The simplest RL in an MDP

**Notation**

- in $X_t$, choose $A_t$
- get $(X_{t+1}, R_{t+1}) \sim P(\cdot | X_t, A_t)

New state reward

IID

likely not known

Q: estimate

$$
V(\pi) = E \left[ \sum_{t=0}^{\infty} Y_t R_{t+1} \left| X_0 = x \right. \right]
$$

here you act under $\pi$

Simplest

MC estimate

1. Start at $x$, run $\pi$, observe reward
2. This assumes MDP has a terminal state
3. Do this many times, average.
I could form \[ \frac{1}{\# \text{offs}} \sum (\text{Reward of seqs starting at } x_i) \]

but what if I form get a new seq.

Instead

\[ V(x) \rightarrow \hat{V}(x) + \alpha \left[ \text{Reward of new seq starting at } x \right] \]

\[ - \hat{V}(x) \]

(Easy) This is unbiased est.

(Easy) But may not be very efficient

Because we have ests of \[ \hat{V}(\text{something in } x_i's \text{ future}) \]
\[ V(x_t) \text{ is a seq. by sum of discounted rewards along seq.} \]

But we have (say) \( \hat{V}(x_t) \),

So another est is
\[ \hat{V}(x_t) \text{ is } R + \gamma \hat{V}(x_{t+1}). \]

This has to be unbiased, and might be better.

**TD(0)**

\[ S_{t+1} = R_{t+1} + \gamma \hat{V}(x_{t+1}) - \hat{V}(x_t) \]

\[ \hat{V}(x) = \hat{V}(x_0) + \alpha \sum_{t=1}^{\infty} \mathbb{1} [x_t = x] \]
Assume you have

\[(X_t, R_{t+1}, Y_{t+1})\]

where

\[X_t \sim \text{some ab. ergodic MC over } X\]

\[(Y_{t+1}, R_{t+1}) \sim P(\cdot | X_t, A_t)\]

\[\text{from the policy}\]

Then

\[S_{t+1} = R_{t+1} + \sum \hat{V}(Y_{t+1}) - \hat{V}(X_t)\]

Works.

\[\text{Which is better?}\]

Which is better?\[\text{depends}\]

Example in Szepesvari, p 22

Stress small change in model that favors either TDC(0), MC, depending on state.
Notice that TD(\(\lambda\)) and MC are two points on a family of algos.

MC: est by counting reward from a start

est by multiple steps of reward,
then pass to est

\[ = \text{TD}(\lambda) \]

TD(0): est by reward from one step, then add est of new state

TD(\(\lambda\)) - eqns p 24 of Szepesvari
What if there are too many states? e.g. continuous, etc.

- estimate $\hat{V}(s)$ with some form of function approximator. (Neural network these days)

$TD(x)$ then updates $\theta$

rather than $V$ directly

(Stepesvari, p21 gives eqns.)

→ this can diverge
two fairly natural strategies.

- Build a model of the Q function

\[ Q(x, a) = E_{\text{best policy}} \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid X_0 = x, A_0 = a \right] \]

- Particularly nice if I can do this without looking at sequences

\[ S_{t+1} = R_{t+1} + \gamma \max_{a' \in A} Q(Y_{t+1}, a') - Q(x_t, A_t) \]

\[ Q(x, a) \rightarrow Q(x, a) + \alpha S_{t+1} \mathbb{I}\{x = x, a = A\} \]

**Notice**: - TD update

- \((Y_{t+1}, R_{t+1}) \sim p(\cdot \mid x_t, A_t)\)

- But we don't need a seq (as long as \(x_t\) is ergodic.)
*e.g.*, you can build a local model, which is rather nice for many problems.

**Alternative:**

- Build a model of policy, update.

**Policy model:**

- **Model:**
  \[ \pi_{\theta} = p(c_t | s_t, \theta) \]  
  (common in some form of neural network predicting means, var, log stds of a Gaussian policy)

**Update by gradient ascent:**

\[ \theta_{k+1} = \theta_k + \alpha \nabla_{\theta} J(\pi_{\theta}) \bigg|_{\theta_k} \]

*expected return*
\[ J(\pi_\theta) = \mathbb{E}_{\pi_\theta} \left[ R(t) \right] \]

\[ \text{expected return under } \pi_\theta \]

Assume \( y_0 \)

Write \( \tau = (s_0, a_0, s_1, \ldots, s_{T+1}) \) as trajectory

\[ P(\tau | \theta) = \mathbb{P}(s_0) \prod_{t=0}^{T} \mathbb{P}(s_{t+1} | s_t, a_t) \cdot \pi_\theta(a_t | s_t) \]

prior of starting in 0

Notice \( \nabla_\theta \log f(\theta) = \frac{1}{f(\theta)} \cdot \nabla_\theta f(\theta) \)

(logarithmic gradient reinvented!)
\[
\log P(c|\theta) = \log P_0 + \sum_{t=0}^{T} \left\{ \log P(s_{t+1}|a_t, s_t) + \log \pi_\theta(a_t|s_t) \right\} \\
\text{NO } \theta!
\]

So:
\[
\nabla_\theta \log P(c|\theta)
= \sum_{t=0}^{T} \left\{ \log \pi_\theta(a_t|s_t) \right\}
\]

So:
\[
\nabla_\theta J = \nabla_\theta \int_{c} P(c|\theta) R(t)
= \int_{c} \nabla_\theta P(c|\theta) R(t)
= \int_{c} P \nabla_\theta \log P R(t)
= E \left[ \sum_{t=0}^{\infty} \theta \log \pi_\theta(a_t|s_t) R(c_t) \right]
\]

\text{But this is reward for whole trajectory}

\text{Expectation}
\[ \nabla \theta J = E \left[ \sum_{t=0}^{T} \nabla \theta \log \Pi \left( \sum_{i \approx t} R(S_u, a_u, S_{u+1}) \right) \right] \]

This is a better estimate:

we had \[ \nabla \theta J = E \left[ \frac{1}{T} \left( s_1(\alpha) + s_2(\alpha) \right) \right] \]

But \[ E_{\alpha} [s_1(\alpha)] = 0 \]

→ removing \( f \) reduces variance - this is not just manipulation for showing off.
Always assume $E[g_f] + g_f$ and need to estimate $E[\mathcal{L}]$ to use $\mathcal{L}$. It's easy to integrate, but (or should but it isn't necessarily) notice that it adds large terms to $f$ worse if $f$.
Now
\[ E_{a \sim \pi} \left[ \nabla_{a} \log \Pi(a_{t} | s_{t}) \right] \cdot b(s_{t}) \]

\[ = \mathbb{E}_{a \sim \pi} \left[ P(a_{t} | s_{t}) \cdot \nabla_{a} \log \Pi(a_{t} | s_{t}) \right] \cdot b(s_{t}) \]

\[ \downarrow \]

\[ \Pi(a_{t} | s_{t}) \]

But
\[ \int P \nabla_{a} \log P = \int \nabla_{a} \log P = \nabla_{a} \int P = 0 \]

\[ \therefore E_{a \sim \pi} \left[ \nabla_{a} \log \Pi(a_{t} | s_{t}) \cdot b(s_{t}) \right] = 0 \]

This means
\[ E_{a \sim \pi} \left[ \nabla_{a} \log \Pi(a_{t} | s_{t}) \cdot b(s_{t}) \right] = 0 \]

often referred to as a baseline
Natural choice of evidence

\[ \Pi \]

\[ \sqrt{V(s_t)} \quad \text{rest as above} \]

\[ \nabla_{\Theta} J(\Pi_{\Theta}) = \mathbb{E} \left[ \sum_{t=0}^{T} \nabla_{\Theta} \log \Pi(a_t | s_t) \left( \sum_{u=t}^{T} R(s_u, a_u, s_{u+1}) - V(s_u) \right) \right] \]

other base lines

\[ \Pi \]

\[ Q(s_t, a_t) \quad \text{works} \]

\[ \Pi \]

\[ Q(s_t, a_{t+1}) - V(s_t) \quad \text{works} \]

Which is best?

Don't know
BE CAREFUL
- the reason there are lots of methods is they mostly DON'T WORK RELIABLY.

- Best cases seem to be ones you could do with LQR, but the linearization is a nuisance.
  - velocity control of vehicle
  - stopping at right place from a single view of stop sign

- Important features: local reward is quite important
  - you can generate a ton of examples from simulator
  - car control is easy
You can test $\text{Var}[r]$, $\text{cov}[L]$ from MC replicates. to get variance reduction.

This suggests a simple way to get improved $\text{var}$ in Policy gradient, cause you know $\text{baseline}$.

This idea extends:

- under some circumstances, you don't need to know $r$
  (it must be easy to integrate, and a covariance could)
- you can use multiple control variates.
we want to know
\[ I_f = \int f \, p \, dx = \mathbb{E}_p \left[ s \right] \]

Assume we have \( g \), so \( \mathbb{E}_p [g] = \gamma \) \( \uparrow \text{known} \)

Then
\[ \frac{1}{N} \sum_i \left[ f(x_i) + c(g(x_i) - \gamma) \right] \leq I_f + \xi \]

Where
\[ \mathbb{E}[\xi] = 0 \]

\[ \text{Var}(\xi) = \text{Var}[\text{est of } f \text{ from } \frac{1}{N} \sum_i g(x_i)] \]
\[ + c^2 \text{Var}[\text{est of } \gamma \text{ from } \frac{1}{N} \sum_i g(x_i)] \]
\[ + 2c \text{ Cov}[\xi, \gamma] \]

Hence
\[ \text{best } c \left( = - \frac{\text{Cov}}{\text{Var}[\xi]} \right) \]

Usually unknown.