Near-Optimal Control of Queueing Systems via Approximate One-Step Policy Improvement

Jefferson Huang

March 21, 2018

"Reinforcement Learning for Processing Networks" Seminar

Cornell University

Performance Evaluation and Optimization

$\exists$ various (approximate) methods for evaluating a fixed policy for an MDP.

- evaluate = compute value function
- methods include LSTD, TD(λ), ...

Policy Improvement: If v^π is the *exact* value function for the policy π, then a policy π^+ that is provably at least as good is given by:

$$\pi^+(x) \in \arg\min_{a \in A(x)} \{c(x,a) + \delta \mathbb{E}[v^\pi(X) \mid X \sim p(\cdot|x,a)]\}, \quad x \in \mathbb{X} \tag{1}$$

- Discounted Costs: $\delta \in [0,1)$, v^π gives expected discounted total cost from each state under π
- Average Costs[1]: $\delta = 1$, v^π is the "*relative value function*" under π

Policy Iteration is based on this idea.

[1] when the MDP is *unichain* (e.g., ergodic under every policy)

Approximate Policy Improvement

If v^π is replaced with an approximation $\hat{v}^\pi$, then the “improved policy” π^+ where

$$\pi^+(x) \in \arg\min_{a \in A(x)} \{c(x,a) + \delta \mathbb{E}[\hat{v}^\pi(X) \mid X \sim p(\cdot|x,a)]\}, \quad x \in \mathbb{X} \tag{2}$$

is not necessarily better than π.

Questions:

1. When is (2) computationally tractable?
2. When is π^+ close to being optimal?

Our focus is on MDPs modeling queueing systems.

Outline

Part 1: Using Analytically Tractable Policies[2]

- Average Costs

Part 2: Using Simulation and Interpolation[3]

- Average Costs

Part 3: Using Lagrangian Relaxations[4]

- Discounted Costs

[2] Bhulai, S. (2017). Value Function Approximation in Complex Queueing Systems. In Markov Decision Processes in Practice (pp. 33-62). Springer.

[3] James, T., Glazebrook, K., & Lin, K. (2016). Developing effective service policies for multiclass queues with abandonment: asymptotic optimality and approximate policy improvement. INFORMS Journal on Computing, 28(2), 251-264.

[4] Brown, D. B., & Haugh, M. B. (2017). Information relaxation bounds for infinite horizon Markov decision processes. Operations Research, 65(5), 1355-1379.

Part 1

Using Analytically Tractable Policies

Analytically Tractable Queueing Systems

Idea:

1. Start with systems whose Poisson equation is analytically solvable.
2. Use them to suggest analytically tractable policies for more complex systems.

Examples: (Bhulai 2017)

- M/Cox(r)/1 queue
- M/M/s queue
- M/M/s/s blocking system
- priority queue

The Poisson Equation

Let π be a policy (e.g., a fixed admission rule, a fixed priority rule).

In general, the Poisson equation looks like this:

$$g + h(x) = c(x, \pi(x)) + \sum_{y \in \mathbb{X}} p(y|x, \pi(x))h(y), \quad x \in \mathbb{X}.$$

We want to solve for the average cost g and the relative value function $h(x)$ of π.

The Poisson equation is also called the "evaluation equation".

- e.g., Puterman, M. L. (2005). Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.

The Poisson Equation for Queueing Systems

For queueing systems, the Poisson equation is often a linear difference equation.

- See e.g., Mickens, R. (1991). Difference Equations: Theory and Applications. CRC Press.

Example: Poisson equation for a uniformized M/M/1 queue with $\lambda + \mu < 1$ and linear holding cost rate c:

$$g + h(x) = cx + \lambda h(x+1) + \mu h(x-1) + (1 - \lambda - \mu) h(x), \quad x \in \{1, 2, \ldots\},$$

$$g + h(0) = \lambda h(1) + (1 - \lambda) h(0).$$

This is a "second-order" difference equation.

Linear Difference Equations

Theorem (Bhulai 2017, Theorem 2.1)

Suppose $f : \{0, 1, \dots\} \to \mathbb{R}$ *satisfies*

$$f(x+1) + \alpha(x) f(x) + \beta(x) f(x-1) = q(x), \quad x \geqslant 1$$

where $\beta(x) \neq 0$ *for all* $x \geqslant 1$.

If $f_h : \{0, 1, \dots\} \to \mathbb{R}$ *is a "homogeneous solution", i.e.,*

$$f_h(x+1) + \alpha(x) f_h(x) + \beta(x) f_h(x-1) = 0, \quad x \geqslant 1,$$

then, letting the empty product be equal to one,

$$\frac{f(x)}{f_h(x)} = \frac{f(0)}{f_h(0)} + \left(\frac{f(1)}{f_h(1)} - \frac{f(0)}{f_h(0)}\right) \sum_{i=1}^{x} \prod_{j=1}^{i} \frac{\beta(j) f_h(j-1)}{f_h(j+1)}$$
$$+ \sum_{i=1}^{x} \prod_{j=1}^{i} \frac{\beta(j) f_h(j-1)}{f_h(j+1)} \sum_{j=1}^{i-1} \frac{q(j)}{f_h(j+1) \prod_{k=1}^{j+1} \frac{\beta(k) f_h(k-1)}{f_h(k+1)}}$$

Application: M/M/1 Queue

Rewrite the Poisson equation in the form of the Theorem:

$$h(x+1) + \underbrace{\left(-\frac{\lambda+\mu}{\lambda}\right)}_{\alpha(x)} h(x) + \underbrace{\left(\frac{\mu}{\lambda}\right)}_{\beta(x)} h(x-1) = \underbrace{\frac{g-cx}{\lambda}}_{q(x)}, \quad x \geqslant 1.$$

Note that $f_h \equiv 1$ works as the "homogeneous solution".

We also know that, for an M/M/1 queue with linear holding cost rate c,

$$g = \frac{c\lambda}{\mu-\lambda}.$$

So, according to the Theorem,

$$\boxed{h(x)} = \frac{g}{\lambda}\sum_{i=1}^{x}\left(\frac{\mu}{\lambda}\right)^{i} + \sum_{i=1}^{x}\left(\frac{\mu}{\lambda}\right)^{i}\sum_{j=1}^{i-1}\left(\frac{\lambda}{\mu}\right)^{j+1}\left(\frac{g-cj}{\lambda}\right) = \boxed{\frac{cx(x+1)}{2(\mu-\lambda)}}.$$

Other Analytically Tractable Systems

Relative value functions for the following systems are presented in (Bhulai 2017):

1. M/Cox(r)/1 Queue
   - special cases: hyperexponential, hypoexponential, Erlang, and exponential service times
2. M/M/s Queue
   - with infinite buffer, with no buffer (blocking system)
3. 2-class M/M/1 Priority Queue

Application to Analytically Intractable Systems

Idea:

1. Pick an initial policy whose relative value function can be written in terms of the relative value functions of simpler systems.
2. Do one-step policy improvement using that policy.

In (Bhulai 2017), this is applied to the following problems:

1. Routing Poisson arrivals to two different M/Cox(r)/1 queues.
   - Initial Policy: Bernoulli routing
   - Uses relative value function of M/Cox(r)/1 queue
2. Routing in a Multi-Skill Call Center
   - Initial Policy: Static randomized policy that tries to route calls to agents with the fewest skills first
   - Uses relative value function of M/M/s queue
3. Controlled Polling System with Switching Costs
   - Initial Policy: $c\mu$-rule
   - Uses relative value function of priority queue

Controlled Polling System with Switching Costs

Two queues with independent Poisson arrivals at rates λ_1, λ_2, exponential service times with rates μ_1, μ_2 and holding cost rates c_1, c_2, respectively.

If queue i is currently being served, switching to queue $j \neq i$ costs s_i, $i = 1, 2$.

Problem: Dynamically assign the server to one of the two queues, so that the average cost incurred is minimized.

Do one-step policy improvement on the $c\mu$-rule.

Results for $\lambda_1 = \lambda_2 = 1$, $\mu_1 = 6$, $\mu_2 = 3$, $c_1 = 2$, $c_2 = 1$, $s_1 = s_2 = 2$:

Policy	Average Cost
$c\mu$-Rule	3.62894
One-Step Improvement	**3.09895**
Optimal Policy	**3.09261**

Part 2

Using Simulation and Interpolation

Scheduling a Multiclass Queue with Abandonments

k queues with independent Poisson arrivals at rates $\lambda_1, \ldots, \lambda_k$, exponential service times with rates $\mu_1, \ldots, \mu_k$, and holding cost rates $c_1, \ldots, c_k$, respectively.

Each customer in queue $i = 1, \ldots, k$ remains available for service for an exponentially distributed amount of time, with rate θ_i.

Each service completion from queue $i = 1, \ldots, k$ earns R_i; each abandonment from queue i costs D_i.

Problem: Dynamically assign the server to one of the k queues, so that the average cost incurred is minimized.

Relative Value Function

Let π be a policy, and select any reference state

$$x_r \in \mathbb{X} = \{(i_1, \ldots, i_k) \in \{0, 1, \ldots\}^k\}.$$

$g^\pi =$ average cost incurred under π

$r^\pi(x) =$ expected total cost to reach x_r under π, starting from state x

$t^\pi(x) =$ expected time to reach x_r under π, starting from state x

Then the relative value function is

$$h^\pi(x) = r^\pi(x) - g^\pi t^\pi(x), \quad x \in \mathbb{X}.$$

Approximate Policy Improvement

Exact DP is infeasible for $k > 3$ classes.

(James, Glazebrook, Lin 2016) propose an approximate policy improvement algorithm.

Idea: Given a policy π, approximate its relative value function h^π as follows:

1. Simulate π to estimate its average cost g^π and the long-run frequency with which each state is visited.
2. Based on Step 1, select a set of initial states from which the relative value under π is estimated via simulation.
3. Estimate the relative value function by interpolating between the values estimated in Step 2.
4. Do policy improvement using the estimated relative value function.

Selecting States (Step 2)

S = set of initial states selected in Step 2, from which the relative value is estimated via simulation

$$S = S_{\text{anchor}} + S_{\text{support}},$$

where

1. S_{anchor} = set of most frequently visited states (based on Step 1)
2. S_{support} = set of regularly spaced states

Parameters: How many states to include in S_{anchor} and S_{support}.

Interpolation (Step 3)

Use an (augmented) radial basis function

$$h^{\pi}(x) \approx \sum_{i=1}^{n} \alpha_i \phi(\|x - x_i\|) + \sum_{j=1}^{d} \beta_j p_j(x)$$

where

- $n =$ number of selected states in Step 2
- $x_i = i^{\text{th}}$ selected state in Step 2
- $\phi(r) = r^2 \log(r)$ (thin plate spline)
- $\|\cdot\| =$ Euclidean norm
- $d = k + 1$
- $p_1(x) = 1,\ p_j(x) =$ number of customers in queue $j - 1$

Computing the Interpolation Parameters

$x_i = i^{\text{th}}$ selected state in Step 2

$f_i =$ estimated relative value starting from x_i

$A_{ij} = \phi(\|x_i - x_j\|)$ for $i, j = 1, \ldots n$

$P_{ij} = p_j(x_i)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k+1$

Solve the following linear system of equations:

$$\boxed{\begin{aligned} A\alpha + P\beta &= f \\ P^T\alpha &= 0 \end{aligned}}$$

Example: Interpolation

2 classes, initial policy is the "$R\mu\theta$-rule"

m = number of replications for each selected state

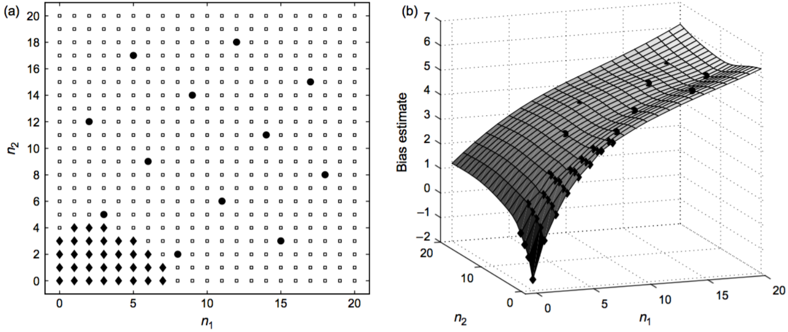

Figure 1 in (James, Glazebrook, Lin 2016)

Example: Approximate Policy Improvement

Same problem

API = policy from one-step policy improvement

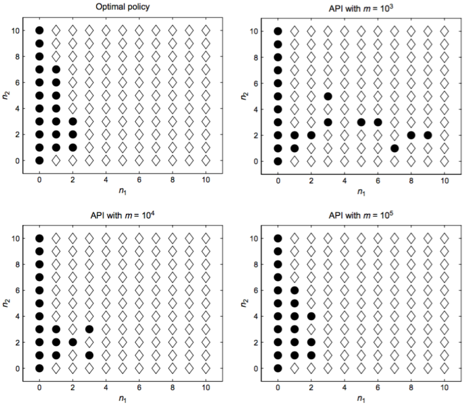

Figure 2 in (James, Glazebrook, Lin 2016)

Suboptimality of Heuristics

API(π) = one-step approximate policy improvement applied to π

$k = 3$ classes, $\rho = \sum_{i=1}^{k}(\lambda_i/\mu_i)$

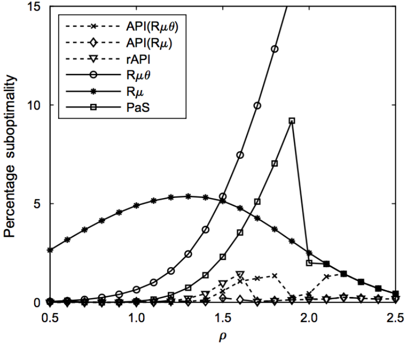

Figure 3 in (James, Glazebrook, Lin 2016); see the paper details on this and other numerical studies.

Part 3

Using Lagrangian Relaxations

Multiclass Queue with Convex Holding Costs

k queues with independent Poisson arrivals at rates $\lambda_1, \ldots, \lambda_k$ and exponential service times with rates $\mu_1, \ldots, \mu_k$, respectively.

If there are x_i customers in queue i, the holding cost rate is $c_i(x_i)$ where $c_i : \{0, 1, \ldots\} \to \mathbb{R}$ is nonnegative, increasing, and convex.

Each queue i has a buffer of size B_i

Problem: Dynamically assign the server to one of the k queues, so that the discounted cost incurred is minimized.

Relaxed Problem

The following relaxation is considered in (Brown, Haugh 2017):

- The queues are grouped into G groups.
- The server can serve at most one queue per group; can serve multiple groups simulataneously.
- Penalty ℓ for serving multiple groups simultaneously.

Under this relaxation, the value function decouples across groups ($\alpha =$ discount factor):

$$v^{\ell}(x) = \frac{(G-1)\ell}{1-\delta} + \sum_{g} v_g^{\ell}(x_g)$$

v^{ℓ} can be used in both one-step "policy improvement", and to construct a lower bound on the optimal cost via an information relaxation.

Suboptimality of Heuristics

Myopic: use one-step improvement with the value function $v^m(x) = \sum_i c_i(x_i)$

	Approximate value function (v), used in heuristic policy and in penalty															
	Myopic				LR, groups of size 1				LR, groups of size 2				LR, groups of size 4			
	Mean	SE	Gap %	Time (s)	Mean	SE	Gap %	Time (s)	Mean	SE	Gap %	Time (s)	Mean	SE	Gap %	Time (s)
$\delta = 0.9$																
Cost of heuristic policy	14.05	1.10	—	1.6	13.20	0.05	—	1.6	13.23	0.07	—	2.1	13.21	0.05	—	1.6
Gap from heuristic to v	14.05	1.10	100.0	—	1.30	0.05	9.84	1.5	1.22	0.07	9.21	0.5	1.00	0.05	7.58	18.6
Gap from heuristic to information relaxation	6.12	0.90	43.6	0.5	**0.19**	**0.04**	**1.47**	**0.5**	0.32	0.06	2.44	1.1	0.25	0.04	1.86	0.7
$\delta = 0.99$																
Cost of heuristic policy	201.73	16.5	—	18.6	204.00	0.68	—	18.3	203.66	0.44	—	29.8	203.39	0.09	—	26.6
Gap from heuristic to v	201.73	16.5	100.0	—	12.12	0.68	5.97	13.1	10.16	0.44	4.99	6.9	4.32	0.09	2.12	340.2
Gap from heuristic to information relaxation	197.98	16.5	98.1	5.2	8.12	0.67	3.98	5.1	6.44	0.43	3.16	9.9	**1.24**	**0.06**	**0.61**	**6.5**
$\delta = 0.999$																
Cost of heuristic policy	1,058.58	44.0	—	204.1	944.82	1.02	—	196.8	947.14	0.91	—	362.9	943.93	0.56	—	330.1
Gap from heuristic to v	1,058.58	44.0	100.0	—	25.98	1.02	2.75	113.8	23.47	0.91	2.48	60.1	13.36	0.56	1.42	3,665.2
Gap from heuristic to information relaxation	1,058.10	44.0	99.9	51.8	24.25	1.02	2.57	50.5	21.79	0.91	2.30	98.5	**11.98**	**0.56**	**1.27**	**63.8**

Notes. The perfect information relaxations use the uncontrolled formulation, and the heuristic policy selects actions using v as an approximate value function in (20). Bold highlights the results for the best gap for each δ. LR denotes Lagrangian relaxation.

From (Brown, Haugh 2017); see the paper for details.

Research Questions

1. Applications to other systems?
2. Performance guarantees for one-step improvement?
3. Other functions to use in one-step improvement?
4. Conditions under which one-step improvement is practical?