Computational Limits & The LP formulation

Announcements

- HW0: due this Thursday 11:59pm
- Gradescope (please self-enroll)

- Norms for class?
  - Video:
  - Questions:

Today:

- Recap:
  - value/policy iteration + contraction

- Today: computational complexity & the linear programming approach

Question: Given an MDP $\mathcal{M} = (S, A, P, r, \gamma)$ can we exactly compute $Q^\star$ (or find $\pi^\star$) in polynomial time?

Recap

Define Bellman Operator $\mathscr{T}$:

Given a function $f : S \times A \mapsto \mathbb{R}$,

$$\mathscr{T}f : S \times A \mapsto \mathbb{R},$$

$$\left(\mathscr{T}f\right)(s,a) := r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \max_{a' \in A} f(s',a'), \forall s,a \in S \times A$$

$\mathcal{T}Q = Q \Leftarrow$ Bellman eq.

Value Iteration Algorithm:

1. Initialization: $Q^0 : \|Q^0\|_\infty \in (0, \frac{1}{1-\gamma})$

2. Iterate until convergence: $Q^{t+1} = \mathcal{T} Q^t$

Policy Iteration Algorithm:

Closed-form for PE
(see 1.1.3 in Monograph)

1. Initialization: $\pi^0 : S \mapsto \Delta(A)$

2. Policy Evaluation: $Q^{\pi^t}(s, a), \forall s, a$

3. Policy Improvement $\pi^{t+1}(s) = \arg\max_a Q^{\pi^t}(s, a), \forall s$

Final Quality of the Policy (for VI):

$$\pi^t : \pi^t(s) = \arg\max_a Q^t(s,a)$$

Theorem: $V^{\pi^t}(s) \geq V^\star(s) - \frac{2\gamma^t}{1-\gamma} \underbrace{\|Q^0 - Q^\star\|_\infty}_{\leq \frac{1}{1-\gamma}} \forall s \in S$

$\Longrightarrow$

$$\frac{2\gamma^t}{(1-\gamma)^2} \leq \epsilon$$

Set $Q^0 = 0$. After $t \geq \dfrac{\log \frac{2}{\epsilon(1-\gamma)^2}}{1-\gamma}$ iterations, we have:

$V^{\pi^t}(s) \geq V^\star(s) - \epsilon \quad \forall s \in S$

$$\gamma^t = (1-(1-\gamma))^t \leq e^{-(1-\gamma)\cdot t}$$

Same rate for PI.

Today

Polynomial Time & Strongly Polynomial Complexity

- Complexity to compute an exact solution given $\mathscr{M}$. (Aside: Why?)
  - Assume that basic arithmetic operations (+,-,x,÷) take unit time.

Polynomial Time & Strongly Polynomial Complexity

- Complexity to compute an exact solution given $\mathscr{M}$. (Aside: Why?
  - Assume that basic arithmetic operations (+,-,x,÷) take unit time.
- Polytime computation: Suppose that (P, r, γ) in our MDP $\mathscr{M}$ is specified with rational entries, where $L(P, r, \gamma)$ is total bit-size required to specify (P, r, γ).

  $P(s=4 \mid s=2, a=8) = 10/11$ $\quad 1-\gamma = 10^{-4}$

  Can we (exactly) compute $Q^\star$ in time that is polynomial in $L(P, r, \gamma)$, # states S, and #actions A.

Polynomial Time & Strongly Polynomial Complexity

- Complexity to compute an exact solution given $\mathscr{M}$. (Aside: Why?
    - Assume that basic arithmetic operations (+,-,x,÷) take unit time.
- Polytime computation: Suppose that (P, r, γ) in our MDP $\mathscr{M}$ is specified with rational entries, where $L(P, r, \gamma)$ is total bit-size required to specify (P, r, γ).
  Can we (exactly) compute $Q^\star$ in time that is polynomial in $L(P, r, \gamma)$, # states S, and #actions A.
- Strongly polynomial time: Suppose (P, r, γ) is specified with real numbers. Can we compute $Q^\star$ in poly($S, A, \log(1/(1-\gamma))$), with no dependence on $L(P, r, \gamma)$?

Computational Complexities of our Iterative Algorithms

Value Iteration

- When the gap in the current objective value and the optimal objective value is smaller than $2^{-L(P,r,\gamma)}$, then the greedy policy will be optimal.
  (this is a standard argument in optimization)

Value Iteration

- When the gap in the current objective value and the optimal objective value is smaller than $2^{-L(P,r,\gamma)}$, then the greedy policy will be optimal.
(this is a standard argument in optimization)

$\epsilon \simeq O(2^{-L})$

- VI:
  - needs $\log\Big(1/(\epsilon(1-\gamma)\Big)/(1-\gamma)$ iterations to obtain an ϵ accurate solution.
  - Per iteration complexity: S^2A

Value Iteration

- When the gap in the current objective value and the optimal objective value is smaller than $2^{-L(P,r,\gamma)}$, then the greedy policy will be optimal. (this is a standard argument in optimization)
- VI:
  - needs $\log\Big(1/(\epsilon(1-\gamma)\Big)/(1-\gamma)$ iterations to obtain an ϵ accurate solution.
  - Per iteration complexity: S^2A
- Poly runtime? For fixed γ, VI is poly:

$$O\left(S^2A\frac{L(P,r,\gamma)+\log(1/(1-\gamma))}{1-\gamma}\right)$$

Technically, not even poly.
(For fixed γ it is!)

Value Iteration

- When the gap in the current objective value and the optimal objective value is smaller than $2^{-L(P,r,\gamma)}$, then the greedy policy will be optimal.
(this is a standard argument in optimization)
- VI:
  - needs $\log\Big(1/(\epsilon(1-\gamma)\Big)/(1-\gamma)$ iterations to obtain an ϵ accurate solution.
  - Per iteration complexity: S^2A
- Poly runtime? For fixed γ, VI is poly:

$$S^2A\frac{L(P,r,\gamma)\log(1/(1-\gamma))}{1-\gamma}$$

- Strongly poly? No

must have dep. on $L(P,r,\gamma)$

compute $V^{\pi} \to Q^{\pi}$

Policy Iteration

to compute V^{π} for given π, involves solving a linear system.

- PI Per iteration complexity: $S^3 + S^2 A$
  - PI is more costly than VI per iteration.
  - PI is observed to be much faster than VI to obtain an exact opt policy.

Policy Iteration

- PI Per iteration complexity: $S^3 + S^2 A$
    - PI is more costly than VI per iteration.
    - PI is observed to be much faster than VI to obtain an exact opt policy.
- Poly runtime? For fixed γ, ~~VI~~ PI is poly: (upper band)

$$\underbrace{(S^3 + S^2 A)}_{\text{difference}} \frac{L(P, r, \gamma)\ \log(1/(1-\gamma))}{1-\gamma}$$

Policy Iteration

- PI Per iteration complexity: $S^3 + S^2 A$
  - PI is more costly than VI per iteration.
  - PI is observed to be much faster than VI to obtain an exact opt policy.
- Poly runtime? For fixed γ, VI is poly:

$$(S^3 + S^2 A)\frac{L(P, r, \gamma)\,\log(1/(1-\gamma))}{1-\gamma}$$

- Does PI compute an optimal policy in time independent of $L(P, r, \gamma)$?

Yes! finite # of policies

A^S

Is PI a strongly poly algo?

- Does PI compute an optimal policy in time independent of $L(P, r, \gamma)$?
  - Yes: after A^S iterations (A^S is the number of policies)
    Refinement: [Mansour & Singh '99] PI halts after A^S/S iterations.

Is PI a strongly poly algo?

- Does PI compute an optimal policy in time independent of $L(P, r, \gamma)$?
  - Yes: after A^S iterations (A^S is the number of policies)
    Refinement: [Mansour & Singh '99] PI halts after A^S/S iterations.

- Is PI strongly polynomial?
  For $\boxed{\text{fixed } \gamma,}$ yes:
  [Ye '12] PI halts after $\dfrac{S^2 A \log(S^2/(1-\gamma))}{1-\gamma}$ iterations.

no dependence on $L(P, r, \gamma)$

Summary Table

	Value Iteration	Policy Iteration	LP-based Algorithms
Poly.?	$S^2A\frac{L(P,r,\gamma)\,\log\frac{1}{1-\gamma}}{1-\gamma}$	$(S^3+S^2A)\frac{L(P,r,\gamma)\,\log\frac{1}{1-\gamma}}{1-\gamma}$	?
Strongly Poly.?	X	$(S^3+S^2A)\cdot\min\left\{\frac{A^S}{S},\frac{S^2A\log\frac{S^2}{1-\gamma}}{1-\gamma}\right\}$	?

for VI & PI these are for fixed γ.

- VI Per iteration complexity: S^2A
- PI Per iteration complexity: S^3+S^2A

Are VI and PI Polynomial Time algorithms?
(technically, no)

yes, for fixed γ.

Is there a polytime (and strongly polytime) algo
for an MDP??
YES! Linear Programming

The Primal Linear Program

- We can write the Bellman equations with values rather than Q-values:

V

$$V(s) = \max_a \left\{ r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V(s) \right] \right\}$$

- An equivalent way to write the Bellman equations is as a linear program. With variables $V \in \mathbb{R}^S$, the LP is:

$$\min V(s_0)$$

$$\text{s.t.} \quad V(s) \geq r(s,a) + \mathbb{E}_{s' \sim P(\cdot|s,a)} V(s') \quad \forall s, a \in S \times A$$

if V is feasible

$\Rightarrow V \geq V^*$

$\sum_{s'} P(s'|s,a)\, V(s')$

LP Runtimes and Comments

$n + rd\,\gamma^{cl}$ $\qquad$ $\lg\left(\frac{1}{1-\gamma}\right)$

- Using a polytime LP solver, gives us a poly time algorithm.

LP Runtimes and Comments

- Using a polytime LP solver, gives us a poly time algorithm.
- [Ye, '05]: there is an interior point algorithm (CIPA) which is also strongly polynomial. $\text{By } (1/1-\gamma)$ and no dep. on $L(P, r, \gamma)$

LP Runtimes and Comments

- Using a polytime LP solver, gives us a poly time algorithm.
- [Ye, ’05]: there is an interior point algorithm (CIPA) which is also strongly polynomial.

- Relations:
  - VI is best thought of as a fixed point algorithm
  - PI is equivalent to a (block) simplex algorithm
    (Recall the simplex algo, in general, could be exp time.
    But not for MDPS, at least for fixed γ.)

Summary Table

	Value Iteration	Policy Iteration	LP-based Algorithms
Poly.	$S^2 A \frac{L(P,r,\gamma)\,\log\frac{1}{1-\gamma}}{1-\gamma}$	$(S^3 + S^2 A)\frac{L(P,r,\gamma)\,\log\frac{1}{1-\gamma}}{1-\gamma}$	$S^3 A\, L(P, r, \gamma)$
Strongly Poly.	X	$(S^3 + S^2 A) \cdot \min\left\{ \frac{A^S}{S}, \frac{S^2 A \log\frac{S^2}{1-\gamma}}{1-\gamma} \right\}$	$S^4 A^4 \log\frac{S}{1-\gamma}$

fixed γ

- VI Per iteration complexity: $S^2 A$
- PI Per iteration complexity: $S^3 + S^2 A$
- The LP approach is only logarithmic in $1 - \gamma$

What about the Dual LP?

- The linear programming is helpful in understanding the problem. (even though it is not used often)
- Let us now consider the dual LP.
  - It is also very helpful conceptually.
  - In some cases, it also provides a reasonable algorithmic approach

- Let us start by understanding the dual variables and the "state-action polytope"

State-Action Visitation Measures

- For a fixed (possibly stochastic) policy π, define the state-action visitation distribution ν^π as:

$$\nu^\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr^\pi(s_t = s, a_t = a \mid s_0)$$

$Q^\pi(s,a)$

$= \frac{1}{1-\gamma}\sum_{s,a} r(s,a)\,\nu^\pi(s,a)$

where $\Pr^\pi(s_t = s, a_t = a \mid s_0)$ is the state-action visitation probability when we execute π starting at state s_0.

State-Action Visitation Measures

- For a fixed (possibly stochastic) policy π, define the state-action visitation distribution ν^π as:
$$\nu^\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathrm{Pr}^\pi(s_t = s, a_t = a \,|\, s_0)$$
where $\mathrm{Pr}^{\pi}(s_t = s, a_t = a \,|\, s_0)$ is the state-action visitation probability when we execute π starting at state s_0.

- We can verify that have ν^π satisfies, for all states $s \in S$:
$$\sum_a \nu^\pi(s,a) = (1-\gamma)I(s = s_0) + \gamma \sum_{s',a'} P(s \,|\, s', a')\nu^\pi(s', a')$$

The "State-Action" Polytope

- Let us define the state-action polytope K as follows:

$$K := \left\{ \nu \,|\, \nu \geq 0 \text{ and } \right.$$

$$\left. \sum_a \nu(s,a) = (1-\gamma) I(s = s_0) + \gamma \sum_{s',a'} P(s \,|\, s', a') \nu(s', a') \right\}$$

The “State-Action” Polytope

- Let us define the state-action polytope K as follows:

$$K := \left\{ \nu \,|\, \nu \geq 0 \;\; \text{and} \right.$$

$$\left. \sum_a \nu(s,a) = (1-\gamma)I(s=s_0) + \gamma \sum_{s',a'} P(s\,|\,s',a')\nu(s',a') \right\}$$

The "State-Action" Polytope

- Let us define the state-action polytope K as follows:

$$K := \Big\{ \nu \,|\, \nu \geq 0 \ \text{ and }$$

$$\sum_a \nu(s,a) = (1-\gamma) I(s = s_0) + \gamma \sum_{s',a'} P(s \,|\, s', a') \nu(s', a') \Big\}$$

- This set precisely characterizes all state-action visitation distributions:

The "State-Action" Polytope

- Let us define the state-action polytope K as follows:

$$K := \left\{ \nu \,|\, \nu \geq 0 \ \text{ and } \sum_a \nu(s,a) = (1-\gamma) I(s = s_0) + \gamma \sum_{s',a'} P(s \,|\, s', a') \nu(s', a') \right\}$$

- This set precisely characterizes all state-action visitation distributions:

Lemma: K is equal to the set of all feasible state-action distributions, i.e. $\nu \in K$ if and only if there exists a (possibly randomized) policy π s.t. $\nu^\pi = \nu$

The Dual LP

$$\max \sum_{s,a} \nu(s,a) r(s,a)$$

$$\text{s.t.} \quad \nu \in K$$

- One can verify that this is the dual of the primal LP.
- Note that K is a polytope