Introduction to Markov Decision Processes

INDE 597

March 6, 2019

Saumya Sinha

Stochastic Processes

Stochastic Processes

- Random variable X – function with many possible outcomes, each with some probability.

Stochastic Processes

- Random variable X – function with many possible outcomes, each with some probability.
- Examples:
  - $X =$ outcome of a (fair) coin toss: $\{H, T\}$.
  - $Y =$ cars sold in Houston till date in 2019: $\{0, 1, \ldots, M_1\}$.
  - $Z =$ length of queue at the Coffeehouse at noon: $\{0, 1, \ldots, M_2\}$.

Stochastic Processes

- Random variable X – function with many possible outcomes, each with some probability.
- Examples:
  - $X =$ outcome of a (fair) coin toss: $\{H, T\}$.
  - $Y =$ cars sold in Houston till date in 2019: $\{0, 1, \ldots, M_1\}$.
  - $Z =$ length of queue at the Coffeehouse at noon: $\{0, 1, \ldots, M_2\}$.
- Stochastic process $\{X_t\}_{t=1}^{T}$ – family of random variables, usually indexed by time.

Stochastic Processes

- Random variable X – function with many possible outcomes, each with some probability.
- Examples:
  - $X =$ outcome of a (fair) coin toss: $\{H, T\}$.
  - $Y =$ cars sold in Houston till date in 2019: $\{0, 1, \ldots, M_1\}$.
  - $Z =$ length of queue at the Coffeehouse at noon: $\{0, 1, \ldots, M_2\}$.
- Stochastic process $\{X_t\}_{t=1}^T$ – family of random variables, usually indexed by time.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses.
  - $(Y_1, \ldots, Y_{365})$: cars sold in Houston up to day t of 2019.
  - $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.

Markov Processes

Markov Processes

- Stochastic process is 'Markovian' if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.

Markov Processes

- Stochastic process is 'Markovian' if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses

    Each toss independent of all previous tosses, hence Markovian.

Markov Processes

- Stochastic process is 'Markovian' if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses

    Each toss independent of all previous tosses, hence Markovian.
  - $(Y_1, \ldots, Y_{365})$: cars sold in Houston up to day t of 2019.

    $Y_{t+1} = Y_t +$ number of cars sold on day t, Markovian.

Markov Processes

- Stochastic process is 'Markovian' if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses

    Each toss independent of all previous tosses, hence Markovian.
  - $(Y_1, \ldots, Y_{365})$: cars sold in Houston up to day t of 2019.

    $Y_{t+1} = Y_t +$ number of cars sold on day t, Markovian.
  - $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.

    $Z_{t+1} = Z_t +$ number of arrivals in hour $t -$ number of people served in hour t, Markovian.

Markov Processes

- Stochastic process is ‘Markovian’ if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses

    Each toss independent of all previous tosses, hence Markovian.
  - $(Y_1, \ldots, Y_{365})$: cars sold in Houston up to day t of 2019.

    $Y_{t+1} = Y_t +$ number of cars sold on day t, Markovian.
  - $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.

    $Z_{t+1} = Z_t +$ number of arrivals in hour $t -$ number of people served in hour t, Markovian.
  - Bag with 5 red and 5 black balls, $W_t =$ color of t-th ball drawn, $t = 1, \ldots, 10$.

    If balls are replaced, all draws are independent; Markov process.

Markov Processes

- Stochastic process is 'Markovian' if X_{t+1} depends only on X_t (and not on $X_1, \ldots, X_{t-1}$).
  Formally, $\mathbb{P}(X_{t+1}|X_t) = \mathbb{P}(X_{t+1}|X_1, \ldots, X_t)$.
- Examples:
  - $(X_1, \ldots, X_{20})$: outcomes of 20 (fair) coin tosses

    Each toss independent of all previous tosses, hence Markovian.
  - $(Y_1, \ldots, Y_{365})$: cars sold in Houston up to day t of 2019.

    $Y_{t+1} = Y_t +$ number of cars sold on day t, Markovian.
  - $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.

    $Z_{t+1} = Z_t +$ number of arrivals in hour $t -$ number of people served in hour t, Markovian.
  - Bag with 5 red and 5 black balls, $W_t =$ color of t-th ball drawn, $t = 1, \ldots, 10$.

    If balls are replaced, all draws are independent; Markov process.
    If balls are not replaced, W_{t+1} depends on <u>all</u> previous draws; not Markovian.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
    - Suppose each counter serves 30 people in an hour.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
  - Suppose each counter serves 30 people in an hour.
  - At the start of hour t:

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
    - Suppose each counter serves 30 people in an hour.
    - At the start of hour t:
    - observe Z_t.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
  - Suppose each counter serves 30 people in an hour.
  - At the start of hour t:
  - observe Z_t.
  - decide how many counters to open (say $a_t = 1$,2 or 3).

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
    - Suppose each counter serves 30 people in an hour.
    - At the start of hour t:
    - observe Z_t.
    - decide how many counters to open (say $a_t = 1$,2 or 3).
    - then, $Z_{t+1} = Z_t - 30a_t +$ number of new arrivals.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
    - Suppose each counter serves 30 people in an hour.
    - At the start of hour t:
    - observe Z_t.
    - decide how many counters to open (say $a_t = 1$,2 or 3).
    - then, $Z_{t+1} = Z_t - 30a_t +$ number of new arrivals.
    - profit = revenue from coffees sold - wages paid to servers.

Markov Decision Processes

- Stochastic process $\{s_t\}$, $t = 0, 1, \ldots, T$.
- Every t, an agent observes s_t, and takes an action a_t.
- Next state s_{t+1} depends on s_t (Markov) and a_t (decision).
- Example: $(Z_1, \ldots, Z_T)$: length of queue at the Coffeehouse every hour.
    - Suppose each counter serves 30 people in an hour.
    - At the start of hour t:
    - observe Z_t.
    - decide how many counters to open (say $a_t = 1,2$ or 3).
    - then, $Z_{t+1} = Z_t - 30a_t +$ number of new arrivals.
    - profit = revenue from coffees sold - wages paid to servers.
    - Goal: maximize total profit.

Markov Decision Processes

Markov Decision Processes

- A Markov decision process (MDP) consists of:
    - Decision epochs $t = 0, 1, \ldots, T-1$.
    - State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
    - Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
    - Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
    - Rewards $R(s, a)$.

Markov Decision Processes

- A Markov decision process (MDP) consists of:
    - Decision epochs $t = 0, 1, \ldots, T-1$.
    - State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
    - Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
    - Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
    - Rewards $R(s, a)$.
- Policy/decision rule π assigns an action to every state.

Markov Decision Processes

- A Markov decision process (MDP) consists of:
    - Decision epochs $t = 0, 1, \dots, T-1$.
    - State space $\mathcal{S} = \{1, 2, \dots, S\}$.
    - Action space $\mathcal{A} = \{1, 2, \dots, A\}$.
    - Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
    - Rewards $R(s, a)$.
- Policy/decision rule π assigns an action to every state.
- Total expected reward for a policy

$$V^\pi(s_0) = \mathsf{E}\left[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\right],$$

where r is a terminal reward.

Markov Decision Processes

- A Markov decision process (MDP) consists of:
  - Decision epochs $t = 0, 1, \ldots, T-1$.
  - State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
  - Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
  - Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
  - Rewards $R(s, a)$.
- Policy/decision rule π assigns an action to every state.
- Total expected reward for a policy

$$V^{\pi}(s_0) = \mathsf{E}\left[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\right],$$

  where r is a terminal reward.
- V^{π} is called the value (function) for policy π.

Markov Decision Processes

- A Markov decision process (MDP) consists of:
  - Decision epochs $t = 0, 1, \ldots, T-1$.
  - State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
  - Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
  - Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
  - Rewards $R(s, a)$.
- Policy/decision rule π assigns an action to every state.
- Total expected reward for a policy

$$V^\pi(s_0) = \mathrm{E}\left[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\right],$$

  where r is a terminal reward.
- V^π is called the value (function) for policy π.
- Goal: find π that maximizes this reward. Optimal value function

$$V^*(s_0) = \max_\pi V^\pi(s_0).$$

Examples: Queuing

- Determine optimal service schedule for incoming jobs.

Examples: Queuing

- Determine optimal service schedule for incoming jobs.
- Decision epochs – every hour.
- State = number of jobs waiting to be served.
- Action = number of jobs accepted for service in each period, $\mathcal{A} = 0, 1, \ldots, s_t\}$.
- Next state = $s_{t+1} = s_t - a_t +$ new arrivals (random).
- Reward for completing a job, cost for keeping jobs waiting.

Examples: Inventory management

- Determine optimal policy for ordering inventory in the face of uncertain demand.

Examples: Inventory management

- Determine optimal policy for ordering inventory in the face of uncertain demand.
- Decision epochs – every week.
- Seller observes the current inventory (state) and decides how much more inventory (action) to acquire for the week.
- Random demand d_t during the week.
- $s_{t+1} = \max\{s_t + a_t - d_t, 0\}$.
- Reward = sale revenue - purchase cost.

Example: medical treatment planning

- Find optimal dosing policy for ‘best’ disease progression.
- Observe a ‘disease score’ (state) of a patient, and prescribe a dose (action) of medication (rheumatoid arthritis, radiation therapy).
- $s_{t+1} = f(s_t, a_t)$ clinically determined.
- Reward = improvement in patient’s condition.

Solving MDPs: Bellman's Principle of Optimality

Recall objective

$$V^*(s_0) = \max_\pi V^\pi(s_0) = \max_\pi \ \mathbf{E}\Big[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big].$$

Solving MDPs: Bellman's Principle of Optimality

Recall objective

$$V^*(s_0) = \max_\pi V^\pi(s_0) = \max_\pi \; \mathbf{E}\Big[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big].$$

Define 'rewards-to-go':

$$V_t^\pi(s_t) = \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big]$$

$$V_t^*(s_t) = \max_\pi \; \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big],$$

Solving MDPs: Bellman's Principle of Optimality

Recall objective

$$V^*(s_0) = \max_\pi V^\pi(s_0) = \max_\pi \ \mathbf{E}\Big[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big].$$

Define 'rewards-to-go':

$$V_t^\pi(s_t) = \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big]$$

$$V_t^*(s_t) = \max_\pi \ \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big],$$

<u>Bellman's principle of optimality</u>

$$V_t^*(s_t) = \max_{a\in\mathcal{A}} \Big\{ R(s_t, a) + \lambda V_{t+1}^*(s_{t+1}) \Big\}$$

$$V_T^*(s_T) = r(s_T).$$

Solving MDPs: Bellman's Principle of Optimality

Recall objective

$$V^*(s_0) = \max_\pi V^\pi(s_0) = \max_\pi \; \mathbf{E}\Big[\sum_{t=0}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big].$$

Define 'rewards-to-go':

$$V_t^\pi(s_t) = \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big]$$

$$V_t^*(s_t) = \max_\pi \; \mathbf{E}\Big[\sum_{t=t}^{T-1} R(s_t, \pi(s_t)) + r(s_T)\Big],$$

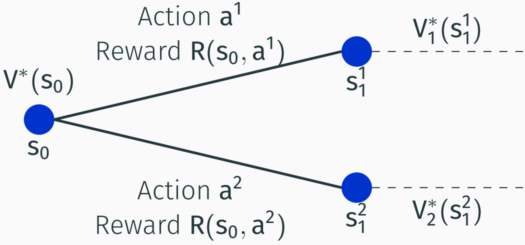
<u>Bellman's principle of optimality</u>

$$V_t^*(s_t) = \max_{a\in\mathcal{A}} \Big\{ R(s_t, a) + \lambda V_{t+1}^*(s_{t+1}) \Big\}$$

$$V_T^*(s_T) = r(s_T).$$

Bellman equations, solved by backward recursion.

Solving MDPs: Bellman's Principle of Optimality



$$V^*(s_0) = \max\left\{R(s_0, a^1) + \lambda V_1^*(s_1^1), R(s_0, a^2) + \lambda V_1^*(s_1^2)\right\}.$$

Infinite-horizon MDPs

- Decision epochs $t = 0, 1, \ldots$
- State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
- Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
- Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
- Rewards $R(s, a)$.

Infinite-horizon MDPs

- Decision epochs $t = 0, 1, \ldots$
- State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
- Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
- Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
- Rewards $R(s, a)$.

Total expected reward for a policy

$$V^\pi(s_0) = \mathbf{E}\left[\sum_{t=0}^{\infty} \lambda^t R(s_t, \pi(s_t))\right],$$

where $\lambda \in (0, 1)$ is a discount factor.

Infinite-horizon MDPs

- Decision epochs $t = 0, 1, \ldots$
- State space $\mathcal{S} = \{1, 2, \ldots, S\}$.
- Action space $\mathcal{A} = \{1, 2, \ldots, A\}$.
- Transition probabilities $P_a(s, s') = \mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$.
- Rewards $R(s, a)$.

Total expected reward for a policy

$$V^\pi(s_0) = \mathbf{E}\left[\sum_{t=0}^{\infty} \lambda^t R(s_t, \pi(s_t))\right],$$

where $\lambda \in (0, 1)$ is a discount factor.

No 'terminal' reward.

- Principle of optimality still holds.

Solving MDPs: Bellman equations

- Principle of optimality still holds.
- Bellman equations

$$V^*(s) = \max_{a \in \mathcal{A}} \left\{ R(s, a) + \lambda \mathbf{E}[V^*(s')] \right\}.$$

Solving MDPs: Bellman equations

- Principle of optimality still holds.
- Bellman equations

$$V^*(s) = \max_{a \in \mathcal{A}} \left\{ R(s, a) + \lambda \mathbf{E}[V^*(s')] \right\}.$$

- No backward recursion.

Solving MDPs: Bellman equations

- Principle of optimality still holds.
- Bellman equations

$$V^*(s) = \max_{a \in \mathcal{A}} \left\{ R(s, a) + \lambda \mathbf{E}[V^*(s')] \right\}.$$

- No backward recursion.
- Solution methods:
  - Value iteration
  - Policy iteration
  - Linear programming

Value iteration

- Define Bellman operator

$$\mathcal{L}(v)(s) = \max_{a \in \mathcal{A}} \; R(s,a) + \lambda \mathbf{E}[v(s')].$$

- V^* is the fixed point of this operator.
- Value iteration relies on the Contraction Mapping Theorem.
- Algorithm:
    - Initial guess $V^{(0)}$.
    - Compute $V^{(k)} = \mathcal{L}(V^{(k-1)})$.
    - Iterate until stopping condition.

Policy Iteration

- Now we iterate over policies to find a better one.
- Algorithm:
  - Initial guess $\pi^{(0)}$.
  - Evaluate policy using matrix inversion:

$$V^{\pi^{(k)}}(s) = R(s, \pi(s)) + \mathrm{E}[V^{(\pi^{(k)}}(s')].$$

  - Improve policy

$$\pi^{(k+1)}(s) = \operatorname*{argmax}_{a \in \mathcal{A}} \left\{ R(s, a) + \mathrm{E}[V^{(\pi^{(k)}}(s')] \right\}.$$

  - Iterate until stopping condition.

Linear Programming

- Note that

$$\begin{aligned} V^*(s) &= \max_{a \in \mathcal{A}} \left\{ R(s,a) + \mathbf{E}[V^*(s')] \right\} \; \forall \, s \\ &\geq R(s,a) + \mathbf{E}[V^*(s')] \; \forall \, s, a. \end{aligned}$$

- V^* is the solution to the following LP:

$$\begin{aligned} \min \; & \sum_{s \in \mathcal{S}} \beta(s) v(s) \\ \text{s.t. } & v(s) \geq R(s,a) + \mathbf{E}[v(s')] \; \forall \, s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned}$$

Conclusion

- MDPs form a flexible modeling environment for sequential decision making problems.
- Several extensions – partially observable MDPs, reinforcement learning.
- Limitations:
  - 'Curse of dimensionality'
  - Data requirements.