 1999): find r_{t+1} solving an approximate projected Bellman equation

$$\Phi r_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi r_{t+1})$$

Not viable for optimal stopping because F is non-linear[6]

Comparison with Temporal Difference Learning Algorithm (Tsitsiklis and Van Roy 1999):[7]

$$r_{t+1} = r_t + \gamma_t\, \phi(x_t)\big(g(x_t, x_{t+1}) + \alpha \min\{c(x_{t+1}),\, \phi(x_{t+1})' r_t\} - \phi(x_t)' r_t\big)$$

- TD: use each sample state only once; averaging through long time interval, approximately perform the mapping ΠF
- Least squares (LS) methods: use effectively the past information; no need to store the past (in policy evaluation context)

[6] In the case of policy evaluation, this is a linear equation and can be solved efficiently.

[7] Abusing notation, we denote by $g(i, j)$ the one-stage cost of transiting from state i to j under the continuation action.

Least Squares Q-Learning

The Algorithm

$(x_0, x_1, \ldots)$ unstopped state process, $\gamma \in (0, \frac{2}{1+\alpha})$ constant stepsize

$$r_{t+1} = r_t + \gamma(\hat{r}_{t+1} - r_t) \tag{1}$$

where $\hat{r}_{t+1}$ is the LS solution:

$$\hat{r}_{t+1} = \arg\min_{r \in \Re^s} \sum_{k=0}^{t} \left(\phi(x_k)'r - g(x_k, x_{k+1}) - \alpha \min\left\{c(x_{k+1}), \phi(x_{k+1})'r_t\right\}\right)^2 \tag{2}$$

Can compute $\hat{r}_{t+1}$ almost recursively:

$$\hat{r}_{t+1} = \left(\sum_{k=0}^{t} \phi(x_k)\phi(x_k)'\right)^{-1} \sum_{k=0}^{t} \phi(x_k)\left(g(x_k, x_{k+1}) + \alpha \min\left\{c(x_{k+1}), \phi(x_{k+1})'r_t\right\}\right)$$

except the calculation of $\min\left\{c(x_{k+1}), \phi(x_{k+1})'r_t\right\}, k \leq t$ requires repartitioning past states into stopping or continuation sets (a remedy will be discussed later)

Convergence Analysis

Express LS solution in matrix notation as[8]

$$\Phi \hat{r}_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi r_t) = \widehat{\Pi}_t \left(\hat{g}_t + \alpha \tilde{P}_t \min\{c, \Phi r_t\} \right) \tag{3}$$

With probability 1 (w.p.1), for all t sufficiently large,

- $\widehat{\Pi}_t \widehat{F}_t$ is $\|\cdot\|_\pi$-contraction with modulus $\hat{\alpha} \in (\alpha, 1)$
- $(1-\gamma)I + \gamma \widehat{\Pi}_t \widehat{F}_t$ is $\|\cdot\|_\pi$-contraction for $\gamma \in (0, \frac{2}{1+\alpha})$

Proposition

$$\text{For all } \gamma \in \left(0, \frac{2}{1+\alpha}\right), \quad r_t \to r^*, \text{ as } t \to \infty, w.p.1.$$

Note: Unit stepsize is in the convergence range

[8] Here $\widehat{\Pi}_t$, $\hat{g}_t$ and $\tilde{P}_t$ are increasingly accurate simulation-based approximations of Π, g and P, respectively.

Comparison to an LSTD Analogue

$$\text{LS } Q\text{-learning:} \qquad \Phi r_{t+1} = (1-\gamma)\Phi r_t + \gamma \widehat{\Pi}_t \widehat{F}_t(\Phi r_t) \tag{4}$$

$$\text{LSTD analogue:} \qquad \Phi \tilde{r}_{t+1} = \widehat{\Pi}_t \widehat{F}_t(\Phi \tilde{r}_{t+1}) \tag{5}$$

Eq. (4) is one *single* fixed point iteration for solving Eq. (5). Yet, the LS Q-learning algorithm and the idealized LSTD algorithm have the *same* convergence rate [two-time scale argument, similar to a comparison analysis of LSPE/LSTD (Yu and Bertsekas 2006)]:[9]

Proposition

$$\textit{For all } \gamma \in \left(0\,,\, \frac{2}{1+\alpha}\right), \qquad t(\Phi r_t - \Phi \tilde{r}_t) < \infty, \quad w.p.1.$$

Implications: for all stepsize γ in the convergence range

- empirical phenomenon: r_t "tracks" $\tilde{r}_t$
- more precisely: $r_t - \tilde{r}_t \to 0$ at the rate of $O(t)$, faster than $r_t, \tilde{r}_t \to r^*$ at the rate of $O(\sqrt{t})$

[9] A coarse explanation is as follows: $\tilde{r}_{t+1}$ changes slowly at the rate of $O(t)$ and can be viewed as if "frozen" for iteration (4), which, being a contraction mapping, has geometric rate of convergence to the vicinity of the "fixed point" $\tilde{r}_{t+1}$.

Variants with Reduced Computation

Motivation

LS solution

$$\hat{r}_{t+1} = \left(\sum_{k=0}^{t} \phi(x_k)\phi(x_k)' \right)^{-1} \sum_{k=0}^{t} \phi(x_k) \left(g(x_k, x_{k+1}) + \alpha \min \left\{ c(x_{k+1}), \phi(x_{k+1})' r_t \right\} \right)$$

requires extra overhead/repartition per iteration:

$$\min \left\{ c(x_{k+1}), \phi(x_{k+1})' r_t \right\}, \quad k \leq t$$

Introduce algorithms with limited repartition at the expense of likely worse asymptotic convergence rate

First Variant: Forgo Repartition

With an Optimistic Policy Iteration Flavor

Set of past stopping decisions for state samples

$$K = \{k \mid c(x_{k+1}) \leq \phi(x_{k+1})' r_k\}$$

Replace the terms $\min\{c(x_{k+1}), \phi(x_{k+1})' r_t\}, k \leq t$ by

$$\tilde{q}(x_{k+1}, r_t) = \begin{cases} c(x_{k+1}) & \text{if } k \in K \\ \phi(x_{k+1})' r_t & \text{if } k \notin K \end{cases}$$

Algorithm

$$r_{t+1} = \left(\sum_{k=0}^{t} \phi(x_k)\phi(x_k)' \right)^{-1} \left(\sum_{k=0}^{t} \phi(x_k) g(x_k, x_{k+1}) \right.$$
$$\left. + \alpha \sum_{k \leq t,\, k \in K} \phi(x_k) c(x_{k+1}) + \alpha \sum_{k \leq t,\, k \notin K} \phi(x_k)\phi(x_{k+1})' r_t \right)$$

Can compute recursively; LSTD approach is also applicable[10]

But we have no proof of convergence at present[11]

[10] This is because the r.h.s. above is linear in r_t.

[11] Note that if the algorithm converges, it converges to the correct solution r^*.

Second Variant: Repartition within a Finite Window

Repartition at most m times per state sample, $m \geq 1$: window size
Replace the terms $\min\left\{c(x_{k+1}), \phi(x_{k+1})' r_t\right\}, k \leq t$ by

$$\min\left\{c(x_{k+1}), \phi(x_{k+1})' r_{l_{k,t}}\right\}, \qquad l_{k,t} = \min\{k+m-1, t\}$$

Algorithm

$$r_{t+1} = \arg\min_{r \in \Re^s} \sum_{k=0}^{t} \left(\phi(x_k)' r - g(x_k, x_{k+1}) - \alpha \min\left\{c(x_{k+1}), \phi(x_{k+1})' r_{l_{k,t}}\right\}\right)^2 \qquad (6)$$

Special cases

- $m \to \infty$: LS Q-learning algorithm
- $m = 1$: the fixed point Kalman filter (TD with scaling), (Choi and Van Roy 2006)

$$r_{t+1} = r_t + \frac{1}{t+1} B_t^{-1} \phi(x_t)\left(g(x_t, x_{t+1}) + \alpha \min\{c(x_{t+1}),\ \phi(x_{t+1})' r_t\} - \phi(x_t)' r_t\right)$$

Second Variant: Convergence

Proposition

*For all $m \geq 1$, r_t defined by Eq. (6) converges to r^* as $t \to \infty$, w.p.*1.

About Proof

- Two proofs are given in the extended report (Yu and Bertsekas 2006): a proof based on o.d.e. analysis (Borkar 2006, Borkar and Meyn 2001), and an alternative "direct" proof. (A weaker result w/ a boundedness assumption is mentioned in the ECC paper.)

Convergence Rate Comparison

- A simple example illustrates that

$$\text{for LS } Q\text{-learning}: \quad tE\{\|r_t - r^*\|^2\} < \infty$$

$$\text{for variant with } m \geq 1: \quad tE\{\|r_t - r^*\|^2\} = \infty$$

- Expect $m > 1$ to have practical (but not likely asymptotic) improvement of convergence speed over $m = 1$

Summary

New Q-learning Algorithm for Optimal Stopping

- Based on projected value iteration and least squares
- Convergence/convergence rate analysis
- Variants with reduced computation overhead

Future Work

- Convergence analysis of the first variant
- Empirical studies

References

For a detailed presentation and analysis see:


H. Yu and D. P. Bertsekas.
A Least Squares Q-Learning Algorithm for Optimal Stopping Problems.
LIDS report 2731, MIT, 2006; revised 2007.


H. Yu and D. P. Bertsekas.
Q-learning Algorithms for Optimal Stopping Based on Least Squares.
European Control Conference, 2007.

Available from

- Janey's web site: http://cs.helsinki.fi/u/hyu/
- Dimitri's web site: http://web.mit.edu/dimitrib/www/home.html