Reinforcement Learning (Learning Algos)

Friday, March 27, 2020 3:05 PM

Previously, we covered the planning problem (aka "Dynamic Programming")
which finds a good policy π assuming the environment (reward + transition probabilities) is known.
This was "tractable" (eg, equiv. to an LP) though still have difficulties when $|S| = 10^{400}$...

Now, consider learning problems, where not only do we want π, but the environment is unknown.
Simultaneously learn π and transition probabilities. There is now a clear exploration-exploitation tradeoff.
This is fundamentally harder.

Types of learning methods:

"Evolutionary methods" vs Structured RL methods (based on estimating $V(s)$)

"Evolutionary methods"
genetic algos, simulated annealing
Don't exploit structure
Sometimes, not often, effective
(We won't cover)

Structured RL methods (based on estimating $V(s)$)
exploit structure of RL problem

"Model-based"
Phase 1: learn environment
Phase 2: use planning algo
often not as efficient
ex: TD(0)

"Model-free"
learn environment and π simultaneously. Our focus
ex: Q-learning

Background: Stochastic Approx.

Law of Large Numbers (LLN)

Let X_i be iid (or mutually independent) realizations of a r.v. X s.t. $\mathbb{E}[X] = \mu$ and $\mathbb{E}[X^2] < \infty$

Define $\hat{\mu}_m = \frac{1}{m} \sum_{i=1}^{m} X_i$ (note: $\hat{\mu}_{m+1} = \underbrace{\frac{1}{m+1}}_{\alpha} X_{m+1} + \underbrace{\frac{m}{m+1}}_{1-\alpha} \hat{\mu}_m$)

then

Weak LLN: $\hat{\mu}_m \xrightarrow{P} \mu$, i.e., $\lim_{m \to \infty} \mathbb{P}(|\hat{\mu}_m - \mu| > \varepsilon) = 0 \quad (\forall \varepsilon > 0)$

Strong LLN: $\hat{\mu}_m \xrightarrow{a.s.} \mu$, i.e., $\mathbb{P}(\lim \hat{\mu}_m = \mu) = 1$

$\alpha_m = \frac{1}{m}$ (indexing is off due to 0 vs 1 based)

Thm 17.4 (mean estimation, generalization of SLLN)

$X \in [0,1]$ a r.v., w/ iid realizations $\{X_0, \dots, X_m\}$, and define $\hat{\mu}_{m+1} = (1 - \alpha_m)\hat{\mu}_m + \alpha_m X_m$ (From now on, drop $\hat{}$ notation)

s.t. $\hat{\mu}_0 = X_0$, $\alpha_m \in [0,1]$, $\sum \alpha_m = +\infty$, $\sum \alpha_m^2 < \infty$ (ex: $\alpha_m = \frac{1}{m}$)

($\Rightarrow \mathrm{Var}(X) < \infty$)

then $\hat{\mu}_m \xrightarrow{a.s.} \mathbb{E}[X]$ ($\Rightarrow \hat{\mu}_m \xrightarrow{q.m.} \mathbb{E}[X]$ quadratic mean, aka L^2, convergence means $\lim_{m \to \infty} \mathbb{E}[(\hat{\mu}_m - \mathbb{E}[X])^2] = 0$)

Proof (of q.m., not a.s., for simplicity)

- $\mathbb{E}[\mu_0] = \mathbb{E}[X_0] = \mathbb{E}[X]$, and by linearity and induction, $\mathbb{E}[\mu_m] = \mathbb{E}[X]$ also

thus convergence in L^2 just means $\lim_m \mathrm{Var}[\mu_m] = 0$

- By independence, $\mathrm{Var}(\mu_{m+1}) = (1-\alpha_m)^2 \mathrm{Var}(\mu_m) + \alpha_m^2 \cdot \underbrace{\mathrm{Var}(X)}_{\le 1}$

$\le (1-\alpha_m)^2 \mathrm{Var}(\mu_m) + \alpha_m^2$ $\quad$ $1-\alpha_m = t$, $t \in [0,1]$ so $t^2 \le t$

$\le (1-\alpha_m)\mathrm{Var}(\mu_m) + \alpha_m^2$

we're going to start by showing $\liminf_m \mathrm{Var}(\mu_m) = 0$

... Suppose not. Since $\mathrm{Var}(\cdot) \ge 0$, this means $\exists \varepsilon > 0$, $\exists N$ s.t. $\forall m \ge N$, $\mathrm{Var}(\mu_m) \ge \varepsilon$ ← wlog, reindex sequence so $N = 0$

then take above eq'n,

$$\mathrm{Var}(\mu_{m+1}) \le \mathrm{Var}(\mu_m) - \alpha_m \underbrace{\mathrm{Var}(\mu_m)}_{\ge \varepsilon} + \alpha_m^2$$

$$\le \mathrm{Var}(\mu_m) - \alpha_m \varepsilon + \alpha_m^2$$

$$\le \mathrm{Var}(\mu_{m-1}) - \varepsilon(\alpha_m + \alpha_{m-1}) + \alpha_m^2 + \alpha_{m-1}^2 \quad \text{(recursed)}$$

$$\cdots$$

$$\le \mathrm{Var}(\mu_0) - \varepsilon \underbrace{\sum_{k=0}^{m} \alpha_k}_{\to +\infty} + \underbrace{\sum_{k=0}^{m} \alpha_k^2}_{\text{bounded}}$$

$\Rightarrow \lim \mathrm{Var}(\mu_{m+1}) = -\infty$, impossible.

Now, since $\sum \alpha_m^2 < \infty \Rightarrow \alpha_m \to 0$, so $\exists M$ s.t. $\alpha_m < \varepsilon$ $\forall m \ge M$

So, $\exists m \ge M$ s.t. $\mathrm{Var}(\mu_m) < \varepsilon$ ($\underline{\text{not}}$ $\forall m \ge M$... that's what we need to show)

Again, use $\mathrm{Var}(\mu_{m+1}) \le (1-\alpha_m) \underbrace{\mathrm{Var}(\mu_m)}_{<\varepsilon} + \alpha_m^2$ $\quad$ $\alpha_m^2 = \alpha_m \cdot \alpha_m < \varepsilon \cdot \alpha_m$

$$< (1-\alpha_m)\varepsilon + \alpha_m \varepsilon = \varepsilon$$

$\Rightarrow \forall m \ge M$, $\mathrm{Var}(\mu_m) < \varepsilon$ $\square$

Stochastic Approximation

Solves things like $x = H(x)$ where $H(x)$ can't be queried directly, but we can observe noisy measurements, $y_i = H(x_i) + w_i$, $\mathbb{E}[w_i] = 0$

we already saw in SGD chapter. SAA (aka ERM) is a special case where $w \sim \mathrm{Uniform}[m]$

Ex: $\min f(x) := \mathbb{E} f(x,w)$ w.r. $\mathbb{E}$ over w

If f is convex and smooth, minimizer is characterized by $0 = \nabla f(x)$

i.e., $x = \underbrace{(I - \nabla f)}_{H}(x)$. Under some conditions (e.g., discrete distribution, or, $\nabla f(x,w)$ is uniformly Lipschitz cts $\forall w$) then $\nabla f(x) := \nabla \mathbb{E} f(x,w) = \mathbb{E} \nabla f(x,w)$

So if we can sample w, we can sample ∇f and hence H

So, iteration

$$X_{t+1} = (1-\alpha_t) X_t + \alpha \underbrace{(H(X_t) + w_t)}_{\text{"}X_m\text{" w.r. "}\mathbb{E}[X_m] = H(x_t)\text{"}} \quad \text{looks like Thm 17.4}$$

$$= X_t + \alpha_t (H(X_t) + w_t - X_t)$$

$$= X_t + \alpha_t D(X_t, w_t) \quad \text{more generally}$$

("μ_m")

Background:

Martingale (discrete-time, but can be defined for cts. time also)

$(X_t)_{t \in \mathbb{N}}$ a sequence of r.v. w.r. $\mathbb{E}[|X_t|] < \infty$

s.t. $\mathbb{E}[X_{n+1} \mid X_1, X_2, \ldots, X_n] = X_n$ $\quad$ Ex. a random walk

" ——— " $\le X_n$ $\quad$ supermartingale

" ——— " $\geq X_n$ Submartingale

" $= 0$ Martingale difference

ex: if (X_n) is a Martingale, $Y_n = X_n - X_{n-1}$ is a Martingale Difference Sequence

Fact (Thm 17.5, supermartingale convergence)

$(X_t)_{t\in\mathbb{N}}, (Y_t)_{t\in\mathbb{N}}, (Z_t)_{t\in\mathbb{N}}$ non-neg. r.v. s.t. $\sum Y_t < \infty$, and $\mathcal{F}_t$ the filtration generated by all r.v. for $t' \leq t$

Then if $\mathbb{E}[X_{t+1} | \mathcal{F}_t] \leq X_t + Y_t - Z_t$, then

① X_t converges w.p.1 (i.e., $Y_t, Z_t = 0 \Rightarrow$ supermartingales converge)

② $\sum Z_t < \infty$

Use this to show Thm 17.16, 17.17 which generalizes our simple SA "LLN"-like theorem

... too far off topic, so see book (Mahsi et al.)

Temporal Difference algorithm (TD(0))

most Key (and novel) idea in RL, combining dynamic programming ideas w/ Monte Carlo (Sutton + Barto)

All based on estimating $V(s)$

(V is a function of the policy π, $V_\pi(s)$, but if we find $V^*(s)$ $(= V_{\pi^*}(s))$

then we can recover π, like in the greedy update of Policy Iteration, $\pi \leftarrow \operatorname{argmax}_\pi R_\pi + \gamma P_\pi V$)

Based on Eq 17.5 / 17.6a ("Bellman's Eq." but not his optimality one) — his linear ones

$$V_\pi(s) = \mathbb{E}\left[r(s,\pi(s)) \right] + \gamma \sum_{s'\in S} \mathbb{P}[s' | s, \pi(s)] \cdot V_\pi(s')$$

$$= \mathop{\mathbb{E}}_{s'}\left[r(s,\pi(s)) + \gamma \cdot V_\pi(s') \mid s \right]$$

matrix notation: i.e., $V_\pi = R_\pi + \gamma P_\pi V_\pi$

So $V_\pi = (I - \gamma P_\pi)^{-1} R_\pi$

... but $s' \sim \mathbb{P}[s' | s, \pi(s)]$ is unknown, but we can sample from it

[setting: a model of the environment is unknown, but we still interact w/ it]

We'll start w/ a guess for V (so $V(s)$ defined $\forall s \in S$), assume deterministic policies

- then sample a new state s' (environment does this for us, as a black box)
- reward $r(s, \pi(s))$ is also sampled (by the environment)
- we update policy value

$$V(s) \leftarrow (1-\alpha) V(s) + \alpha \left(r(s,\pi(s)) + \gamma V(s') \right)$$

$$= V(s) + \alpha \cdot \left(r(s,\pi(s)) + \gamma V(s') - V(s) \right)$$

Bellman's Linear Equation: $\mathbb{E}(r(s,\pi(s)) + \gamma V(s')) = V(s)$ ... So this should get small as V is close to being accurate

α depends on how many times we've visited state s before

temporal difference of V values "TD"

Algo: TD(0) Input: π

Initialize V arbitrary

For $t = 0, 1, \ldots, T$ ← "t" is now an epoch, like a trial or a game or experiment

S initialized arbitrarily/randomly (other than a terminal state) (new game)

for each time step within epoch
(rounds of the game)

observe ("sample") $r' = \text{Reward}(s, \pi(s))$ from environment (eg robot simulation)

observe ("sample") s' as our next state, from environment

$V(s) \leftarrow (1-\alpha)V(s) + \alpha(r' + \gamma V(s'))$

$s \leftarrow s'$

(Stop when reach a terminal state)

end

end

Finite-time MDP has a final state

(ex: in a board game, winning/losing is final state)

may not know a priori how many rounds it will last

For ∞-horizon, maybe approximate?

You can prove convergence via Thm. 17.17

$(\forall \pi),\ V(s) \to V_\pi(s)\ \forall s \in S$

... but how to choose π not clear. — both taken care of by Q-learning framework.

Q-learning

If we know optimal state-action value function $Q^*(s,a)$, then

1) it gives $V^*(s) = \max_{a \in A} Q^*(s,a)$

2) it gives $\pi^*(s) = \arg\max_{a \in A} Q^*(s,a)$

For simplicity, now assume deterministic reward function.

Bellman optimality equations, (Eq (17.4) + Eq (17.1))

$$Q^*(s,a) = \mathbb{E}[r(s,a)] + \gamma \sum_{s' \in S} \mathbb{P}[s'|s,a] V^*(s')$$

$$= \mathbb{E}_{s'}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a') \right] \leftarrow (*)$$

ALGO: Q-learning

Initialize Q arbitrarily // no π needed

For epochs $t = 0, 1, \ldots, T$

Pick a state s // same as TD(0)

For each round in the epoch (implicitly depends on Q ★)

Choose action a based on π, s

Observe reward r' from environment

Observe state s' from environment

$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r' + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$

end when reach a terminal state

End

End

So, a bit like TD(0) but update Q, not V, and π doesn't need to be pre-specified

Update can also be written $Q(s,a) \leftarrow \underbrace{(1-\alpha)Q(s,a)}_{\text{old}} + \alpha\underbrace{\left(r(s,a) + \max_{a'} Q(s',a') \right)}_{\text{cf } (*)}$ Stochastic Approx.

① update rule

② learning policy ↳ Policy π

- For general convergence, this is arbitrary as long as every pair (s,a) is visited ∞-ly often

(...of course, at convergence, recover π^* via maximization in 2) above)

$$\pi^*(s) \in \operatorname*{argmax}_a Q^*(s,a)$$

- For efficient convergence,

$\pi(s) \in \operatorname*{argmax}_a Q_t(s,a)$ is natural choice ...

...but this is all exploitation and if

Q_t isn't optimal, we may be missing important (s,a) pairs

So

"ε-greedy policy" is to take this greedy step w.p. $1-\varepsilon$,

and take a random step (exploration) (eg; uniformly at random) w.p. ε

or ... Boltzmann exploration ... ("temperature" $\to 0$ as $t \to \infty$)

Many other variations on ① update rule and ② learning policy

- TD(λ)
- on-policy methods like SARSA
- See Sutton and Barto for a lot more

Thm 17.18 For a finite MDP, if $\forall s, \forall a$ $\sum_{t=0}^{\infty} \alpha_t(s,a) = +\infty$, $\sum_{t=0}^{\infty} \alpha_t^2(s,a) < \infty$ (for Q-learning)

$\alpha_t \in [0,1]$ then $Q \to Q^*$ a.s.

↑ t is epoch index, not rounds in the game.

i.e., $\alpha_t(s,a) = \begin{cases} \alpha_t(s_t,a_t) & \text{if } s=s_t, a=a_t \\ 0 & s \neq s_t \text{ or } a \neq a_t \end{cases}$

So $\sum_t \alpha_t(s,a) = \infty \ \forall (s,a) \Rightarrow$ every (s,a) visited ∞-ly often.

Proof sketch: use Thm 17.17. See Mohri

Large State Spaces

Still a problem, as they were for planning algorithms

Very active research

Example approach, map $\Phi: S \to \mathbb{R}^N$ ($N \approx 100$, $|S| \approx 10^{100}$)

$V_\pi(s) \approx f_w(s)$,

$f_w(s) = \langle w, \underline{\Phi}(s) \rangle$ for example.

Learn $\underline{w}$