Dynamic Programming

V. Leclère (ENPC)

March 1, 2024

Convention in these slides

Just a quick point about some unusual conventions I am using :

- ♡ means that the results in the slides are really important
- ♢ means that the content is more advanced
- ♣ is a very simple exercise (can be done in class)
- ♠ is a somewhat more difficult exercise that you can use as a training
- [BV x.y] means that the content is covered in the Convex Optimization book in chapter x, section y.

Why should I bother to learn this stuff?

- Markov Chains and Markov Decision Programms are very powerful modeling tools for a lot of practical applications.
- Dynamic programming is a flexible tool, easy to implement, that can efficiently address these problems.
- $\Longrightarrow$ useful for any "manager"

Contents

1. Controlled Markov Chain

2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation

3. Infinite horizon

4. Parting thoughts

5. Wrap-up

Introduction ♡

- A Markov Chain $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a *memoryless* stochastic process.
- A classical example is the random walk : let $(\boldsymbol{\xi_t})_{t\in\mathbb{N}}$ be a sequence of i.i.d. centered random variables and define

$$\boldsymbol{X_0} = 0, \qquad \boldsymbol{X_{t+1}} = \boldsymbol{X_t} + \boldsymbol{\xi_{t+1}}.$$

- A Markov chain can represent a large number of systems affected by random noises.
- A controlled Controlled Markov Chain is a Markov Chain such that the evolution is affected by an action.

Markov chain: definition

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Let $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ be a sequence of discrete random variables taking value in $\mathcal{X}$. Let $\mathcal{F}_t = \sigma(\boldsymbol{X_0}, \dots, \boldsymbol{X_t})$ be the σ-algebra generated by all $\boldsymbol{X_\tau}$ for $\tau \leq t$.

We say that $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a Markov Chain if

$$\mathbb{P}(\boldsymbol{X_t} \in A \mid \mathcal{F}_s) = \mathbb{P}(\boldsymbol{X_t} \in A \mid \boldsymbol{X_s}), \qquad \forall s \leq t, \forall A \text{ measurable}$$

or equivalently

$$\mathbb{E}[f(\boldsymbol{X_t}) \mid \mathcal{F}_s] = \mathbb{E}[f(\boldsymbol{X_t}) \mid \boldsymbol{X_s}], \qquad \forall s \leq t, \quad \forall f \quad \text{bounded and measurable}$$

If all $\boldsymbol{X_t}$ are discrete, this reads

$$\mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_0} = x_0, \dots, \boldsymbol{X_s} = x_s) = \mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_s} = x_s) \qquad \forall s \leq t, \forall x_0, \dots, x_t$$

Markov chain: definition ♢

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Let $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ be a sequence of discrete random variables taking value in $\mathcal{X}$. Let $\mathcal{F}_t = \sigma(\boldsymbol{X_0}, \dots, \boldsymbol{X_t})$ be the σ-algebra generated by all $\boldsymbol{X_\tau}$ for $\tau \leq t$.

We say that $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a Markov Chain if

$$\mathbb{P}(\boldsymbol{X_t} \in A \mid \mathcal{F}_s) = \mathbb{P}(\boldsymbol{X_t} \in A \mid \boldsymbol{X_s}), \qquad \forall s \leq t, \forall A \text{ measurable}$$

or equivalently

$$\mathbb{E}[f(\boldsymbol{X_t}) \mid \mathcal{F}_s] = \mathbb{E}[f(\boldsymbol{X_t}) \mid \boldsymbol{X_s}], \qquad \forall s \leq t, \quad \forall f \quad \text{bounded and measurable}$$

If all $\boldsymbol{X_t}$ are discrete, this reads

$$\mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_0} = x_0, \dots, \boldsymbol{X_s} = x_s) = \mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_s} = x_s)$$
$$\forall s \leq t, \forall x_0, \dots, x_t$$

Markov chain: definition ◊

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Let $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ be a sequence of discrete random variables taking value in $\mathcal{X}$. Let $\mathcal{F}_t = \sigma(\boldsymbol{X_0}, \dots, \boldsymbol{X_t})$ be the σ-algebra generated by all $\boldsymbol{X_\tau}$ for $\tau \leq t$.

We say that $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a Markov Chain if

$$\mathbb{P}(\boldsymbol{X_t} \in A \mid \mathcal{F}_s) = \mathbb{P}(\boldsymbol{X_t} \in A \mid \boldsymbol{X_s}), \qquad \forall s \leq t, \forall A \text{ measurable}$$

or equivalently

$$\mathbb{E}[f(\boldsymbol{X_t}) \mid \mathcal{F}_s] = \mathbb{E}[f(\boldsymbol{X_t}) \mid \boldsymbol{X_s}], \qquad \forall s \leq t, \quad \forall f \quad \text{bounded and measurable}$$

If all $\boldsymbol{X_t}$ are discrete, this reads

$$\mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_0} = x_0, \dots, \boldsymbol{X_s} = x_s) = \mathbb{P}(\boldsymbol{X_t} = x_t \mid \boldsymbol{X_s} = x_s)$$
$$\forall s \leq t, \forall x_0, \dots, x_t$$

Exercises

♣ Exercise: Show that if $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a sequence of independent random variables then it is a Markov Chain.

♠ Exercise: Let $(\boldsymbol{\xi_t})_{t\in\mathbb{N}}$ be i.i.d. Assume that, for all $t \in \mathbb{N}$,

$$\boldsymbol{X_{t+k}} = \sum_{\kappa=0}^{k-1} \alpha_\kappa \boldsymbol{X_{t+\kappa}} + \boldsymbol{\xi_t}.$$

Show that $\boldsymbol{X_t}$ can easily be deduced from a Markov chain.

Discrete Markov chains ♡

Let $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ be a Markov chain s.t. $\mathrm{supp}(\boldsymbol{X_t}) \subset \mathcal{X}$ where $\mathcal{X}$ is finite[1].

- We call $P_t : \mathcal{X}^2 \to [0,1]$ the matrix such that,

  $$P_t(x,y) = \mathbb{P}(\boldsymbol{X_{t+1}} = y | \boldsymbol{X_t} = x)$$

  the t-transition kernel of the Markov Chain $(\boldsymbol{X_t})_{t\in\mathbb{N}}$.
- A time-homogeneous Markov chain is such that $P_t = P$ for all t.

[1] extension to countable case are straightforward.

Time homogeneous Markov chain graph representation

A simple way to represent a discrete Markov chain is through a directed graph:

- each node represents a state,
- we add an arc between node x and y iff $P(x, y) > 0$,
- when positive, we add the value $P(x, y)$ on the arc between x and y.

A time homogenous Markov chain is irreduccible if, starting from any point you can eventually reach any other points. More precisely, if for all $x, y \in \mathcal{X}$ there exists $t \in \mathbb{N}$ such that $\mathbb{P}(\boldsymbol{X_t} = y | \boldsymbol{X_0} = x) > 0$. Or equivalently if its graph is strongly connected.

Time homogeneous Markov chain graph representation

A simple way to represent a discrete Markov chain is through a directed graph:

- each node represents a state,
- we add an arc between node x and y iff $P(x,y) > 0$,
- when positive, we add the value $P(x,y)$ on the arc between x and y.

A time homogenous Markov chain is irreduccible if, starting from any point you can eventually reach any other points. More precisely, if for all $x, y \in \mathcal{X}$ there exists $t \in \mathbb{N}$ such that $\mathbb{P}(\boldsymbol{X_t} = y | \boldsymbol{X_0} = x) > 0$. Or equivalently if its graph is strongly connected.

Absorbing state

- An absorbing state of a Markov chain, is a state x such that there is no positive transition from x to another state $y \neq x$, that is such that $P(x, x) = 1$.
- If, from any state x there is a path to an absorbing state, then the Markov chain will almost surely end in an absorbing state.

Controlled Markov chains ♡

A controlled Markov chain is a Markov Chain whose transition kernel at time t is decided by an action $a_t \in \mathcal{A}$:

$$\mathbb{P}(\boldsymbol{X_{t+1}} = y | \boldsymbol{X_t} = x) = P_t^{a_t}(x, y).$$

- We consider a set of actions (or control) $\mathcal{A}$, assumed finite for simplicity.
- For all $t \in \mathbb{N}$ and $a \in \mathcal{A}$, let P_t^a be a transition kernel.
- We call a function π mapping the states $\mathcal{X}$ in to the action $\mathcal{A}$ a policy, and a collection $\pi = (\pi_t)_{t\in\mathbb{N}}$ a strategy.
- For any strategy π we define $(\boldsymbol{X_t^\pi})_{t\in\mathbb{N}}$ such that $(\boldsymbol{X_t}, \boldsymbol{a_t})_{t\in\mathbb{N}}$ is a Markov chain with

$$\mathbb{P}(\boldsymbol{X_{t+1}^\pi} = y, \boldsymbol{a_{t+1}} = b | \boldsymbol{X_t^\pi} = x, \boldsymbol{a_t} = a) = P_t^a(x, y)\mathbb{1}_{\pi_t(y)=b}.$$

Example and representation of Controlled Markov Chain

We consider a maintenance problem. A unit U can be either *working* or *broken*. When it is in a working state there is a 20% chance of being broken at the next time step. When it is broken it must be replaced and will be working at the next step.

♣ Exercise: Model this as a Markov Chain.

♣ Exercise: We now assume that at each time step, if the unit is working, we can decide to maintain it (keeping it in a working state) or not. And if broken we can repair it, or not. Model this modified version as a controlled Markov Chain.

Stochastic Dynamic System

- A (discrete time) stochastic dynamic system is a stochastic process $(\boldsymbol{X}_t)_{t\in\mathbb{N}}$ such that

$$\boldsymbol{X_{t+1}} = f_t(\boldsymbol{X_t}, \boldsymbol{a_t}, \boldsymbol{\xi_t}), \qquad \forall t$$

  where f_t is a deterministic function, $\boldsymbol{a_t}$ takes values in $\mathcal{A}$, and $\boldsymbol{\xi_t}$ is an exogenous random variable (i.e. its law is not affected by $\boldsymbol{X_t}$ and $\boldsymbol{a_t}$).
- All controlled Markov chains can be written as a stochastic dynamic system.
- If $(\boldsymbol{\xi_t})_{t\in\mathbb{N}}$ is an independent sequence of random variables, then $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ is a controlled Markov chain.

Contents

1 Controlled Markov Chain

2 Dynamic Programming
- Markov Decision Problem
- Dynamic Programming: Intuition
- Dynamic Programming: Value function
- Dynamic Programming: implementation

3 Infinite horizon

4 Parting thoughts

5 Wrap-up

Contents

1. Controlled Markov Chain
2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation
3. Infinite horizon
4. Parting thoughts
5. Wrap-up

Markov Decision Problem ♡

- Let $(\boldsymbol{X_t})_{t\in\mathbb{N}}$ be a controlled Markov chain, with action in $\mathcal{A}$. We denote Π the set of associated policies.
- Let, for all t, $c_t : \mathcal{X}^2 \to \mathbb{R} \cup +\infty$ be a transition cost.[2]
- A Markov Decision Problem is

$$\underset{\pi\in\Pi}{\text{Min}} \qquad \mathbb{E}\Big[\sum_{t\in\mathbb{N}} \rho^t c_t(\boldsymbol{X_t^\pi}, \boldsymbol{X_{t+1}^\pi})\Big],$$

where $\rho \in [0,1]$ is a discount factor.

[2] the transition cost can also be dependent on action a.

Another point of view

We can also write the MDP problem in the following way

$$\begin{aligned} \underset{(\pi_t)_{t\in\mathbb{N}}}{\text{Min}} \quad & \mathbb{E}\Big[\mathbb{E}\Big[\sum_{t=1}^{\infty} \rho^t c_t(\boldsymbol{X_t}, \boldsymbol{X_{t+1}}) \mid \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t})\Big]\Big] \\ \text{s.t.} \quad & \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t}) \qquad \forall t \end{aligned}$$

Equivalently, with a stochastic dynamic system point of view, we have

$$\begin{aligned} \underset{(\pi_t)_{t\in\mathbb{N}}}{\text{Min}} \quad & \mathbb{E}\Big[\sum_{t=1}^{\infty} \rho^t c_t(\boldsymbol{X_t}, \boldsymbol{X_{t+1}})\Big] \\ \text{s.t.} \quad & \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t}) \qquad \forall t \\ & \boldsymbol{X_{t+1}} = f_t(\boldsymbol{X_t}, \boldsymbol{a_t}, \boldsymbol{\xi_t}) \qquad \forall t \end{aligned}$$

Another point of view

We can also write the MDP problem in the following way

$$\begin{aligned} \underset{(\pi_t)_{t\in\mathbb{N}}}{\text{Min}} \quad & \mathbb{E}\Big[\mathbb{E}\Big[\sum_{t=1}^{\infty} \rho^t c_t(\boldsymbol{X_t}, \boldsymbol{X_{t+1}}) \mid \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t})\Big]\Big] \\ \text{s.t.} \quad & \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t}) & \forall t \end{aligned}$$

Equivalently, with a stochastic dynamic system point of view, we have

$$\begin{aligned} \underset{(\pi_t)_{t\in\mathbb{N}}}{\text{Min}} \quad & \mathbb{E}\Big[\sum_{t=1}^{\infty} \rho^t c_t(\boldsymbol{X_t}, \boldsymbol{X_{t+1}})\Big] \\ \text{s.t.} \quad & \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t}) & \forall t \\ & \boldsymbol{X_{t+1}} = f_t(\boldsymbol{X_t}, \boldsymbol{a_t}, \boldsymbol{\xi_t}) & \forall t \end{aligned}$$

Finite horizon problem

We now assume that for $t > T$, $c_t \equiv 0$, $\rho = 1$ and $c_T(x,y) = K(x)$. Thus the problem reads

$$\begin{aligned} \underset{(\pi_t)_{t\in\mathbb{N}}}{\text{Min}} \quad & \mathbb{E}\Big[\sum_{t=1}^{T-1} c_t(\boldsymbol{X_t}, \boldsymbol{X_{t+1}}) + K(\boldsymbol{X_T})\Big] & \\ \text{s.t.} \quad & \boldsymbol{a_t} = \pi_t(\boldsymbol{X_t}) & \forall t \\ & \boldsymbol{X_{t+1}} = f_t(\boldsymbol{X_t}, \boldsymbol{a_t}, \boldsymbol{\xi_t}) & \forall t \end{aligned}$$

Further, we often assume that the initial state $\boldsymbol{X_0} = x_0$ is known.

This will be known as the Finite Horizon Problem.

Contents

1. Controlled Markov Chain

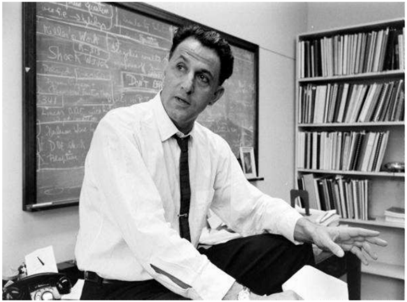
2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation

3. Infinite horizon

4. Parting thoughts

5. Wrap-up

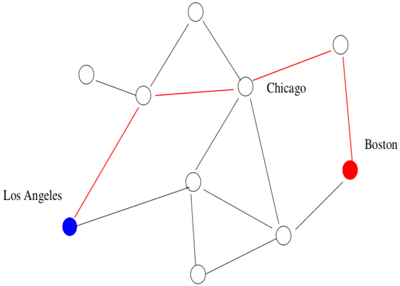
Bellman's Principle of Optimality



Richard Ernest Bellman (August 26, 1920 – March 19, 1984)

An optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Richard Bellman)

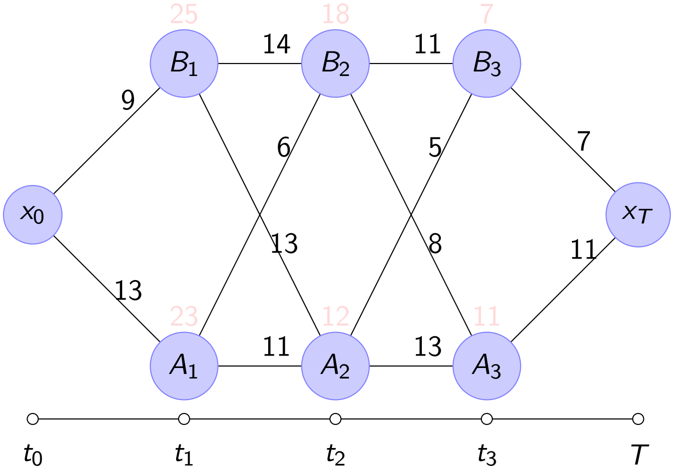
The shortest path on a graph illustrates Bellman's Principle of Optimality



For an auto travel analogy, suppose that the fastest route from Los Angeles to Boston passes through **Chicago**.
The principle of optimality translates to the obvious fact that the Chicago to Boston portion of the route is also the fastest route for a trip that starts from Chicago and ends in Boston. (Dimitri P. Bertsekas)

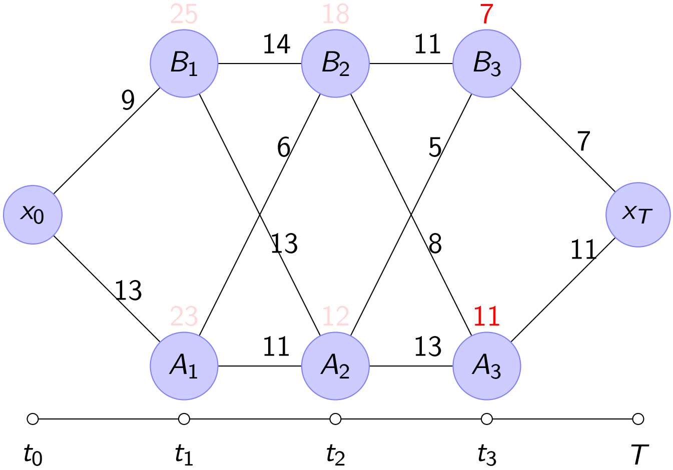
Idea Behind Dynamic Programming

Suppose that we have two states A and B and 4 timesteps.
For all t, we pay a cost to move from one node to one-another.



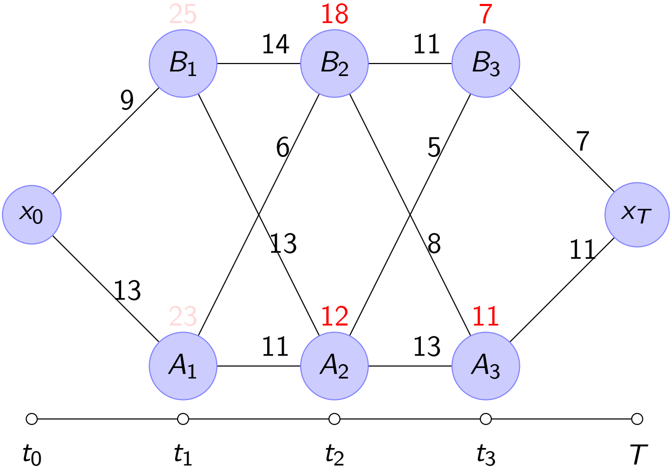
Idea Behind Dynamic Programming

We start from final position x_T, and computes cost to move from A_3 or B_3 to final position at time $t = 3$



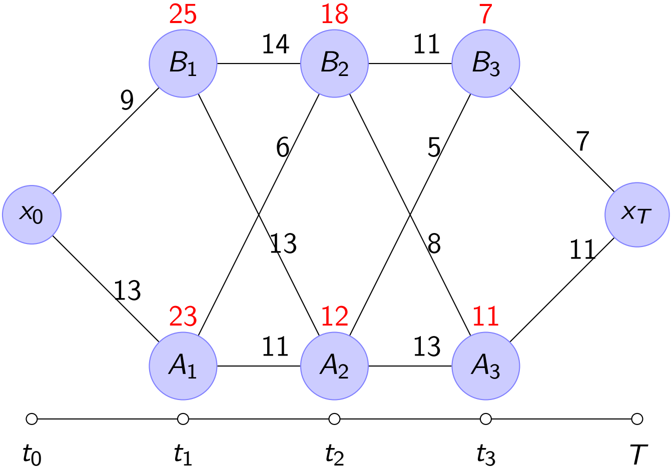
Idea Behind Dynamic Programming

We do the same at time $t = 2$, considering the cost to go to A_3 or B_3.



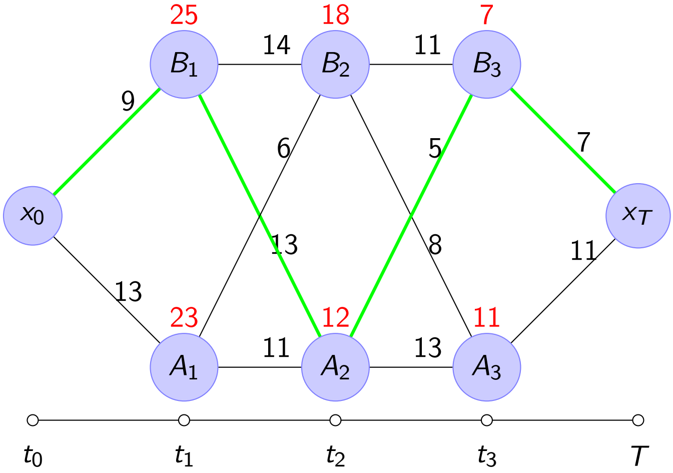
Idea Behind Dynamic Programming

We do the same at time $t = 1$, considering the cost to go to A_2 or B_2.



Idea Behind Dynamic Programming

Then, we can deduce easily the optimal trajectories from time $t = 0$ to T.



Idea behind dynamic programming

If we are in a Markovian setting, that is such that noises are time independent, then

1. The cost-to-go at time t depends only upon the current state.
2. We can compute recursively the cost to go for each position, starting from the terminal state and computing optimal trajectories backward.

Contents

1. Controlled Markov Chain

2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation

3. Infinite horizon

4. Parting thoughts

5. Wrap-up

Bellman's function: finite horizon ♡

In the finite horizon setting, the Bellman function reads

$$V_t(x) := \operatorname*{Min}_{\pi \in \Pi} \quad \mathbb{E}\Big[\sum_{\tau=t}^{T-1} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) + K(\boldsymbol{X}_T^\pi) \quad \Big| \quad \boldsymbol{X}_t = x \Big]$$

and in particular $V_T = K$.

Or, in the stochastic dynamic system point of view

$$\begin{aligned} V_t(x) = \quad & \operatorname*{Min}_{(\pi_\tau)_{\tau \in [\![t, T-1]\!]}} \quad \mathbb{E}\Big[\sum_{\tau=t}^{T-1} c_\tau(\boldsymbol{X}_\tau, \boldsymbol{X}_{t+1}) + K(\boldsymbol{X}_T) \quad \Big| \quad \boldsymbol{X}_t = x \Big] \\ & \text{s.t.} \qquad \boldsymbol{a}_\tau = \pi_\tau(\boldsymbol{X}_\tau) \\ & \qquad\qquad \boldsymbol{X}_{t+1} = f_\tau(\boldsymbol{X}_\tau, \boldsymbol{a}_\tau, \xi_\tau) \\ & \qquad\qquad \boldsymbol{X}_t = x \end{aligned}$$

Bellman's function: finite horizon ♡

In the finite horizon setting, the Bellman function reads

$$V_t(x) := \mathop{\mathrm{Min}}_{\pi \in \Pi} \quad \mathbb{E}\Big[\sum_{\tau=t}^{T-1} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) + K(\boldsymbol{X}_T^\pi) \quad \Big| \quad \boldsymbol{X}_t = x \Big]$$

and in particular $V_T = K$.
Or, in the stochastic dynamic system point of view

$$\begin{aligned} V_t(x) = \quad \mathop{\mathrm{Min}}_{(\pi_\tau)_{\tau \in [\![t, T-1]\!]}} \quad & \mathbb{E}\Big[\sum_{\tau=t}^{T-1} c_\tau(\boldsymbol{X}_\tau, \boldsymbol{X}_{t+1}) + K(\boldsymbol{X}_T) \quad \Big| \quad \boldsymbol{X}_t = x \Big] \\ \text{s.t.} \quad & \boldsymbol{a}_\tau = \pi_\tau(\boldsymbol{X}_\tau) \\ & \boldsymbol{X}_{t+1} = f_\tau(\boldsymbol{X}_\tau, \boldsymbol{a}_\tau, \boldsymbol{\xi}_\tau) \\ & \boldsymbol{X}_t = x \end{aligned}$$

Bellman's recursion : finite horizon ♡

In the finite horizon setting, we have

$$\begin{cases} V_T(x) & = K(x) \\ V_t(x) & = \min_{a \in \mathcal{A}} \mathbb{E}\Big[c_t(x, \boldsymbol{X_{t+1}}) + V_{t+1}(\boldsymbol{X_{t+1}}) \;\Big|\; \boldsymbol{X_t} = x,\ \boldsymbol{a_t} = a\Big] \\ & = \min_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x,y)\Big(c_t(x,y) + V_{t+1}(y)\Big) \end{cases}$$

An optimal policy is given by

$$\pi_t(x) \in \arg\min_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x,y)\Big(c_t(x,y) + V_{t+1}(y)\Big).$$

Bellman's recursion : finite horizon ♡

In the finite horizon setting, we have

$$\begin{cases} V_T(x) & = K(x) \\ V_t(x) & = \min_{a \in \mathcal{A}} \mathbb{E}\Big[c_t(x, \boldsymbol{X_{t+1}}) + V_{t+1}(\boldsymbol{X_{t+1}}) \;\Big|\; \boldsymbol{X_t} = x,\ \boldsymbol{a_t} = a\Big] \\ & = \min_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x,y)\Big(c_t(x,y) + V_{t+1}(y)\Big) \end{cases}$$

An optimal policy is given by

$$\pi_t(x) \in \arg\min_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x,y)\Big(c_t(x,y) + V_{t+1}(y)\Big).$$

Contents

1 Controlled Markov Chain

2 Dynamic Programming
- Markov Decision Problem
- Dynamic Programming: Intuition
- Dynamic Programming: Value function
- Dynamic Programming: implementation

3 Infinite horizon

4 Parting thoughts

5 Wrap-up

Dynamic Programming Algorithm - Discrete Case

```
Data: Problem parameters
Result: optimal trajectory and value;
$V_T \equiv K$ ; for $t : T-1 \to 0$ do
    for $x \in \mathcal{X}$ do
        $V_t(x) = \min_{a \in \mathcal{A}} \mathbb{E}\Big[c_t(x, \boldsymbol{X_{t+1}}) + V_{t+1}(\boldsymbol{X_{t+1}}) \;\Big|\; \boldsymbol{X_t} = x,\ \boldsymbol{a}_t = a\Big]$
```

Algorithm 1: Classical stochastic dynamic programming algorithm

Dynamic Programming Algorithm - Discrete Case

```
**Data:** Problem parameters
**Result:** optimal trajectory and value;
$V_T \equiv K$ ; **for** $t : T-1 \to 0$ **do**
    **for** $x \in \mathcal{X}$ **do**
        $V_t(x) = \infty$
        **for** $a \in \mathcal{A}$ **do**
            $Q(x,a) = \mathbb{E}\Big[c_t(x, \boldsymbol{X_{t+1}}) + V_{t+1}(\boldsymbol{X_{t+1}}) \Big| \boldsymbol{X_t} = x,\ \boldsymbol{a}_t = a\Big]$
            **if** $Q(x,a) < V_t(x)$ **then**
                $V_t(x) = Q(x,a)$
                $\pi_t(x) = a$
```

Algorithm 2: Classical stochastic dynamic programming algorithm

Dynamic Programming Algorithm - Discrete Case

```
Data: Problem parameters
Result: optimal trajectory and value;
$V_T \equiv K$ ; for $t : T-1 \to 0$ do
    for $x \in \mathcal{X}$ do
        $V_t(x) = \infty$
        for $a \in \mathcal{A}$ do
            $Q(x,a) = 0$
            for $y \in \mathcal{X}$ do
                $Q(x,a) = Q(x,a) + P^a(x,y)[c_t(x,y) + V_{t+1}(y)]$
            if $Q(x,a) < V_t(x)$ then
                $V_t(x) = Q(x,a)$
                $\pi_t(x) = a$
```

Algorithm 3: Classical stochastic dynamic programming algorithm

3 curses of dimensionality ♡

Complexity $= O(T \times |\mathcal{X}|^2 \times |\mathcal{A}|)$
Linear in the number of time steps, but we have 3 curses of dimensionality:

1. State. Complexity is exponential in the dimension of $\mathcal{X}$
   e.g. 3 independent states each taking 10 values lead to a loop over 10^3 points.
2. Decision. Complexity is exponential in the dimension of $\mathcal{X}_t$.
   $\rightsquigarrow$ due to exhaustive minimization of the inner problem. Can be accelerated using a faster method (e.g. MILP solver).
3. Expectation. Complexity is exponential in the dimension of Ξ_t.
   $\rightsquigarrow$ due to expectation computation. Can be accelerated through Monte-Carlo approximation (still at least 1000 points)

In practice, DP is not used for state of dimension more than 5.

3 curses of dimensionality ♡

Complexity $= O(T \times |\mathcal{X}|^2 \times |\mathcal{A}|)$
Linear in the number of time steps, but we have 3 curses of dimensionality:

1. State. Complexity is exponential in the dimension of $\mathcal{X}$
   e.g. 3 independent states each taking 10 values lead to a loop over 10^3 points.
2. Decision. Complexity is exponential in the dimension of $\mathcal{X}_t$.
   $\rightsquigarrow$ due to exhaustive minimization of the inner problem. Can be accelerated using a faster method (e.g. MILP solver).
3. Expectation. Complexity is exponential in the dimension of Ξ_t.
   $\rightsquigarrow$ due to expectation computation. Can be accelerated through Monte-Carlo approximation (still at least 1000 points)

In practice, DP is not used for state of dimension more than 5.

Some remarks

◊

- The loop on the next state y does not need to be on all state, but only on all *reachable* next state from state x.
- In some cases you do not need to compute the $V_t(x)$ for all $x \in \mathcal{X}$, indeed you might be able to show that some parts of the state space $\mathcal{X}$ are not reachable (or not reachable under an optimal policy) at time t.
- To represent that, at some time t, some state $x \in \mathcal{X}$ are forbidden, you can simply encode $V_t(x) = +\infty$.
- To represent that, at some time t, the transition $x \to y$ is forbidden, you can simply encode $c_t(x, y) = +\infty$.

Some remarks

- The loop on the next state y does not need to be on all state, but only on all *reachable* next state from state x.
- In some cases you do not need to compute the $V_t(x)$ for all $x \in \mathcal{X}$, indeed you might be able to show that some parts of the state space $\mathcal{X}$ are not reachable (or not reachable under an optimal policy) at time t.
- To represent that, at some time t, some state $x \in \mathcal{X}$ are forbidden, you can simply encode $V_t(x) = +\infty$.
- To represent that, at some time t, the transition $x \to y$ is forbidden, you can simply encode $c_t(x, y) = +\infty$.

Some remarks

- The loop on the next state y does not need to be on all state, but only on all *reachable* next state from state x.
- In some cases you do not need to compute the $V_t(x)$ for all $x \in \mathcal{X}$, indeed you might be able to show that some parts of the state space $\mathcal{X}$ are not reachable (or not reachable under an optimal policy) at time t.
- To represent that, at some time t, some state $x \in \mathcal{X}$ are forbidden, you can simply encode $V_t(x) = +\infty$.
- To represent that, at some time t, the transition $x \to y$ is forbidden, you can simply encode $c_t(x, y) = +\infty$.

Exercise

- Let $\mathcal{X} = \{0, 1, 2, 3\}$, $\mathcal{A} = \{0, 1\}$.
- Let $(\boldsymbol{X_t})_{t \in [\![1,5]\!]}$ be a controlled Markov chain, such that, if $a = 0$, it stays in its state, and if $a = 1$ it has a probability 0.5 of going 1 up (if possible, otherwise stay in place), and 0.5 of going 1 down (if possible, otherwise stay in place).

Solve by Dynamic Programming the following optimization problem.

$$\text{Max} \qquad \mathbb{E}\Big[\sum_{t=0}^{4} \boldsymbol{X_t}^2 \mid \boldsymbol{X_0} = 0\Big]$$

Contents

1 Controlled Markov Chain

2 Dynamic Programming
- Markov Decision Problem
- Dynamic Programming: Intuition
- Dynamic Programming: Value function
- Dynamic Programming: implementation

3 Infinite horizon

4 Parting thoughts

5 Wrap-up

Bellman's value function

The Bellman's value function, a.k.a cost-to-go function, is defined as (when the expectation make sense)

$$V_t(x) := \operatorname*{Min}_{\pi \in \Pi} \qquad \mathbb{E}\Big[\sum_{\tau=t}^{+\infty} \rho^{\tau-t} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \quad | \quad \boldsymbol{X}_t = x\Big]$$

It is the value of the problem starting from time t in state x.

The expectation is well-defined for example if we consider a finite controlled Markov chain and one of the following holds:

- we are in the finite horizon framework,
- $c_t = c$ and $\rho < 1$,
- $c_t = c$ and there is a cemetery state (that is an aborbing state with null transition cost) that is almost surely reached.

Bellman's value function

The Bellman's value function, a.k.a cost-to-go function, is defined as (when the expectation make sense)

$$V_t(x) := \underset{\pi \in \Pi}{\mathrm{Min}} \qquad \mathbb{E}\Big[\sum_{\tau=t}^{+\infty} \rho^{\tau-t} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \quad | \quad \boldsymbol{X}_t = x\Big]$$

It is the value of the problem starting from time t in state x.

The expectation is well-defined for example if we consider a finite controlled Markov chain and one of the following holds:

- we are in the finite horizon framework,
- $c_t = c$ and $\rho < 1$,
- $c_t = c$ and there is a cemetery state (that is an aborbing state with null transition cost) that is almost surely reached.

Bellman's recursion

$$
\begin{aligned}
V_t(x) &= \underset{\pi\in\Pi}{\mathrm{Min}}\ \mathbb{E}\Big[\sum_{\tau=t}^{+\infty}\rho^{\tau-t}c_\tau(\boldsymbol{X}_\tau^\pi,\boldsymbol{X}_{\tau+1}^\pi)\ \Big|\ \boldsymbol{X}_t^\pi = x\Big] \\
&= \underset{\pi\in\Pi}{\mathrm{Min}}\ \mathbb{E}\Big[c_\tau(\boldsymbol{X}_t^\pi,\boldsymbol{X}_{t+1}^\pi)\ +\sum_{\tau=t+1}^{+\infty}\rho^{\tau-t}c_\tau(\boldsymbol{X}_\tau^\pi,\boldsymbol{X}_{\tau+1}^\pi)\Big|\ \boldsymbol{X}_t^\pi = x\Big] \\
&= \underset{a\in\mathcal{A}}{\mathrm{Min}}\sum_{y\in\mathcal{X}}P^a(x,y)\Big(c_t(x,y)+\underset{\pi\in\Pi}{\mathrm{Min}}\,\mathbb{E}\Big[\sum_{\tau=t+1}^{+\infty}\rho^{\tau-t}c_\tau(\boldsymbol{X}_\tau^\pi,\boldsymbol{X}_{\tau+1}^\pi)\Big|\ \boldsymbol{X}_{t+1}^\pi = y\Big]\Big) \\
&= \underset{a\in\mathcal{A}}{\mathrm{Min}}\sum_{y\in\mathcal{X}}P^a(x,y)(c_t(x,y)+\rho V_{t+1}(y)) \\
&= \underset{a\in\mathcal{A}}{\mathrm{Min}}\,\mathbb{E}\Big[c_t(\boldsymbol{X}_t,\boldsymbol{X}_{t+1})+\rho V_{t+1}(\boldsymbol{X}_{t+1})\ \Big|\ \boldsymbol{X}_t = x, \boldsymbol{a}_t = a\Big]
\end{aligned}
$$

This equation should be understood as *the cost-to-go from state x and time t is equal to the minimum expected current cost plus futur cost.*

Bellman's recursion

$$
\begin{aligned}
V_t(x) &= \operatorname*{Min}_{\pi \in \Pi} \ \mathbb{E}\Big[\sum_{\tau=t}^{+\infty} \rho^{\tau-t} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \ \Big| \ \boldsymbol{X}_t^\pi = x \Big] \\
&= \operatorname*{Min}_{\pi \in \Pi} \ \mathbb{E}\Big[c_\tau(\boldsymbol{X}_t^\pi, \boldsymbol{X}_{t+1}^\pi) \ + \sum_{\tau=t+1}^{+\infty} \rho^{\tau-t} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \Big| \ \boldsymbol{X}_t^\pi = x \Big] \\
&= \operatorname*{Min}_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x, y) \Big(c_t(x, y) + \operatorname*{Min}_{\pi \in \Pi} \mathbb{E}\Big[\sum_{\tau=t+1}^{+\infty} \rho^{\tau-t} c_\tau(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \Big| \ \boldsymbol{X}_{t+1}^\pi = y \Big] \Big) \\
&= \operatorname*{Min}_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x, y)(c_t(x, y) + \rho V_{t+1}(y)) \\
&= \operatorname*{Min}_{a \in \mathcal{A}} \mathbb{E}\Big[c_t(\boldsymbol{X}_t, \boldsymbol{X}_{t+1}) + \rho V_{t+1}(\boldsymbol{X}_{t+1}) \ \Big| \ \boldsymbol{X}_t = x, \boldsymbol{a}_t = a \Big]
\end{aligned}
$$

This equation should be understood as *the cost-to-go from state x and time t is equal to the minimum expected current cost plus futur cost*.

Stationary problem

From now on we make the following assumption:

- the set of possible values $\mathcal{X}$ is finite,
- the transition cost is not time dependent, i.e., $c_t = c$,
- the transition kernel is not time-dependent, i.e. $P_t^a = P^a$.

Then the MDP problem is said to be stationary.

A strategy $s = (\pi_t)_{t\in\mathbb{N}}$ is said to be stationary iff it is not time dependent, i.e. $\pi_t = \pi$.

A stationary MDP admits an optimal stationary policy.

Stationary problem

From now on we make the following assumption:

- the set of possible values $\mathcal{X}$ is finite,
- the transition cost is not time dependent, i.e., $c_t = c$,
- the transition kernel is not time-dependent, i.e. $P_t^a = P^a$.

Then the MDP problem is said to be stationary.

A strategy $s = (\pi_t)_{t\in\mathbb{N}}$ is said to be stationary iff it is not time dependent, i.e. $\pi_t = \pi$.

A stationary MDP admits an optimal stationary policy.

Stationary problem

From now on we make the following assumption:

- the set of possible values $\mathcal{X}$ is finite,
- the transition cost is not time dependent, i.e., $c_t = c$,
- the transition kernel is not time-dependent, i.e. $P_t^a = P^a$.

Then the MDP problem is said to be stationary.

A strategy $s = (\pi_t)_{t\in\mathbb{N}}$ is said to be stationary iff it is not time dependent, i.e. $\pi_t = \pi$.

A stationary MDP admits an optimal stationary policy.

Stochastic Shortest Path problem ◇

We consider a stationary MDP, with a cemetery state.

Stopping assumption

We assume that, for every state x, there exists T such that, under any (stationary) strategy π there is a positive probability of reaching the cemetery state.

♣ Exercise: Show that the finite horizon problem satisfies this stopping assumption.

♠ Exercise: Show that, if $\rho < 1$, even without an absorbing state, we can construct an equivalent MDP satisfying the stopping assumption.

Dynamic Programming equation ♡

Under this stopping assumption, the value function

$$V^\sharp(x) := \underset{\pi\in\Pi}{\mathrm{Min}} \quad \mathbb{E}\Big[\sum_{\tau=0}^{+\infty} \rho^\tau c(\boldsymbol{X}_\tau^\pi, \boldsymbol{X}_{\tau+1}^\pi) \quad \Big]$$

Is the only function V satisfying the Dynamic Programming equation

$$\begin{cases} V(x) & = \min_{a\in\mathcal{A}} \mathbb{E}\Big[c(x, \boldsymbol{X}_{t+1}) + \rho V(\boldsymbol{X}_{t+1}) \Bigm| \boldsymbol{X}_t = x,\ \boldsymbol{a}_t = a\Big] \\ & = \min_{a\in\mathcal{A}} \sum_{y\in\mathcal{X}} P^a(x,y)\Big(c(x,y) + \rho V(y)\Big) \end{cases}$$

Value iteration ♡

Define the following sequence of functions through the so-called Value Iteration procedure

$$\begin{cases} V_0 : x \mapsto 0 \\ V_{t+1} : x \mapsto \min_{a \in \mathcal{A}} \sum_{y \in \mathcal{X}} P^a(x, y)\Big(c_t(x, y) + \rho V_t(y)\Big) \end{cases}$$

Then we have, under the stopping assumption, $V_t \to V^\sharp$.

♠ Exercise: Recognize the Dynamic Programming algorithm of the finite horizon case. Interpret this result in terms of finite horizon approximation.

Optimal policy

Naturally, a stationary policy π is optimal iff the minimum is attained in the DP equation, i.e.

$$V^\sharp(x) = \sum_{y \in \mathcal{X}} P^{\pi(x)}(x, y)\Big(c_t(x, y) + \rho V^\sharp(y)\Big), \qquad \forall x \in \mathcal{X}$$

Contents

1. Controlled Markov Chain
2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation
3. Infinite horizon
4. Parting thoughts
5. Wrap-up

Law of large number and Central Limit Theorem ♡

Let $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ be a sequence of independent and identically distributed, real valued random variables. We denote the empirical mean $M_N = \frac{1}{N}\sum_{i=1}^N X_i$.

Theorem (LLN)

If $\boldsymbol{X_1}$ admits first order moment, then the empirical mean $\boldsymbol{M_N}$ converge almost surely toward the expectation $\mathbb{E}[\boldsymbol{X_1}]$.

Law of large number and Central Limit Theorem ♡

Let $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ be a sequence of independent and identically distributed, real valued random variables. We denote the empirical mean $M_N = \frac{1}{N}\sum_{i=1}^N X_i$.

Theorem (LLN)

If $\boldsymbol{X_1}$ admits first order moment, then the empirical mean $\boldsymbol{M_N}$ converge almost surely toward the expectation $\mathbb{E}[\boldsymbol{X_1}]$.

Theorem (CLT)

If $\boldsymbol{X_1}$ admits second order moment, then we have

$$\sqrt{n}\Big(\boldsymbol{M_N} - \mathbb{E}\big[\boldsymbol{X}\big]\Big) \to \mathcal{N}(0,\sigma)$$

where the convergence is in law and σ is the standard deviation of $\boldsymbol{X_1}$.

Law of large number and Central Limit Theorem ♡

Let $\{\boldsymbol{X}_i\}_{i\in\mathbb{N}}$ be a sequence of independent and identically distributed, real valued random variables. We denote the empirical mean $M_N = \frac{1}{N}\sum_{i=1}^N X_i$.

Theorem (LLN)

If $\boldsymbol{X}_1$ admits first order moment, then the empirical mean $\boldsymbol{M}_N$ converge almost surely toward the expectation $\mathbb{E}[\boldsymbol{X}_1]$.

Theorem (CLT)

If $\boldsymbol{X}_1$ admits second order moment, then we have

$$\sqrt{n}\Big(\boldsymbol{M}_N - \mathbb{E}\big[\boldsymbol{X}\big]\Big) \to \mathcal{N}(0,\sigma)$$

where the convergence is in law and σ is the standard deviation of $\boldsymbol{X}_1$.

In particular, the CLT means that, for $\boldsymbol{G} \sim \mathcal{N}(0,\sigma)$ and any $[a,b]$,

$$\mathbb{P}\Big(\sqrt{n}(\boldsymbol{M}_N - \mathbb{E}\big[\boldsymbol{X}\big]) \in [a,b]\Big) \to_N \mathbb{P}\Big(\boldsymbol{G} \in [a,b]\Big).$$

Monte-Carlo method

♡

- Let $\{\boldsymbol{X_i}\}_{i\in\mathbb{N}}$ be a sequence of rv iid with finite variance.
- We have $\mathbb{P}\Big(\boldsymbol{M_N} \in \Big[\mathbb{E}\big[\boldsymbol{X}\big] \pm \frac{\Phi^{-1}(1-p/2)std(\boldsymbol{X})}{\sqrt{N}}\Big]\Big) \approx p$
- In order to estimate the expectation $\mathbb{E}\big[\boldsymbol{X}\big]$, we can
  - sample N independent realizations of $\boldsymbol{X}$, $\{X_i\}_{i\in[\![1,N]\!]}$
  - compute the empirical mean $M_N = \frac{\sum_{i=1}^N X_i}{N}$, and standard-deviation s_N
  - choose an error level p (e.g. 5%) and compute $\Phi^{-1}(1-p/2)$ (1.96)
  - and we know that, asymptotically, the expectation $\mathbb{E}\big[\boldsymbol{X}\big]$ is in $\Big[M_N \pm \frac{\Phi^{-1}(p)s_N}{\sqrt{N}}\Big]$ with probability (on the sample) $1-p$

Good practice in optimization under uncertainty ♡

- Optimization under uncertainty is hard.
- You should first decide on a simulator for your problem, as precise as possible.
- Then, you should decide which problem you are going to solve. Most of the time it will be an approximation of the true problem.
- You can now solve, exactly or approximately this problem. Once you have a solution you should simulate it on your simulator (expected cost can be estimated by Monte Carlo).
- It is good practice to come up with reasonable *heuristic* to test your solution.

Good practice in optimization under uncertainty ♡

- Optimization under uncertainty is hard.
- You should first decide on a simulator for your problem, as precise as possible.
- Then, you should decide which problem you are going to solve. Most of the time it will be an approximation of the true problem.
- You can now solve, exactly or approximately this problem. Once you have a solution you should simulate it on your simulator (expected cost can be estimated by Monte Carlo).
- It is good practice to come up with reasonable *heuristic* to test your solution.

Contents

1. Controlled Markov Chain
2. Dynamic Programming
   - Markov Decision Problem
   - Dynamic Programming: Intuition
   - Dynamic Programming: Value function
   - Dynamic Programming: implementation
3. Infinite horizon
4. Parting thoughts
5. Wrap-up

What you have to know

- What is a Markov Chain, a Markov Controlled Chain.
- What is a Markov Decision Problem, a state, a policy
- What is the Bellman's value function a.k.a cost-to-go
- Estimate the value of a policy through Monte Carlo

What you really should know

- the complexity of Dynamic Programming
- how to model forbidden state in DP
- How to guarantee that an MDP in infinite horizon admits an optimal stationary policy

What you have to be able to do

- Recognize an MDP
- Write a Dynamic Programming equation
- Solve a simple, finite horizon, MDP problem through Dynamic Programming

What you should be able to do

- Know if a problem can numerically be tackled through Dynamic Programming
- Reframe a non-Markovian problem as a Markovian problem through extending the state
- Implement a value iteration algorithm in infinite horizon setting