Bayesian Reinforcement learning

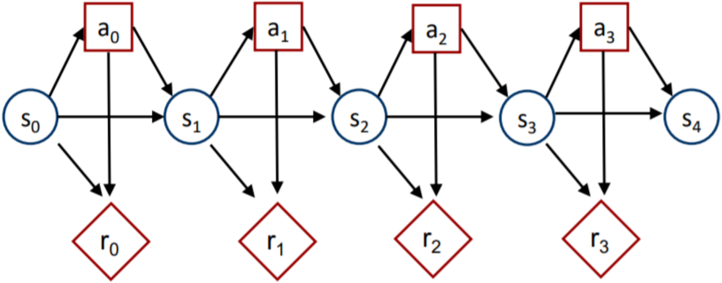
Salim Amoukou & Raphael Cousin

April 19, 2019

Markov Decision Process

Markov process augmented with :

- Action
- Reward



Goal: Control action to maximize rewards: $\sum_t R(s_t, a_t)$

Infinite and deterministic rewards ?

Question : But what if the process is infinite ?

- Solution : Discounted rewards (or average reward)
  - discounted factors: $0 < \gamma < 1$
    - $\sum_t \gamma^t R(s_t, a_t)$

Infinite and deterministic rewards ?

Question : But what if the process is infinite ?

- Solution : Discounted rewards (or average reward)
  - discounted factors: $0 < \gamma < 1$
    - $\sum_t \gamma^t R(s_t, a_t)$

Questions : In general case, the reward is stochastic.

- Solution : In practice, we will work with the average.

Formal Markov decision Process

- Set of states : S
- Set of actions : A
- Transition model : $\mathbb{P}(s_t|s_{t-1}, a_{t-1})$
- Reward model : $R(s_t, a_t)$
- Discounted factor : $0 < \gamma < 1$
- Horizon :
  - h $< \infty$ or h $= \infty$

Goal: find optimal policy $\pi : S \rightarrow A$ which maximizes $\sum_t \gamma^t R(s_t, a_t)$

What is the best policy ?

- Value function: $V^\pi(s_0) = \sum_t \gamma^t \sum_{s_t} \mathbb{P}(s_t|s_0, \pi) R(s_t, a_t)$

What is the best policy ?

- Value function: $V^{\pi}(s_0) = \sum_t \gamma^t \sum_{s_t} \mathbb{P}(s_t|s_0, \pi) R(s_t, a_t)$
- Optimal policy : The policy with the best value function ie :

$$V^{\pi^*}(s_0) \geq V^{\pi}(s_0) \forall \pi, s_o$$

How we find the best policy ?

Idea: Think backward...

How we find the best policy ?

Idea: Think backward...

- Best value in the last step: horizon : $V(s_h) = max_{a_h} R(s_h, a_h)$

How we find the best policy ?

Idea: Think backward...

- Best value in the last step: horizon : $V(s_h) = max_{a_h} R(s_h, a_h)$
- Best value time step left iteratively :
  $V(s_{h-1}) = max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \sum_{s_h} \mathbb{P}(s_h | s_{h-1}, a_{h-1}) V(s_h)$
  .
  .
  .

How we find the best policy ?

Idea: Think backward...

- Best value in the last step: horizon : $V(s_h) = max_{a_h} R(s_h, a_h)$
- Best value time step left iteratively :
  $V(s_{h-1}) = max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \sum_{s_h} \mathbb{P}(s_h | s_{h-1}, a_{h-1}) V(s_h)$
  .
  .
  .
- Bellman's equation :
  $V(s_t) = max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1} | s_t, a_t) V(s_{t+1})$

How we find the best policy ?

Idea: Think backward...

- Best value in the last step: horizon : $V(s_h) = max_{a_h} R(s_h, a_h)$
- Best value time step left iteratively :
  $V(s_{h-1}) = max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \sum_{s_h} \mathbb{P}(s_h | s_{h-1}, a_{h-1}) V(s_h)$
  .
  .
  .
- Bellman's equation :
  $V(s_t) = max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1} | s_t, a_t) V(s_{t+1})$
- Extract the best policy :
  $a_t^* = argmax_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1} | s_t, a_t) V(s_{t+1})$

How we find the best policy ?

Idea: Think backward...

- Best value in the last step: horizon : $V(s_h) = max_{a_h} R(s_h, a_h)$
- Best value time step left iteratively :
  $V(s_{h-1}) = max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \sum_{s_h} \mathbb{P}(s_h|s_{h-1}, a_{h-1})V(s_h)$
  .
  .
  .
- Bellman's equation :
  $V(s_t) = max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|s_t, a_t)V(s_{t+1})$
- Extract the best policy :
  $a_t^* = argmax_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|s_t, a_t)V(s_{t+1})$

Remarks : When h is finite, the policy is non stationary.
How do we do when h is infinite ?

If h is infinite

Idea : If h $\to +\infty$, $V_h^\pi \to V_\infty^\pi$ et $V_{h-1}^\pi \to V_\infty^\pi$

- Policy evaluation : $V_\infty^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s, \pi(s)) V_\infty^\pi(s)$ $\forall s$

If h is infinite

Idea : If h $\to +\infty$, $V_h^\pi \to V_\infty^\pi$ et $V_{h-1}^\pi \to V_\infty^\pi$

- Policy evaluation : $V_\infty^\pi(s) = R(s,\pi(s)) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s,\pi(s))V_\infty^\pi(s)$ $\forall s$
- Matrix form : $V = R + \gamma TV$ so we just have to solve this linear equation

If h is infinite

Idea : If h $\to +\infty$, $V_h^\pi \to V_\infty^\pi$ et $V_{h-1}^\pi \to V_\infty^\pi$

- Policy evaluation : $V_\infty^\pi(s) = R(s,\pi(s)) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s,\pi(s))V_\infty^\pi(s)$ $\forall s$
- Matrix form : $V = R + \gamma TV$ so we just have to solve this linear equation
- Solver : Richardson iteration
  - Repeat $V = R + \gamma TV$

How is it working ?

Let $H(V) = R + \gamma TV$ the operator of the policy evaluation solver then:

How is it working ?

Let $H(V) = R + \gamma TV$ the operator of the policy evaluation solver then:

Lemma

H is a contraction mapping:

$$||H(\tilde{V}) - H(V)||_\infty \leq \gamma||\tilde{V} - V||_\infty$$

Idea of proof : T is transition matrix

How is it working ?

Let $H(V) = R + \gamma TV$ the operator of the policy evaluation solver then:

Lemma

H is a contraction mapping:

$$||H(\tilde{V}) - H(V)||_\infty \leq \gamma||\tilde{V} - V||_\infty$$

Idea of proof : T is transition matrix

Theorem

Policy evaluation converges to V^π for any estimate V :

$$\lim_{n\to\infty} H^{(n)}(V) = V^\pi$$

Idea of proof : By definition $V^\pi = H^{(\infty)}(0)$ then conclude with lemma.

Value iteration

One could show that previous statements hold for :
$H^*(V) = max_a R^a + \gamma T^a V$ then,

- $\forall V \lim_{n\to\infty} H^{*^{(\infty)}}(V) = H^{*^{(\infty)}}(0) = V^*$ the optimal value function

Value iteration

One could show that previous statements hold for :
$H^*(V) = max_a R^a + \gamma T^a V$ then,

- $\forall V \lim_{n\to\infty} H^{*^{(\infty)}}(V) = H^{*^{(\infty)}}(0) = V^*$ the optimal value function

valueIteration(MDP)

$V_0^* \leftarrow \max_a R^a$; $\quad n \leftarrow 0$
Repeat
$\quad n \leftarrow n+1$
$\quad V_n \leftarrow \max_a R^a + \gamma T^a V_{n-1}$
Until $||V_n - V_{n-1}||_\infty \leq \epsilon$
Return V_n

Value iteration

One could show that previous statements hold for :
$H^*(V) = max_a R^a + \gamma T^a V$ then,

- $\forall V \lim_{n \to \infty} H^{*^{(\infty)}}(V) = H^{*^{(\infty)}}(0) = V^*$ the optimal value function

valueIteration(MDP)

$V_0^* \leftarrow \max_a R^a \,; \qquad n \leftarrow 0$

Repeat

$\qquad n \leftarrow n + 1$

$\qquad V_n \leftarrow \max_a R^a + \gamma T^a V_{n-1}$

Until $||V_n - V_{n-1}||_\infty \leq \epsilon$

Return V_n

- The optimal policy will be :
  $\pi_n(s) = argmax_a R(s,a) + \gamma \sum_{s'} \mathbb{P}(s'|s,a) V_n(s')$

Some remarks

- How far is V^{π_n} to V^* ?

Some remarks

- How far is V^{π_n} to V^* ?
  - $\rightarrow \frac{2\epsilon}{1-\gamma}$

Some remarks

- How far is V^{π_n} to V^* ?
  - $\rightarrow \frac{2\epsilon}{1-\gamma}$
- Is this fine computationally ?

Some remarks

- How far is V^{π_n} to V^* ?
  - $\rightarrow \frac{2\epsilon}{1-\gamma}$
- Is this fine computationally ?
  - Sol: Policy iteration, modified policy iteration

Some remarks

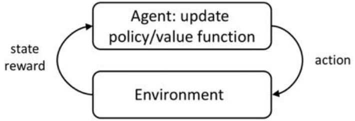
- How far is V^{π_n} to V^* ?
  - $\rightarrow \frac{2\epsilon}{1-\gamma}$
- Is this fine computationally ?
  - Sol: Policy iteration, modified policy iteration
- Optimize directly the policy : policy gradient

Reinforcement learning

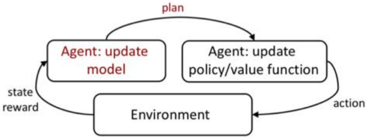
- Set of states : S
- Set of actions : A
- Transition model : $\mathbb{P}(s_t, r_t | s_{t-1}, a_{t-1})$ unknown
- Environement : samples state s and reward r
- Discounted factor : $0 < \gamma < 1$
- Horizon :
  - h $< \infty$ or h $= \infty$

Goal: find optimal policy $\pi : S \to A$ which maximizes $\sum_t R(s_t, a_t)$

Two solutions: Model free vs Model based

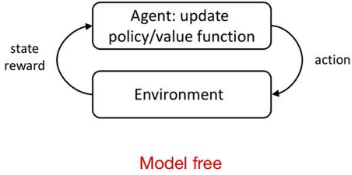


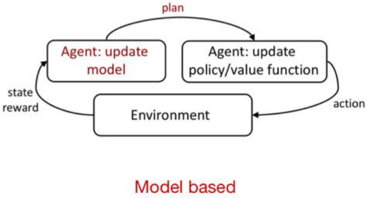
Model free



Model based

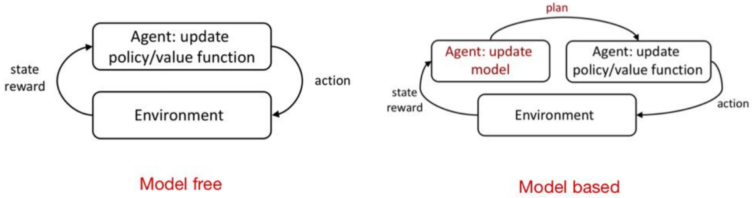
Two solutions: Model free vs Model based





- Model free : Unbiased + large data + low complexity
- Model based : Biased + less data + high complexity

Two solutions: Model free vs Model based



- Model free : Unbiased + large data + low complexity
- Model based : Biased + less data + high complexity

Remarks : In pratice, generally model based outperforms model free.

Example model free: Q-learning

Estimate the best value function by sampling :

- $V^*(s) = max_a \mathbb{E}(r|s,a) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s,a)V^*(s) \approx r + \gamma V^*(s^{'})$

Correct the value iteratively :

- $V_n^*(s) = V_{n-1}^*(s) + \alpha_n(r + \gamma V_{n-1}^*(s^{'}) - V_n^*(s))$

Example model free: Q-learning

Estimate the best value function by sampling :

- $V^*(s) = max_a \mathbb{E}(r|s,a) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s,a)V^*(s) \approx r + \gamma V^*(s^{'})$

Correct the value iteratively :

- $V_n^*(s) = V_{n-1}^*(s) + \alpha_n(r + \gamma V_{n-1}^*(s^{'}) - V_n^*(s))$

Drop the max, to have Q-function (value function according to initial state and action) :

- $Q_n^*(s,a) = Q_{n-1}^*(s,a) + \alpha_n(r + \gamma max_{a'} Q_{n-1}^*(s^{'}, a^{'}) - V_n^*(s))$

Example model free: Q-learning

Estimate the best value function by sampling :

- $V^*(s) = max_a \mathbb{E}(r|s,a) + \gamma \sum_{s'} \mathbb{P}(s^{'}|s,a)V^*(s) \approx r + \gamma V^*(s^{'})$

Correct the value iteratively :

- $V_n^*(s) = V_{n-1}^*(s) + \alpha_n(r + \gamma V_{n-1}^*(s^{'}) - V_n^*(s))$

Drop the max, to have Q-function (value function according to initial state and action) :

- $Q_n^*(s,a) = Q_{n-1}^*(s,a) + \alpha_n(r + \gamma max_{a'} Q_{n-1}^*(s^{'}, a^{'}) - V_n^*(s))$

Q-learning algorithm :

- Interact with the environment and update Q-function until convergence
- Extract policy at the end : $\pi(s) = argmax_a Q^*(s,a)$

Example model based

ModelBasedRL(s)

Repeat

Select and execute a

Observe s' and r

Update counts: $n(s,a) \leftarrow n(s,a) + 1$,

$n(s,a,s') \leftarrow n(s,a,s') + 1$

Update transition: $\Pr(s'|s,a) \leftarrow \frac{n(s,a,s')}{n(s,a)}\ \forall s'$

Update reward: $R(s,a) \leftarrow \frac{r + (n(s,a)-1)R(s,a)}{n(s,a)}$

Solve: $V^*(s) = \max_a R(s,a) + \gamma \sum_{s'} \Pr(s'|s,a)\, V^*(s')\ \forall s$

$s \leftarrow s'$

Until convergence of V^*

Return V^*

Example model based

ModelBasedRL(s)
- Repeat
  - Select and execute a
  - Observe s' and r
  - Update counts: $n(s,a) \leftarrow n(s,a) + 1$,
    $n(s,a,s') \leftarrow n(s,a,s') + 1$
  - Update transition: $\Pr(s'|s,a) \leftarrow \frac{n(s,a,s')}{n(s,a)} \; \forall s'$
  - Update reward: $R(s,a) \leftarrow \frac{r + (n(s,a)-1)R(s,a)}{n(s,a)}$
  - Solve: $V^*(s) = \max_a R(s,a) + \gamma \sum_{s'} \Pr(s'|s,a)\, V^*(s') \; \forall s$
  - $s \leftarrow s'$
- Until convergence of V^*
- Return V^*

- In complex model ?
- Same problem : How do we explore ?

Bayesian reinforcement learning

Idea : Augmented state with distribution on unknowns parameters

Bayesian reinforcement learning

Idea : Augmented state with distribution on unknowns parameters

- Set of states : $(s, b) \in S \times B$
  - physical state : $s \in S$
  - belief state : $b \in B, \;\; b(\theta)$ the prior on θ
- Transition model : $\mathbb{P}(s', r', b' | s, a, b) \leftarrow$ **the model is known**
- Set of action : A
- Reward : $r \in \mathbb{R}$

Goal: find optimal policy $\pi : S \times B \to A$

Why the model is known ?

$\mathbb{P}(r, s', b'|s, b, a) = \mathbb{P}(r, s'|s, b, a)\mathbb{P}(b'|r, s', b, a)$

- $\mathbb{P}(r, s'|s, b, a) = \int P(r, s'|s, a, \theta)b(\theta)d\theta$
- $\mathbb{P}(b'|r, s', b, a)$ corresponds to the posterior

Why the model is known ?

$\mathbb{P}(r, s', b'|s, b, a) = \mathbb{P}(r, s'|s, b, a)\mathbb{P}(b'|r, s', b, a)$

- $\mathbb{P}(r, s'|s, b, a) = \int P(r, s'|s, a, \theta)b(\theta)d\theta$
- $\mathbb{P}(b'|r, s', b, a)$ corresponds to the posterior

Since the model is known, we just have to treat Bayesian RL like an MDP (belief-MDP, POMDP) ie:

- Solve : $V^*(s, b) = max_a \mathbb{E}[r|s, a] + \gamma \sum_{s'} \mathbb{P}(s'|s, b, a)V^*(s', b_{s,a,s'})$

Why the model is known ?

$\mathbb{P}(r, s', b'|s, b, a) = \mathbb{P}(r, s'|s, b, a)\mathbb{P}(b'|r, s', b, a)$

- $\mathbb{P}(r, s'|s, b, a) = \int P(r, s'|s, a, \theta)b(\theta)d\theta$
- $\mathbb{P}(b'|r, s', b, a)$ corresponds to the posterior

Since the model is known, we just have to treat Bayesian RL like an MDP (belief-MDP, POMDP) ie:

- Solve : $V^*(s, b) = max_a \mathbb{E}[r|s, a] + \gamma \sum_{s'} \mathbb{P}(s'|s, b, a)V^*(s', b_{s,a,s'})$

Some solutions :

- POMDP discretization (Jaulmes et al. 2005)
- BEETLE (Poupart et al. 2006)
- Thompson sampling (Strens 2000)

Thompson sampling in Bayesian RL

ThompsonSamplingInBayesianRL(s,b)

Repeat

Sample $\theta_1, \dots, \theta_k \sim \Pr(\theta) \;\; \forall a$

$Q^*_{\theta_i} \leftarrow solve(MDP_{\theta_i})$

$\hat{Q}(s,a) \leftarrow \frac{1}{k} \sum_{i=1}^{k} Q^*_{\theta_i}(s,a)$

$a^* \leftarrow \mathrm{argmax_a}\, \hat{Q}(s,a)$

Execute a^* and receive r, s'

$b(\theta) \leftarrow b(\theta)\Pr(r, s' | s, a, \theta)$

$s \leftarrow s'$

Source

- Reinforcement Learning - Rich Sutton's
- Pascal poubart's course CS885

SIMULATION TIME !

Question : Monte carlo update

- Let G_k be a one-trajectory Monte Carlo target
$$G_k = \sum_t \gamma^t r_t^{(k)}$$
- Approximate value function
$$\begin{aligned} V_n^\pi(s) &\approx \frac{1}{n(s)} \sum_{k=1}^{n(s)} G_k \\ &= \frac{1}{n(s)}\left(G_{n(s)} + \sum_{k=1}^{n(s)-1} G_k\right) \\ &= \frac{1}{n(s)}\left(G_{n(s)} + (n(s)-1)V_{n-1}^\pi(s)\right) \\ &= V_{n-1}^\pi(s) + \frac{1}{n(s)}\left(G_{n(s)} - V_{n-1}^\pi(s)\right) \end{aligned}$$
- **Incremental update**
$$V_n^\pi(s) \leftarrow V_{n-1}^\pi(s) + \alpha_n\left(G_n - V_{n-1}^\pi(s)\right)$$
learning rate $1/n(s)$

Question : Complexity

- Value Iteration:
  - Each iteration: $O(|S|^2|A|)$
  - Many iterations: linear convergence
- Policy Iteration:
  - Each iteration: $O(|S|^3 + |S|^2|A|)$
  - Few iterations: linear-quadratic convergence
- Modified Policy Iteration:
  - Each iteration: $O(k|S|^2 + |S|^2|A|)$
  - Few iterations: linear-quadratic convergence

Question : Complex model based

ModelBasedRL(s)

Repeat

Select and execute a, observe s' and r

Update transition: $w_T \leftarrow w_T - \alpha_T(T(s,a) - s')\nabla_{w_T}T(s,a)$

Update reward: $w_R \leftarrow w_R - \alpha_R(R(s,a) - r)\nabla_{w_R}R(s,a)$

Repeat a few times:

sample $\hat{s}, \hat{a}$ arbitrarily

$\delta \leftarrow R(\hat{s},\hat{a}) + \gamma \max_{\hat{a}'} Q(T(\hat{s},\hat{a}),\hat{a}') - Q(\hat{s},\hat{a})$

Update Q: $w_Q \leftarrow w_Q - \alpha_Q \delta \nabla_{w_Q} Q(\hat{s},\hat{a})$

$s \leftarrow s'$

Until convergence of Q

Return Q

Question : Complex model based

ModelBasedRL(s)

Repeat

  Select and execute a, observe s' and r

  Update transition: $w_T \leftarrow w_T - \alpha_T (T(s,a) - s') \nabla_{w_T} T(s,a)$

  Update reward: $w_R \leftarrow w_R - \alpha_R (R(s,a) - r) \nabla_{w_R} R(s,a)$

  Repeat a few times:

    sample $\hat{s}, \hat{a}$ arbitrarily

    $\delta \leftarrow R(\hat{s},\hat{a}) + \gamma \max_{\hat{a}'} Q(T(\hat{s},\hat{a}),\hat{a}') - Q(\hat{s},\hat{a})$

    Update Q: $w_Q \leftarrow w_Q - \alpha_Q \delta \nabla_{w_Q} Q(\hat{s},\hat{a})$

  $s \leftarrow s'$

Until convergence of Q

Return Q

modifiedPolicyIteration(MDP)

Initialize π_0 and V_0 to anything

$n \leftarrow 0$

Repeat

Eval: Repeat k times

$$V_n \leftarrow R^{\pi_n} + \gamma T^{\pi_n} V_n$$

Improve: $\pi_{n+1} \leftarrow argmax_a \ R^a + \gamma T^a V_n$

$$V_{n+1} \leftarrow max_a \ R^a + \gamma T^a V_n$$

$n \leftarrow n + 1$

Until $\left\| V_n - V_{n-1} \right\|_\infty \leq \epsilon$

Return π_n