Sample-Efficient Reinforcement Learning With Rich Observations

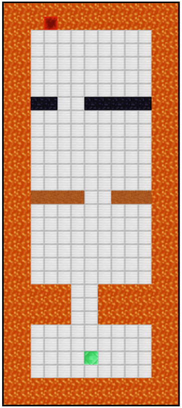
Nan Jiang
Akshay Krishnamurthy
Alekh Agarwal
John Langford
Rob Schapire

Example: Medical Treatment

- on each visit:
  - patient arrives with symptoms, test results, etc.
  - doctor decides on treatment
  - next time, patient's conditions may be different
- goal: maximize long-term favorable outcomes

Example: Robot Navigation

- at each time step, robot:
  - observes environment (say, via camera)
  - decides action to take
- goal: reach exit quickly



Reinforcement Learning

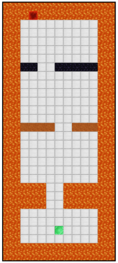
- repeat:
  - observe context
    - provides (partial) information about underlying state
  - choose action
  - get reward
  - state changes in response to selected action (and other factors)
- goal: learn to choose actions to maximize long-term reward
- realistically, context may be rich, high-dimensional, noisy, etc.
  - e.g. images, text documents, patient records, game-board positions, ...

The Challenge of Exploration

- demands experimentation and exploration — challenging!
- actions have long-term effect
- seems must learn entire behavior all at once, not bit-by-bit
  - e.g., combination lock
- learner resposible for gathering own statistics
  - not like supervised learning
    (random examples give all needed statistics)
- every episode yields information about just one trajectory
  (like huge "bandit" problem)
- seems must search entire space to be sure nothing missed
- may face very large spaces
  - can easily be too large to visit every state/context

Rich but Structured

- theory:
  - well-studied when states are visible and state space is small
  - breaks quickly in more general settings
- in practice, RL used in quite rich settings (Atari, go, etc.)
- intuitively, structure helps — e.g.:



  rich visual observations, but simple, underlying structure
- this talk: is it even information-theoretically possible to provably learn in rich but structured environments?

Main Contributions

- new algorithm for systematic exploration to learn optimal behavior
  - provably sample efficient
  - but not computationally efficient
- new measure of "structured-ness" of learning problem: "Bellman rank"
  - determines sample efficiency of algorithm
  - subsumes many previously studied settings (MDP's, POMDP's, PSR's, ...)

Formal Model

- interaction in episodes
- on each episode:
  for $h = 1, \ldots, H$, learner:
  - observes context $x_h \in \mathcal{X}$
  - chooses action $a_h \in \mathcal{A}$
  - receives reward $r_h \in \mathbb{R}$
- goal (roughly): choose actions to maximize cumulative reward

$$\sum_{h=1}^{H} r_h$$

Formal Model (cont.)

- general: x_{h+1}, r_h may depend on entire history up to when generated
- this talk: focus on simpler case:
  - assume x_{h+1}, r_h depend only on x_h, a_h
  - that is: x_h is state of (perhaps huge) MDP
- assumptions:
  - episodes are i.i.d.
  - possibly huge (or infinite) state/context space $\mathcal{X}$
  - fairly small set of possible actions $\mathcal{A}$
  - rewards bounded

Optimal Policy

- want to find good rule or policy for choosing actions based on context

$$\pi : \mathcal{X} \to \mathcal{A}$$

- measure "goodness" of π by its value:

$$\begin{aligned} V(\pi) &= \mathrm{E}\left[\sum_{h=1}^{H} r_h \;\middle|\; \pi\right] \\ &= \text{expected reward if "follow" } \pi \text{ (so } \forall h : a_h = \pi(x_h)\text{)} \end{aligned}$$

- goal: find optimal policy

$$\pi^* = \arg\max_{\pi} V(\pi)$$

Q-Learning

- standard approach: Q-learning with function approximation
- let $Q^*(x, a) =$ expected reward if:
  - start in x
  - execute a
  - then follow π^* to end of episode
- can show:

$$\pi^*(x) = \arg\max_a Q^*(x, a)$$

$\therefore$ if can learn Q^* then also have π^*

- problem: often too many states x to visit every one
  $\Rightarrow$ need to generalize across states

Function Approximation

- powerful practical approach:
  - learn approximation of Q^*
  - use function from some class to elicit generalization (e.g. neural net)
- implicit assumption: true Q^* (approximately) in class
- even with assumption,
  - no guarantee previous methods will work
  - no bound on how much data needed
  - no theory on how to explore in large spaces
- this talk: under same assumption, we give exploration algorithm that is provably correct and sample efficient

Our Setting for Function Approximation

- intuitively, assume know "form" of Q^*
- formally, assume:
  - given class $\mathcal{F}$ of "candidate" functions $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$
  - realizability: $Q^* \in \mathcal{F}$ [for now — later will relax]
  - $|\mathcal{F}|$ finite, but typically huge [can relax]
- learning problem: under these assumptions, efficiently find approximation of π^* through systematic experimentation

A First Attempt

- every $f \in \mathcal{F}$ associated with policy:

$$\pi_f(x) = \arg\max_a f(x, a)$$

- can approximate value

$$V(\pi_f) = \mathrm{E}\left[\sum r_h \mid \pi_f\right]$$

  by trying π_f on many episodes
- can do for every $f \in \mathcal{F}$ and choose best
- problem: requires $O(|\mathcal{F}|)$ episodes — huge!
- in supervised learning, usually only need $O(\ln|\mathcal{F}|)$ examples
- possible to do much better?

Bellman Equations

- to find Q^*, standard to use Bellman equations:

$$\forall x_h, a_h : \quad Q^*(x_h, a_h) = \mathrm{E}\left[r_h + Q^*(x_{h+1}, \pi^*(x_{h+1})) \;\middle|\; x_h, a_h\right]$$

- sufficient to find $f \in \mathcal{F}$ satisfying equations
- if can find then:
    - π_f optimal
    - can show:

$$V(\pi_f) = \underbrace{\mathrm{E}\left[f(x_1, \pi_f(x_1))\right]}_{\tilde{V}(f)}$$

$\Rightarrow$ can estimate π_f's value from f and samples of x_1

- problem: seem to need to visit every state to solve Bellman equations
    - how to do when state space is huge?

Eliminating Candidates

- if $f = Q^*$ then Bellman can be written:

$$\forall x_h, a_h : \; f(x_h, a_h) - \mathrm{E}\left[r_h + f(x_{h+1}, \pi_f(x_{h+1})) \;\middle|\; x_h, a_h\right] = 0$$

- since holds for all x_h, a_h, also holds (in expectation) if:
  - run another policy π for $h-1$ steps
  - arrive at (random) x_h
  - let $a_h = \pi_f(x_h)$
- yields:

$$\underbrace{\mathrm{E}\left[f(x_h, \pi_f(x_h)) - r_h - f(x_{h+1}, \pi_f(x_{h+1})) \;\middle|\; a_{1:h-1} \sim \pi\right]}_{\mathcal{E}^h(f,\pi)} = 0$$

Eliminating Candidates (cont.)

- so: if $f = Q^*$ then $\mathcal{E}^h(f, \pi) = 0$ for all π, h
- contrapositive:
  if find any π, h for which $\mathcal{E}^h(f, \pi) \neq 0$ then $f \neq Q^*$
  - can eliminate f as candidate
- can statistically estimate $\mathcal{E}^h(f, \pi)$ from random trajectories

Algorithm: "Olive" (Optimism Led Iterative Value-function Elimination)

- $\mathcal{F}_0$ = uneliminated candidates (initially $\mathcal{F}_0 = \mathcal{F}$)
- repeat
  - pick $\hat{f} \in \mathcal{F}_0$ which purports to give best policy $\pi_{\hat{f}}$
    - $\hat{f} = \arg\max_{f \in \mathcal{F}_0} \tilde{V}(f)$
      where $\tilde{V}(f) = \mathrm{E}\left[f(x_1, \pi_f(x_1))\right]$
  - test if as good as promised
    - estimate $V(\pi_{\hat{f}}) = \mathrm{E}\left[\sum_h r_h \;\middle|\; \pi_{\hat{f}}\right]$
    - check if $V(\pi_{\hat{f}}) \gtrsim \tilde{V}(\hat{f})$
  - if it is
    - output $\pi_{\hat{f}}$ and exit
  - else
    - eliminate all $f \in \mathcal{F}_0$ for which $\mathcal{E}^h(f, \pi_{\hat{f}}) \not\approx 0$
      (for any h)

Correctness

- Claim: if Olive halts, then $\pi_{\hat{f}}$ is (almost) optimal
- proof:

$$\begin{array}{rcll} V(\pi_{\hat{f}}) & \gtrsim & \tilde{V}(\hat{f}) & \text{[halting condition]} \\ & \geq & \tilde{V}(Q^*) & \text{[choice of } \hat{f}\text{; } Q^* \in \mathcal{F}_0\text{]} \\ & = & V(\pi^*) & [Q^* \text{ satisfies Bellman]} \end{array}$$

Sample Efficiency (per iteration)

- to estimate $\tilde{V}(f) = \mathrm{E}\left[f(x_1, \pi_f(x_1))\right]$ for all $f \in \mathcal{F}$:
  - make $O(\ln|\mathcal{F}|)$ repeated draws of x_1
- to estimate $\mathcal{E}^h(f, \pi_{\hat{f}})$ for all $f \in \mathcal{F}$:
  - repeat $O(|\mathcal{A}| \ln|\mathcal{F}|)$ times:
    - run $\pi_{\hat{f}}$ for $h-1$ steps
    - pick a_h uniformly at random from $\mathcal{A}$
    - observe x_h, a_h, r_h, x_{h+1}
  - to estimate $\mathcal{E}^h(f, \pi_{\hat{f}})$, include only cases for which $a_h = \pi_f(x_h)$
- need only one sample to get accurate estimates simultaneously for all $f \in \mathcal{F}$
- main remaining question: how many iterations?

Bellman Matrix and Its Rank

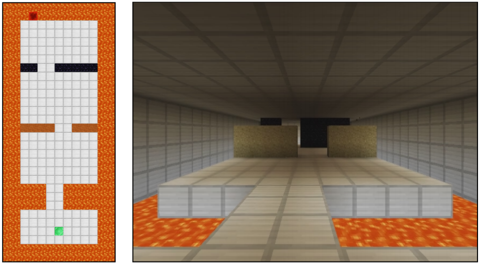
- consider full matrix of Bellman errors (for fixed h)
- rows, columns indexed by $f, f' \in \mathcal{F}$ (so $|\mathcal{F}| \times |\mathcal{F}|$)
- entry (f, f') is: $\mathcal{E}^h(f, \pi_{f'})$

$$\begin{array}{cc} & f' \\ f & \boxed{\begin{matrix} & \vdots & \\ \cdots & \mathcal{E}^h(f, \pi_{f'}) & \cdots \\ & \vdots & \end{matrix}} \end{array}$$

- rows $\leftrightarrow$ "candidates"
- columns $\leftrightarrow$ "witnesses"
- if find column f' with $\mathcal{E}^h(f, \pi_{f'}) \neq 0$, can eliminate row f
- Bellman rank $=$ rank of this matrix

Bellman Rank

- new measure of learning complexity
- claim: number of iterations of Olive is polynomial in Bellman rank
- can be bounded by (or in terms of):
  - number of states of MDP
  - number of “hidden” states, e.g.:



size of grid, not size of observation space

  - rank of PSR
  - dimension of LQR state space

Bounding Olive Iterations by Bellman Rank

- say can estimate all expectations exactly
- on earlier iterations, found $\hat{f}_1, \hat{f}_2, \ldots$
  - correspond to columns of Bellman matrix
- $\hat{f} \in \mathcal{F}_0 = \{$rows f with all 0's in selected columns$\}$

$$\begin{array}{cc} & \begin{array}{cccccccc} & \hat{f}_3 & \hat{f}_1 & \hat{f} & & \hat{f}_2 & & \hat{f}_4 \end{array} \\ \hat{f} & \boxed{\begin{array}{cccccccc} & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ \cdots & 0 & 0 & 2.1 & \cdots & 0 & \cdots & 0 \\ & \vdots & \vdots & \vdots & & \vdots & & \vdots \end{array}} \end{array}$$

- can show: $\tilde{V}(\hat{f}) \neq V(\pi_{\hat{f}}) \Rightarrow \exists h : \mathcal{E}^h(\hat{f}, \pi_{\hat{f}}) \neq 0$
- new column linearly independent of columns already found

$\therefore$ (# iterations) $=$ (# columns) $\leq$ Bellman rank

- for approximate estimates of expectations, use geometric argument based on ellipsoid volumes

Main Theorem

- Theorem: Let M be Bellman rank. With high probability, Olive returns policy $\hat{\pi}$ with $V(\hat{\pi}) \geq V(\pi^*) - \epsilon$, and the number of episodes executed is at most

$$\tilde{O}\left(\frac{M^2 H^3 |\mathcal{A}| \ln |\mathcal{F}|}{\epsilon^2}\right).$$

More General Formulation

- can generalize framework to remove realizability assumption
- given:
  - space Π of policies $\pi : \mathcal{X} \to \mathcal{A}$
  - set $\mathcal{G}$ of candidate "value functions" $g : \mathcal{X} \to \mathbb{R}$
- can show: if there is a "good" policy $\pi \in \Pi$ whose value function is in $\mathcal{G}$ then Olive will learn to do as well as (best such) π
- earlier formulation is special case
- agnostic — don't need $Q^* \in \mathcal{F}$, $\pi^* \in \Pi$, etc.

Generalizations and Extensions

- so far, considered large-state, visible MDP's
- actually holds for much more general processes where
  - context x is any observable information
  - policies π are reactive
  - e.g. POMDP's with rich observations x
- can allow Π, $\mathcal{G}$ (or $\mathcal{F}$) to be infinite
  - get bounds in terms of VC-like measures
- robustness
  - okay if only approximation of value function is in $\mathcal{G}$
  - okay if Bellman error matrix is only approximated by low-rank matrix

Summary

- Bellman rank:
  - new measure of structural complexity
  - captures many other settings
- Olive:
  - first provably sample-efficient exploration algorithm for general contextual decision processes
  - allows rich observations
  - polynomial in Bellman rank
- main open problem: find algorithm with similar properties that is also computationally efficient