policy gradients

goal: derive a model-free algorithm that searches directly over policy parameters

Recht 2018 Part 9

Williams 1992

Fazel, Ge, Kakade, Mesbahi 2018

Kakade 2001

Schulman, Levine, Moritz, Jordan, Abbeel 2015

Mania, Guy, Recht 2018

- suppose we apply parametric, randomized policy $\mu : X \times \Psi \to \Delta(U)$ to SDE $x^+ \sim P(x,u)$
  - this determines a random process $u \sim \mu(x;\psi)$, $x^+ \sim P(x,u)$ with samples / roll-outs / trajectories

$$\tau : [0,t) \to X \times U$$
$$: s \mapsto (x_s, u_s)$$

- express probability distribution over trajectories i.t.o. P, μ

- $P_\psi(\tau) = \prod_{s=0}^{t-1} P(x_{s+1} | x_s, u_s) \cdot \mu_\psi(u_s | x_s)$

- suppose we want to choose $\psi \in \Psi$ to minimize expected finite-horizon cost,

$$E_{P_\psi}[c(\tau)] = E_{P_\psi}\left[\sum_{s=0}^{t} \mathcal{L}(s, x_s, u_s)\right],$$

i.e. we want to solve $\min_{\psi \in \Psi} E_{P_\psi}[c(\tau)]$

- let's see what happens when we apply gradient descent:

$$D_\psi E[c(\tau)] = D_\psi \int c(\tau) \cdot P_\psi(\tau)\, d\tau$$

$$= \int c(\tau) \cdot D_\psi P_\psi(\tau)\, d\tau$$ - assuming Leibniz' rule applies

$$= \int c(\tau) \cdot \left(\frac{D_\psi P_\psi(\tau)}{P_\psi(\tau)}\right) P_\psi(\tau)\, d\tau$$ - assuming $P_\psi(\tau) \neq 0$

$$= \int \left(c(\tau) \cdot D_\psi \log P_\psi(\tau)\right) \cdot P_\psi(\tau)\, d\tau$$ - substitution

$$= E_{P_\psi}[c(\tau) \cdot D_\psi \log P_\psi(\tau)]$$ – def. of expectation

- here's the magic:

→ compute $D_\psi \log P_\psi(\tau)$, simplify

- $\log P_\psi(\tau) = \log \prod_{s=0}^{t-1} P(x_{s+1} | x_s, u_s) \cdot \mu_\psi(u_s | x_s)$

$$= \sum_{s=0}^{t} P(x_{t+1} | x_t, u_t) + \sum_{s=0}^{t} \mu_\psi(u_t | x_t)$$

- so $D_\psi \log P_\psi(\tau) = \sum_{s=0}^{t-1} D_\psi \mu_\psi(u_s | x_s)$

* we only need to know derivative of the policy, <u>not</u> the system model !

* furthermore, Monte Carlo gives us:

$$E_{P_\psi}[c(\tau) \cdot D_\psi \log P_\psi(\tau)]$$
$$\simeq \frac{1}{N} \sum_{n=1}^{N} c(\tau_N) \cdot D_\psi \log P_\psi(\tau_N)$$

∘ if this seems like magic: it is...

- same derivation applies to any

- same derivation applies to any (even deterministic!) optimization problem:

instead of $\min_{u \in \mathcal{U}} c(u)$

consider $\min_{\mu \in \Delta(\mathcal{U})} E_\mu[c(u)]$

where we've randomized the policy

- if we then restrict to policies parameterized by $\psi \in \Psi$ then exactly the same derivation yields

$$D_\psi E_\psi[c(u)] = E_\psi[c(u) \cdot D_\psi \log \mu_\psi(u)]$$

- applying Monte Carlo to approximate this gradient is termed the "REINFORCE" algorithm

NOTE: randomization and parameterization both imply any solution we obtain is sub-optimal, i.e. provides an upper bound on

certain is sub optimal, i.e.
provides an upper "bound" on
actual minimal cost achievable

- since, at the end of the day, this is a "gradient"-based method,
("scare quotes" added because the desired gradient does exist, but is only ever (poorly) approximated)
there are obvious variations based on techniques that (can) improve on steepest descent:

  - Newton-Raphson method uses the Hessian to improve scaling / avoid chattering:

$$\psi^+ = \psi - \left[D^2_\psi E_\psi[c(\tau)]\right]^{-1} \cdot D_\psi E_\psi[c(\tau)]$$

termed "natural policy gradient"
Kakade 2001

- trust-region methods choose step size by solving auxiliary problem:

$$\psi^+ = \psi + \min_{\|\delta\| \le 1} D_\psi E_\psi[c(\tau)] \cdot \delta$$

where the norm $\|\cdot\|$ chosen could be:

$\|\cdot\|_2$ (standard 2-norm): steepest descent

$\|\cdot\|_{D^2_\psi E_\psi[c(\tau)]}$ (Hessian norm): natural P.G.

... or another of your choosing

termed "trust-region policy optimization" (TRPO)

Schulman et al 2015

code: github.com/rll/rllab