Notes CME241

Raphael Abbou

January 27, 2020

1 Lectures 1-2: Markov Decision Processes

1.1 Definitions

A state S_t represent all the information at time t.

$$\mathbb{P}(S_{t+1}|S_t) = \mathbb{P}(S_{t+1}|S_1, ...S_t)$$

We define the state transition matrix by $\mathcal{P}_{ss'} = \mathbb{P}(S_{t+1} = s'|S_t = s)$

Definition 1.1. *Markov Process (MP):* $< \mathcal{S}, \mathcal{P} >$
Markov Reward Process (MRP): $< \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma >$
Markov Decision Process (MDP): $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$

For a MRP:
- $\mathcal{R}$ is a reward function: $\mathcal{R}_s = \mathbb{E}[R_{t+1}|S_t = s]$
- $\gamma \in [0, 1]$ is the discount factor
- $G_t = \sum_k \gamma^k T_{t+k+1}$ is the return

For a MDP:
- $\mathcal{A}$ is the set of actions
- The probabilities of transition depends on the action: $\mathbb{P}^a_{ss'}, \mathcal{R}^a_s$

Definition 1.2. *For MDP, a policy π is a distribution over actions given a state:*

$$\pi(a|s) = \mathbb{P}(A_t = a|S_t = s)$$

Definition 1.3. *For a MDP, we define:*
- *State-value function:*

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$$

- *Action-value function:*

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$$

A MDP with a policy π gives:
- A MP $< S, \mathcal{P}^\pi >$
- A MRP $< S, \mathcal{P}^\pi, \mathcal{R}^\pi >$

1.2 Bellman's Equations

We define Bellman's equation:

For a MRP:

$$v(s) = \mathcal{R}_s + \gamma \sum_{s'\ inS} \mathcal{P}_{ss'} v(s')$$

Or in its matrix form:

$$v = \mathcal{R} + \gamma \mathcal{P} v, v\ in \mathbb{R}^n$$

The resolution of this equation takes $O(n^3)$ (matrix inversion). For large MDP, it can be solved by DP, MC evaluation, or Temporal Difference learning.

For a MDP:

$$v_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_\pi$$

We are looking for:
- optimal state value $v^*(s) = \max_\pi v_\pi(s)$
- optimal action value $q_*(s,a) = \max_\pi q_\pi(s,a)$

How to find the associated optimal policy π^* ? By $a = argmax_{a \in \mathcal{A}} q^*(s,a)$. But we have to know $q^*(s,a)$...

$\Rightarrow$ Obtain it recursively by Bellman Equations

Definition 1.4 (Bellman's Optimality Equations)**.**

$$v^*(s) = \max_a \mathcal{R}_s^a + \sum_{s'\ inS} \mathcal{P}_{ss'}^a v^*(s')$$
$$q^*(s,a) = \mathcal{R}_s^a + \sum_{s' \in S} \mathcal{P}_{ss'}^a \max_{a'} q^*(s',a')$$

Iterative solution methods: Value Iteration, Policy Iteration, Q-learning, SARSA

2 Lecture 3: Dynamic Programming Algorithms

Definition 2.1 (Policy Evaluation)**.** *Gets v_π from iterative application of Bellman's equation. We iterate over $v_1 \to v_2 \to ... \to v_k$. This converges to v_π. We use at each step:*

$$v_{k+1} = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_k$$

Definition 2.2 (Policy Iteration)**.** *Once we have v_π, we choose a π' which greedily improves $v\pi$. We thus generate $\pi' \geq \pi$ with:*

$$\forall s', \pi'(s) = \arg\max_{a \in \mathcal{A}} q_\pi(s,a)$$

In Policy Iteration, we alternate between getting v_π through Policy Evaluation, given a policy π, then finding a better policy π' given those state-values. We then compute $v_{\pi'}$, and iterate... until we get v^*, π^*.

Definition 2.3 (Value Iteration)**.** *Same as Policy Evaluation, but we stop at the first step ($k = 1$), and we use at each step the Bellman's Optimality Equation to get $v_{k+1}(s)$ from the $v_k(s)$.*
We get $v_1 \to v_2 \to ... \to v_k$, which converges to v^*. We use at each step:

$$v_{k+1} = \max_{a \in \mathcal{A}} \mathcal{R}^a + \gamma \mathcal{P}^a v_k$$

We also define Asynchronous Dynamic Programming, which will update inplace the state-values, using most up-to-date values for other states:

$$v_{new}(s) \leftarrow \max_{a \in ac} (\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{new}(s'))$$

3 Lecture 4-8: Utility Theory and Applications

3.1 Utility Theory

Utility functions are concave, and their concavity represent the Risk-Aversion of agents.

Definition 3.1 (Certainty Eqauivalent Value)**.**

$$x_{CE} = U^{-1}(\mathbb{E}[U(x)])$$

Absolute Risk Premium:

$$\pi_A = \mathbb{E}[x] - x_{CE}$$

Relative Risk Premium:

$$\pi_A = \frac{\mathbb{E}[x] - x_{CE}}{\mathbb{E}[x]}$$

Definition 3.2 (CARA)**.** *Constant Absolute Risk Aversion*
Utility: $U(x) = \frac{e^{-ax}}{a}$
Absolute Risk Aversion: $A(x) = -\frac{U''(x)}{U(x)} = a$

With CARA and $x \sim \mathcal{N}(\mu, \sigma^2)$:

$$x_{CE} = \mu - \frac{a\sigma^2}{2}$$

Definition 3.3 (CRRA)**.** *Constant Relative Risk Aversion*
Utility: $U(x) = \frac{x^{1-\gamma}}{1-\gamma}$
Relative Risk Aversion: $A(x) = -\frac{U''(x)x}{U(x)} = \gamma$

With CRRA and $x \sim \mathcal{N}(\mu, \sigma^2)$:

$$x_{CE} = \frac{\mu - r}{a\sigma^2}$$