Experimental performance of empirical dynamic programming

Ugur Akyol and **William B. Haskell** and **Rahul Jain** and **Dileep Kalathil**

Abstract

We consider the computational efficiency of Empirical Dynamic Programming (EDP) algorithms. Empirical Value Iteration (EVI) and Empirical Policy Iteration (EPI). We contrast the performance of these algorithms with other known algorithms in the literature. Q - learning and optimistic policy iteration. We provide analysis on metrics of competing methods against EDP algorithms. We provide experiments to the nature of EDP algorithm's performance in different parameter settings. The performance of EVI and EPI is competitive compared to the major algorithms.

1 Introduction

Markov decision processes (MDPs) are the main models for sequential decision making in an uncertain environemnt, see (Puterman 1994) for a detailed review on MDPs. Value iteration and policy iteration are the classical dynamic programming algorithms for solving MDPs, but as is well known these suffer from a "curse of dimensionality". This has prompted development of a large class of approximate dynamic programming algorithms (Powell 2007). These include reinforcement learning algorithms such as Q-learning, $TD(\lambda)$, etc. However, many such algorithms are not universal, i.e., they require a certain recurrence property to converge. And often the rate of convergence can be slow for the problem dimension. In such cases, computer scientists typically resort to simulations.

In (Haskell, Jain, and Kalathil 2013), a new class of approximate dynamic algorithms for infinite horizon discounted MDPs were propsoed that rely on simulations. Their convergence was proved via stochastic dominance argument by constructing a dominating Markov chain. Non-asymptotic sample complexity bounds were also provided. Both the 'empirical value iteration' (EVI) and the 'empirical policy iteration' (EPI) are defined by replacing the expectation in the Bellman operator (and in policy evaluation) by a simulated sample-average. Thus, EVI, for example is akin to iterating the random Bellman operator. Relevant notions of probabilistic fixed points were introduced, and the algorithms were shown to converge to them.

The paper (Haskell, Jain, and Kalathil 2013) also provides some numerical evidence of convergence. In this paper, we do a more exhaustive numerical study of EVI and EPI, compare their performance to classical VI and PI, and also compare it to other approximate dynamic programming methods such as Q-learning, actor-critic algorithms and Optimistic Policy Iteration (OPI). In OPI, policy updates are carried out without waiting for policy evaluation to converge - so policy improvement is based on an incomplete evaluation (Tsitsiklis and Mahadevan 2002). Optimistic policy iteration using Monte Carlo simulation for policy evaluation, as described in (Tsitsiklis and Mahadevan 2002), offers a way to implement optimistic policy evaluation for MDPs using simulation.

As mentioned earlier, in EVI, the expectation operator of value iteration is replaced with value function's sample average approximation. In EPI, both exact policy evaluation and exact policy update are replaced by simulation. These algorithms differ from Q-Learning, optimistic policy iteration, and actor-critic methods: They are not within the family of stochastic approximation algorithms, as detailed (Borkar 1998; 2008; Borkar and Meyn 2000). An exhaustive survey of ADP methods can be found in (Powell 2007).

There are many other simulation-based approaches for MDPs, see (Chang et al. 2009)(Bhatnagar and Abdulla 2008). In (Kearns and Singh 2002), a class of reinforcement algorithms is proposed with polynomial complexity bounds in both the discounted and average cases. Probably approximately correct (PAC) learning algorithms for Markov chains and MDPs is developed in (Jain and Varaiya 2006; 2010). PAC learning performs simulation and approximation in the space of policies to tackle the unknown MDP problem. In (Jain and Varaiya 2006), uniform estimates for value function estimation are obtained. (Jain and Varaiya 2010) extends the preceding result to optimize over policies in an MDP. See (Strehl et al. 2006; Strehl, Li, and Littman 2009) for further work on PAC learning.

To summarize, our main findings are the following. EVI and EPI perform really very well with their performance comparable to exact VI and PI for even a small number of samples, say $n = 5$. Both EVI and EPI converge faster than Q-Learning and the actor-critic method. The only algorithm with comparable performance is OPI, which performs better than EVI but is outperfomed by EPI.

The paper is organized as follows. First, we recall the standard notation for Markov decision processes. We then propose our EVI and EPI algorithms after which we describe the simulation setup and measures of performance. We then empirically demonstrate the convergence EDP algorithms in comparison with value iteration and policy iteration. Next section establishes confidence intervals in convergence and CPU time of EDP versus benchmark methods. Finally, we introduce newsvendor problem setting and provide empirical estimation of performance guarantees for EVI.

2 Preliminaries

A typical representation of a discrete time MDP is the 5-tuple

$$(\mathbb{S},\, \mathbb{A},\, \{A\,(s)\,:\,s\in S\}\,,\, Q,\, c)\,.$$

The state space $\mathbb{S}$ and the action space $\mathbb{A}$ are both finite. We define $\mathcal{P}\,(\mathbb{S})$ to be the space of probability measures over $\mathbb{S}$, and we define $\mathcal{P}\,(\mathbb{A})$ similarly. For each state $s\in\mathbb{S}$, the set $A\,(s)\subset\mathbb{A}$ is the available set of feasible actions. The entire set of feasible state-action pairs is

$$\mathbb{K}\triangleq\{(s,a)\in\mathbb{S}\times\mathbb{A}: a\in A\,(s)\}\,.$$

The transition law Q governs the system evolution, $Q\,(\cdot\,|s,\,a)\in\mathcal{P}\,(\mathbb{A})$ for all $(s,a)\in\mathbb{K}$. Explicitly, $Q\,(j|s,\,a)$ for $j\in\mathbb{S}$ is the probability of next visiting the state j given the current state-action pair (s,a). Finally, $c:\mathbb{K}\to\mathbb{R}$ is a cost function that depends on state-action pairs.

Define Π to be the class of *stationary deterministic Markov policies*: mappings $\pi:\mathbb{S}\to\mathbb{A}$ which only depend on history through the current state. For a given state $s\in\mathbb{S}$, $\pi\,(s)\in A\,(s)$ is the action chosen in state s under the policy π. We explicitly assume that Π only includes feasible policies that respect the constraints $\mathbb{K}$.

The state and action at time t are denoted s_t and a_t, respectively. Any policy $\pi\in\Pi$ and initial state $s\in\mathbb{S}$ determine a probability measure P_s^π and a stochastic process $\{(s_t,a_t)\,,\,t\geq 0\}$ defined on the canonical measurable space of trajectories of state-action pairs $(Z,\,\mathcal{Z})$. The expectation operator with respect to P_s^π on $(Z,\,\mathcal{Z})$ is denoted $\mathbb{E}_s^\pi\,[\cdot]$.

We will focus on infinite horizon discounted cost MDPs with discounted factor $\alpha\in(0,1)$. For a given initial state $s\in\mathbb{S}$, the optimal cost starting from state s is

$$v^*\,(s)\triangleq\inf_{\pi\in\Pi}\mathbb{E}_s^\pi\left[\sum_{t\geq 0}\alpha^t c\,(s_t,a_t)\right], \qquad (2.1)$$

$v^*\in\mathbb{R}^{|\mathbb{S}|}$ denotes the corresponding optimal value function. The policy which minimizes the expected cost is called the optimal policy and is denoted by π^*. Given v^* one can calculate π^* and vice versa.

The standard way to find the optimal value function/optimal policy is via Dynamic Programming (DP). The two classical dynamic programming methods are, Value Iteration (VI) which computes v^* and Policy Iteration (PI) which computes π^*.

Value Iteration (VI): The Bellman operator $T:\mathbb{R}^{|\mathbb{S}|}\to\mathbb{R}^{|\mathbb{S}|}$ is defined as

$$[T\,v]\,(s)\triangleq\min_{a\in A(s)}\{c\,(s,a)+\alpha\,\mathbb{E}\,[v\,(\tilde{s})\,|s,a]\}\,,\ \forall s\in\mathbb{S}, \qquad (2.2)$$

for any $v\in\mathbb{R}^{|\mathbb{S}|}$, where $\tilde{s}$ is the random next state visited, and

$$\mathbb{E}\,[v\,(\tilde{s})\,|s,a]=\sum_{j\in\mathbb{S}}v\,(j)\,Q\,(j|s,a)$$

is the explicit computation of the expected cost-to-go conditioned on state-action pair $(s,a)\in\mathbb{K}$. VI amounts to iteration of the Bellman operator. We have a sequence $\{v^k\}_{k\geq 0}\subset\mathbb{R}^{|\mathbb{S}|}$ where $v^{k+1}=T\,v^k=T^{k+1}v^0$ for all $k\geq 0$ and an initial seed v^0. It is known that VI converges to v^* as $k\to\infty$.

Policy Iteration (PI) For a fixed policy $\pi\in\Pi$, define $T_\pi:\mathbb{R}^{|\mathbb{S}|}\to\mathbb{R}^{|\mathbb{S}|}$ as

$$[T_\pi v]\,(s)=c\,(s,\pi\,(s))+\alpha\,\mathbb{E}\,[v\,(\tilde{s})\,|s,\pi\,(s)]\,.$$

The first step of PI is the *policy evalution* step. Compute v^π by solving $T_\pi v^\pi=v^\pi$ for v^π. Let $c^\pi\in\mathbb{R}^{|\mathbb{S}|}$ be the vector of one period costs corresponding to a policy π, $c^\pi\,(s)=c\,(s,\pi\,(s))$ and Q^π, the transition kernel corresponding to the policy π. Then, writing $T_\pi v^\pi=v^\pi$ we have the linear system

$$c^\pi+Q^\pi v^\pi=v^\pi.\ \text{(Policy Evaluation)}$$

The second step is the *policy improvement* step. Given a value function $v\in\mathbb{R}^{|\mathbb{S}|}$, find an 'improved' policy $\pi\in\Pi$ with respect to v such that

$$T_\pi v=T\,v.\ \text{(Policy Update)}$$

Thus, policy iteration produces a sequence of policies $\{\pi^k\}_{k\geq 0}$ and $\{v^k\}_{k\geq 0}$ as follows. At iteration $k\geq 0$, we solve the linear system $T_{\pi^k}v^{\pi^k}=v^{\pi^k}$ for v^{π^k}, and then we choose a new policy π^k satisfying

$$T_{\pi^k}v^{\pi^k}=T\,v^{\pi^k},$$

which is greedy with respect to v^{π^k}. It is known that PI converges, i.e., $\pi^k\to\pi^*$ as $k\to\infty$.

3 The EDP algorithms

In (Haskell, Jain, and Kalathil 2013), Haskell, et al developed a theory of empirical dynamic programming (EDP) for MDPs. They proposed Empirical Value Iteration (EVI) and Empirical Policy Iteration (EPI) algorithms and showed that these converge. Each is an empirical variant of the corresponding classical algorithm. In this section, we reproduce these algorithms.

We also give a brief description about the well known Q learning (QL) algorithm, actor-critic algorithm and (modified) optimistic policy iteration algorithm for MDPs. All these three algorithms are used to 'learn' the optimal value function/policy when the underlying transition kernel Q is not known. In the following section, we compare the performance of these algorithms with the EDP algorithms in (Haskell, Jain, and Kalathil 2013).

Empirical Value Iteration (EVI)

The Bellman operator T (c.f. (2.2)) requires the exact evaluation of the expectation which requires the knowledge of the underlying transition kernel Q. To replace the exact expectation, one needs a simulation model for the transition kernel Q. Without loss of generality, one can assume that a sequence of independent uniform random variables drives this MDP. Let

$$\psi : \mathbb{S} \times \mathbb{A} \times [0,1] \to \mathbb{S}$$

be an explicit simulation model for the state evolution. With this convention, the Bellman operator can be written as

$$[T\,v]\,(s) \triangleq \min_{a \in A(s)} \left\{ c\,(s,a) + \alpha\,\mathbb{E}\left[v\left(\psi\,(s,a,\xi)\right)\right]\right\},\ \forall s \in \mathbb{S},$$

where ξ is a uniform random variable on $[0,1]$.

In EVI, the expectation $\mathbb{E}\left[v\left(\psi\,(s,a,\xi)\right)\right]$ is replaced with an empirical estimate. Given a sample of n uniform random variables, $\{\xi_i\}_{i=1}^{n}$, the empirical estimate of $\mathbb{E}\left[v\left(\psi\,(s,a,\xi)\right)\right]$ is $\frac{1}{n}\sum_{i=1}^{n} v\left(\psi\,(s,a,\xi_i)\right)$. These samples are regenerated in In each iteration. EVI algorithm is given below.

```
Algorithm 1 Empirical Value Iteration (EVI) Algorithm
Input: $v^0 \in \mathbb{R}^{|\mathbb{S}|}$, number of iterations $k_{max}$, sample size $n \geq 1$. Set counter $k = 0$.
1. Sample $n$ uniformly distributed random variables $\{\xi_i\}_{i=1}^{n}$, and $\forall s \in \mathbb{S}$, compute
       $v^{k+1}(s) = \min_{a \in A(s)} \left\{ c\,(s,a) + \frac{\alpha}{n}\sum_{i=1}^{n} v^k\left(\psi\,(s,a,\xi_i)\right)\right\}.$
2. If $k < k_{max}$, increment $k := k+1$ and return to step 1.
```

Empirical Policy Iteration (EPI)

For a fixed policy $\pi \in \Pi$, we can estimate $v^{\pi}(s)$ via simulation. Given a sequence of noise $\omega = (\xi_i)_{i \geq 0}$, we have $s_{t+1} = \psi\,(s_t, \pi\,(s_t), \xi_t)$ for all $t \geq 0$. For $\epsilon > 0$, choose a finite horizon $\mathfrak{T}$ such that

$$\max_{(s,a)\in\mathbb{K}} |c\,(s,a)| \sum_{t=\mathfrak{T}+1}^{\infty} \alpha^t < \epsilon.$$

We use the time horizon $\mathfrak{T}$ to truncate simulation, since we must stop simulation after finite time. Let

$$\left[\hat{v}^{\pi}(s)\right](\boldsymbol{\omega}) = \sum_{t=0}^{\mathfrak{T}} \alpha^t c\left(s_t(\boldsymbol{\omega}), \pi\left(s_t(\boldsymbol{\omega})\right)\right)$$

be the realization of $\sum_{t=0}^{\mathfrak{T}} \alpha^t c\,(s_t, a_t)$ on the sample path $\boldsymbol{\omega}$.

The EPI algorithm requires two input parameters, n and q, which determine sample sizes. Parameter n is the sample size for policy improvement and parameter q is the sample size for policy evaluation. In the following algorithm, the notation $s_t(\omega_i)$ is understood as the state at time t in the simulated trajectory ω_i.

EPI algorithm is given below. Step 2 replaces computation of $T_{\pi}v = T\,v$ (policy improvement). Step 3 replaces solution of the system $v = c^{\pi} + \alpha\,Q^{\pi}v$ (policy evaluation).

```
Algorithm 2 Empirical Policy Iteration (EPI) Algorithm
Input: $\pi_0 \in \Pi$, number of iterations $k_{max}$, sample size $n, q$.
Set counter $k = 0$.
1. For each $s \in \mathbb{S}$, draw $\omega_1, \ldots, \omega_q \in \Omega$ and compute
       $\hat{v}^{\pi_k}(s) = \frac{1}{q}\sum_{i=1}^{q}\sum_{t=0}^{\mathfrak{T}} \alpha^t c\left(s_t(\omega_i), \pi\left(s_t(\omega_i)\right)\right).$
2. Draw $\xi_1, \ldots, \xi_n \in [0,1]$. Choose $\pi_{k+1}$ to satisfy
       $\pi_{k+1}(s) \in \arg\min_{a \in A(s)} \left\{ c\,(s,a) + \frac{\alpha}{n}\sum_{i=1}^{n} \hat{v}^{\pi_k}\left(\psi\,(s,a,\xi_i)\right)\right\}$
   $\forall s \in \mathbb{S}$.
3. If $k < k_{max}$, increment $k := k+1$ and return to step 1.
```

Other Simulation Based Algorithms for MDPs

In this subsection we give a brief description about the other simulation based algorithms for MDPs.

Q Learning (QL): Classical (synchronous) Q Learning algorithm for discounted MDPs works as follows (see (**?**, Section 5.6)). For every state-action pair $(s,a) \in \mathbb{S} \times \mathbb{A}$, we maintain a Q function and use the update rule

$$Q^{k+1}(s,a) = Q^k(s,a) + \gamma_k\Big(c\,(s,a) + \alpha \min_{b \in \mathbb{A}} Q^k\left(\psi(s,a,\xi_k), b\right) - Q^k(s,a)\Big) \quad (3.1)$$

where ξ_k is a random noise sample on $[0,1]$ and $\{\gamma_k, k \geq 0\}$ is the standard stochastic approximation step sequence such that $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$. It can be shown that $Q^t \to Q^*$ almost surely (**?**) and $v^*(s) = \min_{a\in\mathbb{A}} Q^*(s,a)$. The rate of convergence depends on the sequence $\{\gamma_k, k \geq 0\}$ (Borkar 2008). In general, the convergence is slow.

Actor-Critic Algorithm: This is a two timescale algorithm where the value estimate as well as the policy estimate are updated in each step. The value function is updated as

$$v^{k+1}(s) = v^k(s) + \gamma_k\left(c(s,\pi^k(s)) + v^k(\psi(s,\pi^k(s),\xi)) - v^k(s)\right) \quad (3.2)$$

and the policy is updated as

$$\pi^{k+1}(s,a) = \frac{\exp(\nu_{k+1}(s,a))}{\sum_{a' \in \mathbb{A}} \exp(\nu_{k+1}(s,a'))}, \tag{3.3}$$

$$\nu_{k+1}(s,a) = \nu_k(s,a) + \beta_k \Big(v^k(s) - $$

$$c(s,a) - \alpha v^k(v^k(\psi(s,\pi^k(s),\xi'))) \Big). \tag{3.4}$$

Here $\{\gamma_k, k \geq 0\}$, $\{\beta_k, k \geq 0\}$ are the standard stochastic approximation sequence such that $\sum_k \gamma_k = \infty, \sum_k \beta_k = \infty, \sum_k(\gamma_k^2 + \beta_k^2) < \infty$ and $\frac{\beta_k}{\gamma_k} \to 0$.

Optimistic Policy Iteration (OPI) OPI was proposed by (Tsitsiklis and Mahadevan 2002). We describe a modified version of OPI where simulation method is used both in the policy evaluation and policy update steps.

Algorithm 3 Optimistic Policy Iteration (OPI) Algorithm

```
Input: $v^0 \in \mathbb{R}^{|\mathbb{S}|}$, $\pi_0 \in \Pi$, number of iterations $k_{max}$, sample size $n, q$. Set counter $k = 0$.
1. For each $s \in \mathbb{S}$, draw $\omega_1, \ldots, \omega_q \in \Omega$ and compute
       $\hat{v}^{\pi_k}(s) = \frac{1}{q} \sum_{i=1}^{q} \sum_{t=0}^{\mathfrak{T}} \alpha^t c(s_t(\omega_i), \pi(s_t(\omega_i)))$.
2. $v^k(s) = (1-\gamma_k) v^{k-1}(s) + \gamma_k \hat{v}^{\pi_k}(s), \forall s \in \mathbb{S}$.
3. Draw $\omega_1, \ldots, \omega_n \in [0,1]$. Choose $\pi_{k+1}$ to satisfy
       $\pi_{k+1}(s) \in \arg\min_{a \in A(s)} \left\{ c(s,a) + \frac{1}{n} \sum_{i=1}^{n} v^k(\psi(s,a,\omega_i)) \right\}$,
4. If $k < k_{max}$, increment $k = k+1$ and return to step 1.
```

4 Simulations

In this section we give the simulation results. All simulations are run on Intel Core i7-2630QM CPU, 2.00GHz Processor with 4.00 GB Ram. Algorithms are coded and run in MATLAB R2011a environment.

We generate a 'random' MDP, i.e., the transition matrix Q and the cost function $c(s,a)$ are generated randomly. We fix the other parameters of the MDPs, namely the number of states $|\mathbb{S}| = 100$, number of actions $|\mathbb{A}| = 10$ for each state and the discount factor $\alpha = 0.9$. Each algorithm is simulated 50 times and the plot shows a 95% confidence interval around each curve. We note that the confidence interval is so close to the mean curve and they are hardly visible. The simulation parameters n and m are as given in the EVI, EPI and OPI algorithms above (Algorithms 1, 2, 3). The stochastic approximation step size for QL, actor-critic, and modified optimistic policy iterations are assumed to be of the form $\gamma_k = 1/k^\theta$, where θ is a simulation parameter. We specify all these parameters in the plots below.

Figure 4.1 compares EVI with the benchmark, the classical VI algorithm. EVI algorithm is run for $n = 1, 5$. EVI performs remarkably well, as good as the exact VI even for a sample size $n = 1$.

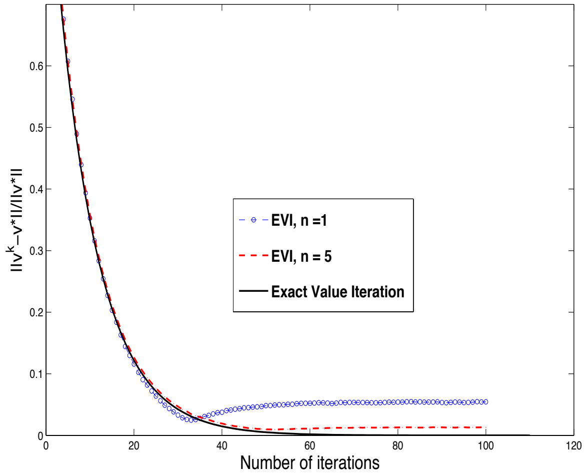

Figure 4.1: Comparison of EVI and VI

Figure 4.2 compares EVI (with $n = 5$) with QL. We also give VI as the benchmark. We set the stochastic approximation step size $\gamma_k = 1/k^\theta$, and QL is run for $\theta = 0.8, 0.6$. We see that EVI clearly outperforms QL in terms of the convergence speed. The convergence rate of QL is limited by the stepsize γ_k and the $\theta > 0.5$ to satisfy the technical conditions.

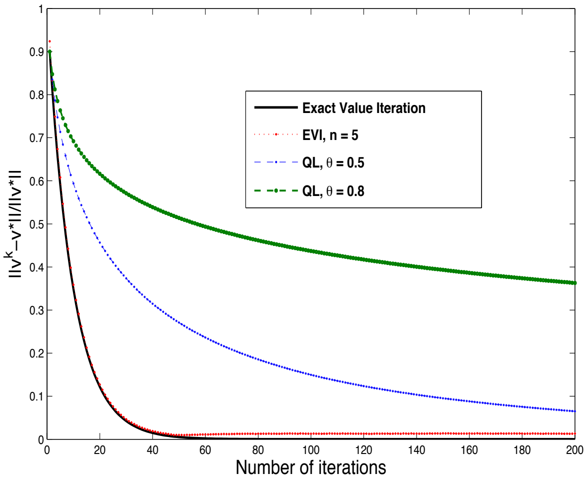

Figure 4.2: Comparison of EVI and QL

Figure 4.3 compares EPI with the benchmark, the classical PI algorithm. EPI algorithm is run for $n = 1, m = 1$ and $n = 5, m = 5$. Again, EPI performs remarkably well, as good as the exact PI even for a sample size $n = 5, m = 5$.

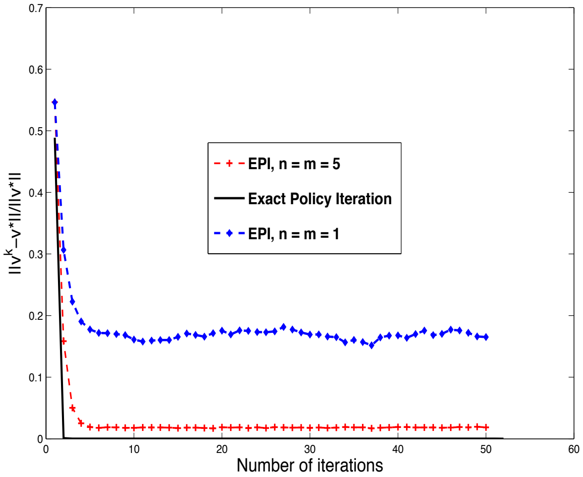

Figure 4.3: Comparison of EPI and PI

Figure 4.4 compares EPI (with $n = 5, m = 5$) with OPI and actor-critic algorithms. We also give PI as the benchmark. In OPI, we set the stochastic approximation step size $\gamma_k = 1/k^{\theta}$ with $\theta = 0.8$. We observe that, actor-critic algorithm's performance is really inferior compare to other schemes. This is expected because it known that the convergence of actor-critic is very slow because it is a two time scale algorithm. We can also observe that EPI outperforms OPI. This can be also be expected because OPI is basically a stochastic approximation scheme whose convergence is limited by the stepsize.

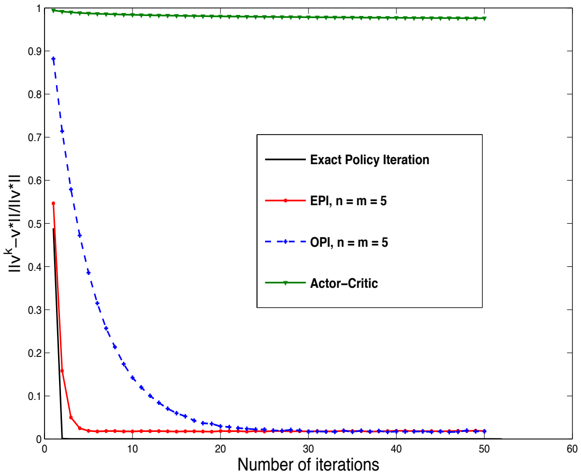

Figure 4.4: Comparison of EPI, OPI and actor-critic

Figure 4.5 gives a comparison of all the algorithms in one plot. It is known that PI iteration converges faster that VI. So, their variations also follow similar convergence behavior. So, EPI converges fastest, followed by OPI, EVI and QL.

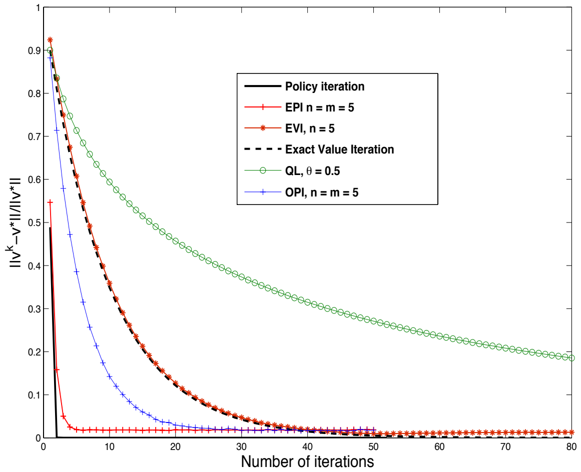

Figure 4.5: Comparison of EVI, AL, EPI, and OPI

5 The Newsvendor Problem

In (Haskell, Jain, and Kalathil 2013), it was shown that the empirical dynamic programming method can sometimes work remarkably well even for continuous states and action spaces, depending on the structure on the MDP problem. It was shown that exploiting the linear structure of the newsvendor problem, one can show the convergence of EVI. In this section we first give a brief description of the newsvendor problem. Then we give the simulation results showing that

Let D be a continuous random variable representing the stationary demand distribution. Let $\{D_k\}_{k \geq 0}$ be independent and identically distributed collection of random variables with the same distribution as D, where D_k is the demand in period k. The unit order cost is c, unit holding cost is h, and unit backorder cost is b. We let x_k be the inventory level at the beginning of period k, and we let $q_k \geq 0$ be the order quantity before demand is realized in period k.

For technical convenience, we only allow stock levels in the compact set $\mathcal{X} = [x_{\min}, x_{\max}] \subset \mathbb{R}$. This assumption is not too restrictive, since a firm would not want a large number of backorders and any real warehouse has finite capacity. Notice that since we restrict to $\mathcal{X}$, we know that no order quantity will ever exceed $q_{\max} = x_{\max} - x_{\min}$. Define the continuous function $\psi : \mathbb{R} \rightarrow \mathcal{X}$ via

$$\psi(x) = \begin{cases} x_{\max}, & \text{if } x > x_{\max}, \\ x_{\min}, & \text{if } x < x_{\min}, \\ x, & \text{otherwise}, \end{cases}$$

The function ψ accounts for the state space truncation. The system dynamic is then

$$x_{k+1} = \psi\left(x_k + q_k - D_k\right), \forall k \geq 0.$$

We want to solve

$$\inf_{\pi \in \Pi} \mathbb{E}_{\nu}^{\pi} \left[\sum_{k=0}^{\infty} \alpha^k \left(c\, q_k + \max\left\{ h\, x_k, -b\, x_k \right\} \right) \right], \qquad (5.1)$$

subject to the preceding system dynamic. We know that there is an optimal stationary policy for this problem which only depends on the current inventory level. The optimal cost-to-go function for this problem, v^*, satisfies

$$\begin{aligned} v^*(x) = \inf_{q \geq 0} \{ & c\, q + \max\left\{ h\, x, -b\, x \right\} \\ & + \mathbb{E}\left[v^*\left(\psi\left(x + q - D \right) \right) \right] \}, \qquad \forall x \in \mathbb{R}, \end{aligned}$$

where, the optimal value function $v^* : \mathbb{R} \to \mathbb{R}$. We will compute v^* by iterating an appropriate Bellman operator.

Now, the Bellman operator $T : \mathcal{C}(\mathcal{X}) \to \mathcal{C}(\mathcal{X})$ for the newsvendor problem is given by

$$\begin{aligned} [T\, v](x) = \inf_{q \geq 0} \{ & c\, q + \max\{ h\, x, -b\, x \} \\ & + \alpha\, \mathbb{E}\left[v\left(\psi\left(x + q - D \right) \right) \right] \}, \ \forall x \in \mathcal{X}. \end{aligned}$$

Value iteration for the newsvendor can then be written succinctly as $v^{k+1} = T\, v^k$ for all $k \geq 0$.

Choose the initial form for the optimal value function as

$$v^0(x) = \max\left\{ h\, x, -b\, x \right\}, \ \forall x \in \mathcal{X}.$$

It is chosen to represent the terminal cost in state x when there are no further ordering decisions. Then, value iteration yields

$$\begin{aligned} v^{k+1}(x) = \inf_{q \geq 0} \{ & c\, q + \max\left\{ h\, x, -b\, x \right\} \\ & + \alpha\, \mathbb{E}\left[v^k\left(\psi\left(x + q - D \right) \right) \right] \}, \ \forall x \in \mathcal{X}. \end{aligned}$$

Now, we do empirical value iteration with the same initial seed $\hat{v}_n^0 = v^0$. For $k \geq 0$,

$$\begin{aligned} \hat{v}_n^{k+1}(x) = \inf_{q \geq 0} \{ & c\, q + \max\left\{ h\, x, -b\, x \right\} \\ & + \frac{\alpha}{n} \sum_{i=1}^{n} \hat{v}_n^k\left(\psi\left(x + q - D_i \right) \right) \}, \ \forall x \in \mathbb{R}. \end{aligned}$$

Note that $\{D_1, \ldots, D_n\}$ is an i.i.d. sample from the demand distribution. Haskel, et al (Haskell, Jain, and Kalathil 2013) showed the convergence of this EVI algorithm for newsvendor problem.

Below we show the simulation results.

6 Conclusion

In conclusion, our experiments have verified the worth of our new EDP algorithms: EVI and EPI. These algorithms are highly competitive against current state of the art simulation-based methods for MDPs.

References

Bertsekas, D. 2011. Approximate policy iteration: a survey and some new methods. *Journal of Control Theory and Applications* 9:310–335. 10.1007/s11768-011-1005-3.

Bhatnagar, S., and Abdulla, M. S. 2008. Simulation-based optimization algorithms for finite-horizon markov decision processes. *Simulation* 84(12):577–600.

Borkar, V., and Konda, V. 1997. The actor-critic algorithm as multi-time-scale stochastic approximation. *Sadhana* 22(4):525–543.

Borkar, V. S., and Meyn, S. P. 2000. The o.d. e. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.* 38(2):447–469.

Borkar, V. S. 1998. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization* 36(3):840–851.

Borkar, V. S. 2005. An actor-critic algorithm for constrained markov decision processes. *Systems & Control Letters* 54(3):207–213.

Borkar, V. S. 2008. Stochastic approximation: a dynamical systems viewpoint.

Chang, H. S.; Fu, M. C.; Hu, J.; Steven; and Marcus, I. 2009. A survey of some simulation-based algorithms for markov decision processes.

Cooper, W., and Rangarajan, B. 2011. Performance guarantees for empirical markov decision processes with applications to multi-period inventory models. *Submitted to Operations Research.*

Haskell, W. B.; Jain, R.; and Kalathil, D. 2013. Empirical dynamic programming. *arXiv preprint arXiv:1311.5918.*

Jain, R., and Varaiya, P. P. 2006. Simulation-based uniform value function estimates of markov decision processes. *SIAM J. Control Optim.* 45(5):1633–1656.

Jain, R., and Varaiya, P. 2010. Simulation-based optimization of markov decision processes: An empirical process theory approach. *Automatica* 46(8):1297–1304.

Kearns, M., and Singh, S. 2002. Near-optimal reinforcement learning in polynomial time. *Mach. Learn.* 49(2-3):209–232.

Konda, V. R., and Borkar, V. S. 1999. Actor-critic–type learning algorithms for markov decision processes. *SIAM J. Control Optim.* 38(1):94–123.

Konda, V. R., and Tsitsiklis, J. N. 2004. Convergence rate of linear two-time-scale stochastic approximation. *Annals of Applied Probability* 796–819.

Powell, W., and Ma, J. 2011. A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications. *Journal of Control Theory and Applications* 9(3):336–352.

Powell, W. B. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics).* Wiley-Interscience.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1st edition.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, 881–888. New York, NY, USA: ACM.

Strehl, A. L.; Li, L.; and Littman, M. L. 2009. Reinforcement learning in finite mdps: Pac analysis. *J. Mach. Learn. Res.* 10:2413–2444.

Tsitsiklis, J. N., and Mahadevan, S. 2002. On the convergence of optimistic policy iteration. *Journal of Machine Learning Research* 3:2002.