Markov Decision problems:
(processes)

(S, A, P, R, \pi, D)

Finiteset of states

Actions:
A: S \rightarrow \text{some discrete set}

Probability that next state is s
PC(S_{t+1} = s | S_t = S, \alpha(S_t) = \alpha)

Model

Policy
\pi: S \rightarrow A

(i.e. in state S, do this)
General, very useful model for action
+ control

+ Can be extended.
+ Note moral similarity to LQR.

(Where we treated had states actions = controls
But transitions were deterministic

Model
- Start in $s_0 \sim D$, $[\text{total reward}] = 0$
- Choose $a_i$
- $s_{i+1} \sim P(s_{i+1} | s_i, a_i)$
- Get reward $R(s_{i+1})$
- $[\text{total reward}] = \sum [\text{total reward}] + R(s_{i+1})$

Notice effect of discount. Long ago rewards have less weight.
Want to max $\sum_{t=0}^{\infty} R(s_t)$ discounted total reward
Natural Question:

- What is the best policy \( \pi^* \) for a given problem? (This might not be unique)

\[ \pi(s_1) = a, \quad \pi(s_2) = a_2 \]

Subordinate Question:

Assume policy is \( \pi \); what is the value of states?

\[ V(s) = E \left[ \text{Reward to be obtained by following } \pi \text{ from } s \right] \]

Where expectation is over \( P \).
Value is important, because it's linked to policy. Imagine we know \( V_{\pi}^*(s) \) then we can recover \( \pi^* \) choose action with best expected value of \( V_{\pi}^*(s) \).

Notice: \( a \) has value \( \sum_s P(s | s', a) V_{\pi^*}^*(s) \)

\[ V_{\pi^*}^*(s) = R(s) + \gamma \max_a \sum_s P(s' | s, a) V_{\pi^*}^*(s') \]

AND \( \pi^* (s) = \arg\max_a \sum_{s'} P(s' | s, a) V_{\pi^*}^*(s') \)

Bellman equation.
Estimating the optimal value function.

Value iteration:

\[ V^{(0)}(s) = 0 \]

update

\[ V^{(n+1)}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V(s') \]

Then

\[ V^{(\infty)} \rightarrow V \]

as \( k \rightarrow \infty \)

Proof:

write

\[ B[V] = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V(s') \]

Bellman Backup Operator.
Lemma below shows

\[
\max_{\theta} | B[V_{\theta}] (s) - B[V_{\theta'}] (s) | \leq \gamma \phi \max_{s} | V_{1}(s) - V_{2}(s) | \]

Note that

\[
B[V_{\pi^*}] = V_{\pi^*}
\]

Assume lemma

then \[ | B[V^{(n)}] - B[V_{\pi^*}] | \leq \gamma \frac{1}{n} | V^{(n)} - V_{\pi^*} | \leq \gamma | V - V_{\pi^*} | \]

and \[ \exists \gamma \leq \frac{1}{n} \]

so \[ V = B[V^{(n)}] \] is closer to \( V_{\pi^*} \) than \( V^{(n)} \)

\[ \square \]
Lemma:

\[ |B[V_1] - B[V_2]|_\infty \]
\[ = \gamma \left[ \max_a \sum_{s'} P(s'|s,a) V_1(s') \right] - \left[ \max_{s'} \sum_a P(s'|s,a) V_2(s') \right] \]
\[ = A \]
\[ \geq \gamma \max_a \sum_{s'} P(s'|s,a) V_1(s') - \sum_{s'} P(s'|s,a) V_2(s') \]
\[ \geq \gamma \max_a \left\{ \max_{s'} P(s'|s,a) V_1(s') - \max_{s'} P(s'|s,a) V_2(s') \right\} \]

\[ \text{because } \max_{x} f(x) - \max_{x} g(x) \leq \max_{x} |f(x) - g(x)| \]

\[ A = \gamma \max_a \sum_{s'} P(s'|s,a) [V_1(s') - V_2(s')] \]
\[ \leq \gamma \max_s |V_1(s) - V_2(s)| \]

(\( \gamma \) is non-neg, and sums to 1)
Convergence:

Assume rewards bounded by $R_{\text{max}}$

$$V^*(s) \leq \sum_{\ell=0}^{\infty} r^\ell R_{\text{max}} = \frac{R_{\text{max}}}{1-r}$$

So

$$\max_{s \in S} |V^\ell(s) - V^*(s)| \leq \frac{r^\ell R_{\text{max}}}{1-r}$$

(i.e. linear; faster for small $r$)

Asynchronous Version:

- Notice what is above assumes all states are updated together.

- i.e. compute $\tilde{V}(s) = B[V^\ell]$

  Then $V^{(n+1)} = \tilde{V}(s)$

- But you could visit states in turn, updating each

  - Converges
  - Usually better
Policy Iteration:

1. Initialize \( \pi \) with policy, perhaps random \( \pi^{(0)} \)
2. Compute \( V_{\pi} \)

\[
V_{\pi}(s) = R(s) + \sum_{s'} P(s' | s, \pi(s)) V_{\pi}(s')
\]

\[
V_{\pi} = R + \gamma T V_{\pi}
\]

\[
(I - \gamma T) V_{\pi} = R.
\]

\[
V_{\pi} = \left[ I + \gamma T + \gamma^2 T^2 + \gamma^3 T^3 + \ldots \right] R
\]

*Neumann Series*

3. Notice this involves solving a linear system

4. Update \( \pi \) to be greedy wrt \( V_{\pi} \)

\[
\pi(s) = \arg \max_a \sum_{s'} P(s' | s, a) V_{\pi}(s')
\]

5. Iterate
Policy iteration converges
- to exact optimal policy
- eventually.

In practice, policy iteration is usually preferred, particularly if transition probabilities are sparse.

MDP's as LP's:

\[ V(s) = R(s) + \gamma \max_{a} \sum_{s'} P(s'|s,a)V(s') \]

implies:

\[ V(s) > R(s) + \sum_{s'} P(s'|s,a)V(s') \]

one constraint for each \( A \)
So we can seek
\[
\min \sum_s V(s) \\
\text{s.t. } V(s) \geq R(s) + \gamma \sum_{s'} p(s'|s,a) V(s') \quad \text{one for each } s, a
\]
and this is the optimal value fn.

Write
\[
\begin{align*}
&\quad \text{V} \\
&\text{s.t. } (I - Ta) V - R \geq 0 \\
&\quad \text{a = 1 .. #A}
\end{align*}
\]

\( V \) is \( #S \) vector
\( \text{one } Ta \text{ per } a \).

Now consider the dual

\[ \lambda^T a \]
\( \lambda \) \text{WM, } #S \text{ dimensional, } 1 \text{ per } a

\[
\begin{align*}
\max \quad &\sum_a \lambda^T a R \\
\text{s.t. } &\sum_a (I - Ta)^T \lambda a + 1 = 0 \\
&\lambda \geq 0
\end{align*}
\]
What are dual vars?

1) Notice one useful interpretation of $\gamma$.

$$(1 - \gamma) = P[\text{MDP stops}]$$

$$E[\text{reward}] = \text{reward after first step} \cdot \text{reward at start} P(1 \text{ step})$$

$$+ (2) \cdot P(2)$$

$$+ \ldots$$

$$= R(s_0) + R(s_1)\gamma + R(s_2)\gamma^2 + \ldots$$

2) Write primal objective

$$\begin{align*}
\max & \quad \sum_a \lambda_a R \\
\text{s.t.} & \quad \sum_a (1 - T_a)^T \lambda_a + \frac{1}{151} = 0 \\
& \quad \lambda_a \geq 0
\end{align*}$$

Dual becomes

$$\begin{align*}
\max & \quad \sum_a \lambda_a R \\
\text{s.t.} & \quad \sum_a (1 - T_a)^T \lambda_a + \frac{1}{151} = 0 \\
& \quad \lambda_a \geq 0
\end{align*}$$
Now consider the expected number of times we are in $S$, take a

$$N(s, a) = \frac{1}{1s1}$$

expected # of times we are in arrive in $S$

is

$$\frac{1}{1s1} + \gamma \sum_{s'} \sum_{a} P(s' | s, a) N(s', a)$$

Because we have a uniform prob

first state is $s$

joint Prob

$P(s' | s, a)$

Notice:

$$\sum_{a} N(s, a) = \text{expected number of times we leave from } s$$

But

$$\sum_{a} N(s, a) = \frac{1}{1s1} + \gamma \sum_{s'} \sum_{a} P(s' | s, a) N(s, a)$$

These are the constraints on dual Vars!
\[ \lambda_n = \lambda_n(s) = N(s, a). \]

These encode the optimal policy:

\[ \Pi_\pi(s) = \text{take the most common action at } s \]

- argmax \[ N(s, a) \]

Natural follow up questions

- What happens if you don't know \( P, R \)?
  - (Reinforcement learning: roughly: act, see what happens; keep notes)

- What happens if you don't know \( P, R \) but see a skilled actor?
  - (our problem)