Dynamic programming, Part 2

Solving stochastic finite horizon MDPs

Cathy Wu

1.041/1.200 Transportation: Foundations and Methods

Readings

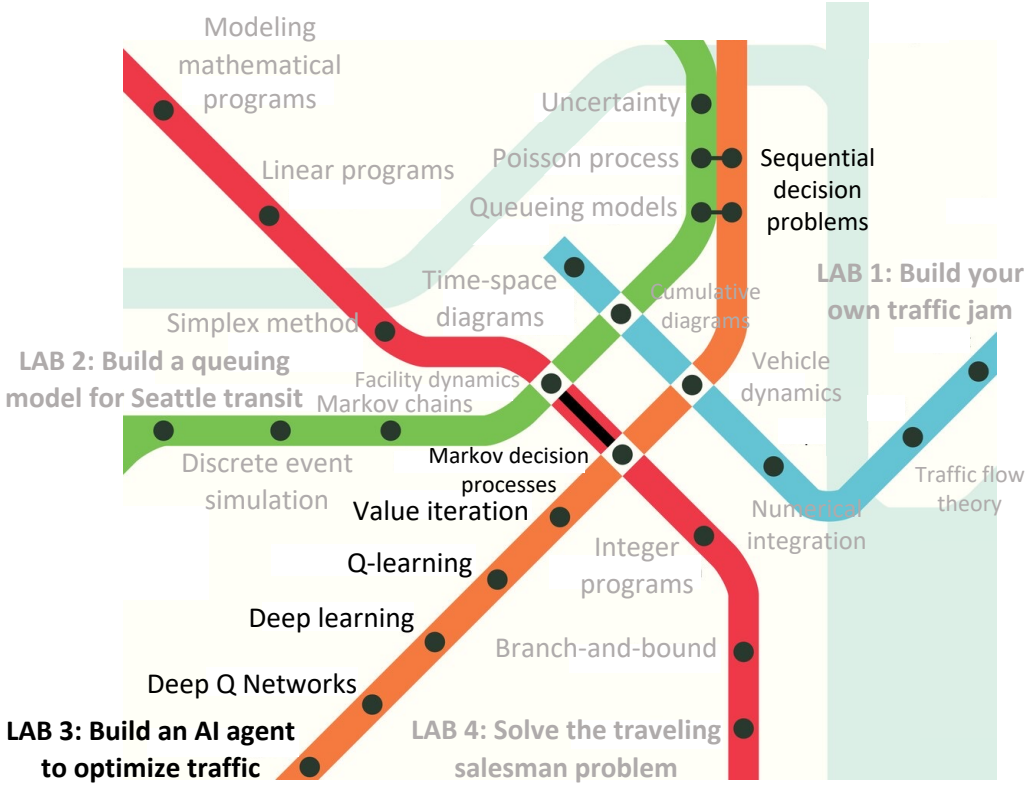
1. (Optional) Bertsekas, D. P. (2005). **Dynamic programming and optimal control**, vol 1. *Belmont, MA: Athena Scientific*, 3rd Edition. Chapter 1: Introduction.

Unit 3: Machine learning for traffic control

Unit 3

Optimizing

Multi-stage



Outline

1. Dynamic programming in transportation
2. Dynamic programming for stochastic problems
3. Parking problem
4. Dynamic programming analysis

Outline

1. **Dynamic programming in transportation**
2. Dynamic programming for stochastic problems
3. Parking problem
4. Dynamic programming analysis

Dynamic programming in transportation

- Optimal capacity expansion
- Routing
- Scheduling
- Machine replacement
- Inventory management
- Queue management
- Production planning
- Optimal stopping

+

- Planes
  - Aviation
  - Drones
  - Runways
- Train
- Automobiles
  - Electric chargers
  - Connected / automated
- Buses
- Micromobility
  - Bikes
  - Scooters
- Ships
- …

Bertsekas, D. P. (2005). **Dynamic programming and optimal control**, vol 1. *Belmont, MA: Athena Scientific*, 3rd Edition. Chapter 1 & 3 (See examples)

Outline

1. Dynamic programming in transportation
2. **Dynamic programming for stochastic problems**
3. Parking problem
4. Dynamic programming analysis

Recall: Deterministic vs stochastic sequential problems

- A deterministic policy is a special case of a stochastic policy when $\pi(a|s)$ is a unit spike at $a = \pi(s)$ for all $s \in \mathcal{S}$ (and 0 otherwise).
- A deterministic transition is a special case of a stochastic transition when $p(s'|s,a)$ is a unit spike at $s' = \mathrm{f_t(s,a)}$ for all $s \in \mathcal{S}, a \in A$ (and 0 otherwise).

That is, a deterministic sequential decision problem is a special case of a stochastic sequential problem. It can still be modeled within the MDP framework.

Uncertainty in Transportation

Recalling from Unit 2, there is uncertainty everywhere in transportation and these can be represented in an MDP, via a general $p(s'|s,a)$.

Examples:

- Public transportation: Bus arrivals at the start of a day are more on-time than at the end of a day.
- Driving and traffic conditions on a road.
- The arrival time of your Lyft Line ride.
- Airline departures and delays.
- Etc., etc.

If we can identify the factors that contribute to the uncertainty, then we can incorporate them into the MDP model.

Recall: Deterministic dynamic programming algorithm

$V_T(s_T) = r_T(s_T)$
for $t = T-1, \dots, 0$ **do**
  for $s_t \in \mathcal{S}_t$ **do**
    $V_t(s_t) = \max_{a_t \in \mathcal{A}_t(s_t)} r_t(s_t, a_t) + V_{t+1}(s_{t+1})$ where $s_{t+1} = f(s_t, a_t)$
end for

- **Proof**: by induction
- "Efficient": O(|S||A|T)
- Equivalent to Bellman-Ford algorithm
- **Strength**: Generality
- Much better than naive approach O(T!)
- **Weakness**: ALL the tail subproblems are solved

Dynamic programming algorithm

$V_T(s_T) = r_T(s_T)$
for $t = T-1, \dots, 0$ **do**
  for $s_t \in \mathcal{S}_t$ **do**
    $V_t(s_t) = \max_{a_t \in \mathcal{A}_t(s_t)} r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}[V_{t+1}(s_{t+1})]$
end for

Dynamic programming algorithm

$V_T(s_T) = r_T(s_T)$
for $t = T-1, \ldots, 0$ **do**
  for $s_t \in \mathcal{S}_t$ **do**
    $V_t(s_t) = \max_{a_t \in \mathcal{A}_t(s_t)} r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}[V_{t+1}(s_{t+1})]$
end for

- **Proof**: by induction
- "Efficient": O(|S|^2|A|T)

- For deterministic shortest path routing
  - Equivalent to Bellman-Ford algorithm
  - **Strength**: Generality
  - "Efficient": O(|S||A|T)
  - Much better than naive approach O(T!)
  - **Weakness**: ALL the tail subproblems are solved

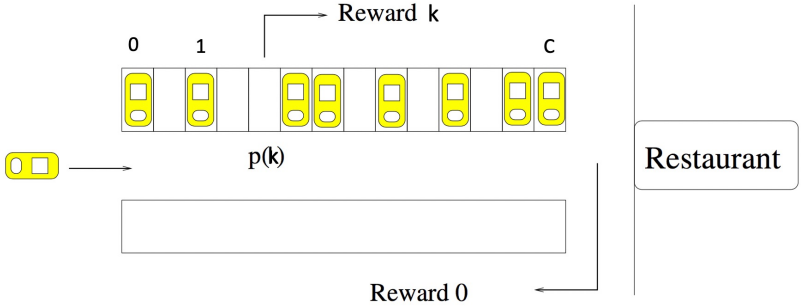
Outline

1. Dynamic programming in transportation
2. Dynamic programming for stochastic problems
3. **Parking problem**
4. Dynamic programming analysis

Example: Parking Problem

A driver wants to park as close as possible to the restaurant.

Objective: maximize the satisfaction
Problem: Formulate the parking problem as an MDP.



We know (assume) that:

- The driver cannot see if a spot is available unless in front of it.
- There are $C + 1$ parking spots.
- Cannot move backwards. At each place k the driver can either move to the next spot or park (if the place is available).
- Each spot is free with probability $p(k)$, independently of the other spots.
- The closer to the restaurant, the higher the satisfaction. Assume that satisfaction grows inversely with the distance to the restaurant.
- However, parking sooner gives the driver satisfaction. In fact, parking at each later slot gives a factor of 0.9 less satisfaction.
- If the driver doesn't park anywhere, then the driver leaves the restaurant and has to find another place to eat.

Adapted from DPOC vol1 (2005) Chapter 3 and Lazaric (2014) Lecture 2

Applications

- This is an instance of an optimal stopping problem.
- At each stage, the decision maker observes the current state of the system and decides whether to continue the process (perhaps at a certain cost) or stop the process and incur a certain loss.
- Applications in transportation:
  - Curbside management: mitigating effects of mobility-on-demand pick-up & drop-offs
  - Infrastructure investment: when to buy land and build infrastructure
  - Dynamic scheduling / vehicle routing: wait for more information or decide
- Famous application: secretary problem

Further reading

[1] P. N. Stueger, F. Fehn, and K. Bogenberger, "Minimizing the Effects of Urban Mobility-on-Demand Pick-Up and Drop-Off Stops: A Microscopic Simulation Approach," *Transportation Research Record*, vol. 2677, no. 1, pp. 814–828, Jan. 2023, doi: 10.1177/03611981221101894.
[2] J.-D. Saphores and M. Boranet, "Investing in urban transportation infrastructure under uncertainty," in *8th annual real options conference*, 2004.
[3] N. Vodopivec and E. Miller-Hooks, "An optimal stopping approach to managing travel-time uncertainty for time-sensitive customer pickup," *Transportation Research Part B: Methodological*, vol. 102, pp. 22–37, Aug. 2017, doi: 10.1016/j.trb.2017.04.017.

Applications

1.200 Class project ideas (Spring 2024)

What if Google Maps routed drivers to available parking, rather than straight to the destination? Current navigation apps send drivers to their destination, and then drivers spend considerable time (and thus energy) manually looking for parking (un-aided by technology). What would be the impact if navigation apps incorporated the search for available parking into their routing recommendation? Model and design this routing feature. Analyze the amount of time wasted by drivers looking for parking in an urban area. Time permitting, incorporate some real data to inform your analysis.

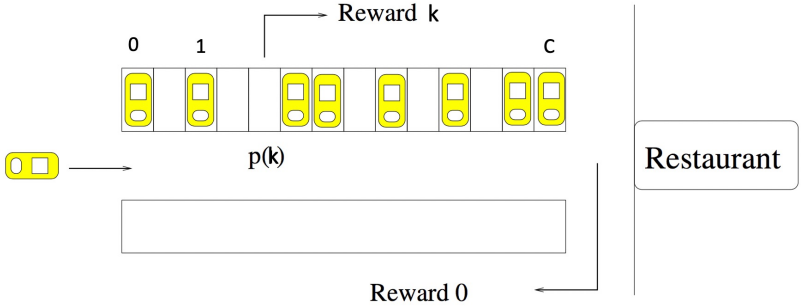
Project mentor(s): Cathy Wu

1.200 Class project ideas – Notion

Example: Parking Problem

A driver wants to park as close as possible to the restaurant.

Objective: maximize the satisfaction
Problem: Formulate the parking problem as an MDP.



Example: Parking Problem, Part II (Lazaric, 2014, Lecture 2)

- Let's solve the parking problem using dynamic programming principles, but now, considering the stochastic nature of the problem.
- As before, we start from the end.
- We use an equivalent discounted version of the dynamic programming recursion:

$$V_k^*(s) = \max_{a \in \mathcal{A}} r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V_{k+1}^*(s')]$$
$$V_C^*(s) = \max_{a \in \mathcal{A}} r(s,a) \quad \text{(terminal reward)}$$

Outline

1. Dynamic programming in transportation
2. Dynamic programming for stochastic problems
3. Parking problem
4. **Dynamic programming analysis**

Dynamic programming algorithm

$V_T(s_T) = r_T(s_T)$
for $t = T-1, \ldots, 0$ **do**
  for $s_t \in \mathcal{S}_t$ **do**
    $V_t(s_t) = \max_{a_t \in \mathcal{A}_t(s_t)} r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}[V_{t+1}(s_{t+1})]$
end for

- **Proof**: by induction
- "Efficient": O(|S|²|A|T)

- For deterministic shortest path routing
  - Equivalent to Bellman-Ford algorithm
  - **Strength**: Generality
  - "Efficient": O(|S||A|T)
  - Much better than naive approach O(T!)
  - **Weakness**: ALL the tail subproblems are solved

Proof of the induction step

Denote tail policy from time t onward as $\pi_{t:T-1} = \{\pi_t, \pi_{t+1}, \dots, \pi_{T-1}\}$

Assume that $V_{t+1}(s_{t+1}) = V_{t+1}^*(s_{t+1})$. Then:

$$V_t^*(s_t) = \max_{(\pi_t, \pi_{t+1:T-1})} \mathbb{E}_{s_{t+1:T-1}} \left\{ r_t\big(s_t, \pi_t(s_t)\big) + r_T(s_T) + \sum_{i=t+1}^{T-1} r_i\big(s_i, \pi_i(s_i)\big) \right\}$$

$$= \max_{\pi_t} r_t\big(s_t, \pi_t(s_t)\big) + \max_{\pi_{t+1:T-1}} \left[\mathbb{E}_{s_{t+1:T-1}} \left\{ r_T(s_T) + \textstyle\sum_{i=t+1}^{T-1} r_i\big(s_i, \pi_i(s_i)\big) \right\} \right] \qquad \text{[exchange]}$$

$$= \max_{\pi_t} r_t\big(s_t, \pi_t(s_t)\big) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, \pi_t(s_t))} \left\{ \max_{\pi_{t+1:T-1}} \left[\mathbb{E}_{s_{t+2:T-1}} \left\{ r_T(s_T) + \textstyle\sum_{i=t+1}^{T-1} r_i\big(s_i, \pi_i(s_i)\big) \right\} \right] \right\}$$

$$= \max_{\pi_t} r_t\big(s_t, \pi_t(s_t)\big) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, \pi_t(s_t))} \{ V_{t+1}^*(s_{t+1}) \} \qquad \text{[definition]}$$

$$= \max_{\pi_t} r_t\big(s_t, \pi_t(s_t)\big) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, \pi_t(s_t))} \{ V_{t+1}(s_{t+1}) \} \qquad \text{[induction hypothesis]}$$

$$= \max_{a_t \in \mathcal{A}_t(s_t)} r_t(s_t, a_t) + \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)} \{ V_{t+1}(s_{t+1}) \}$$

$$= V_t(s_t) \qquad \text{[DP algorithm]}$$

Interpretation as optimal reward-to-go (cost-to-go) function.

Proof of the induction step

For the =, we have:

$$\max_{\pi'} \sum_{s'} p(s'|s,a) V^{\pi'}(s') \leq \sum_{s'} p(s'|s,a) \max_{\pi'} V^{\pi'}(s')$$

But, let $\bar{\pi}(s') = \operatorname{argmax}_{\pi'} V^{\pi'}(s')$

$$\sum_{s'} p(s'|s,a) \max_{\pi'} V^{\pi'}(s') \leq \sum_{s'} p(s'|s,a) V^{\bar{\pi}}(s') \leq \max_{\pi'} \sum_{s'} p(s'|s,a) V^{\pi'}(s')$$

■

References

1. Bertsekas, D. P. (2005). Dynamic programming and optimal control, vol 1. *Belmont, MA: Athena Scientific*, 3rd Edition.
2. Lazaric, A. (2014). Master MVA: Reinforcement Learning.
3. With many slides adapted from Alessandro Lazaric and Matteo Pirotta.