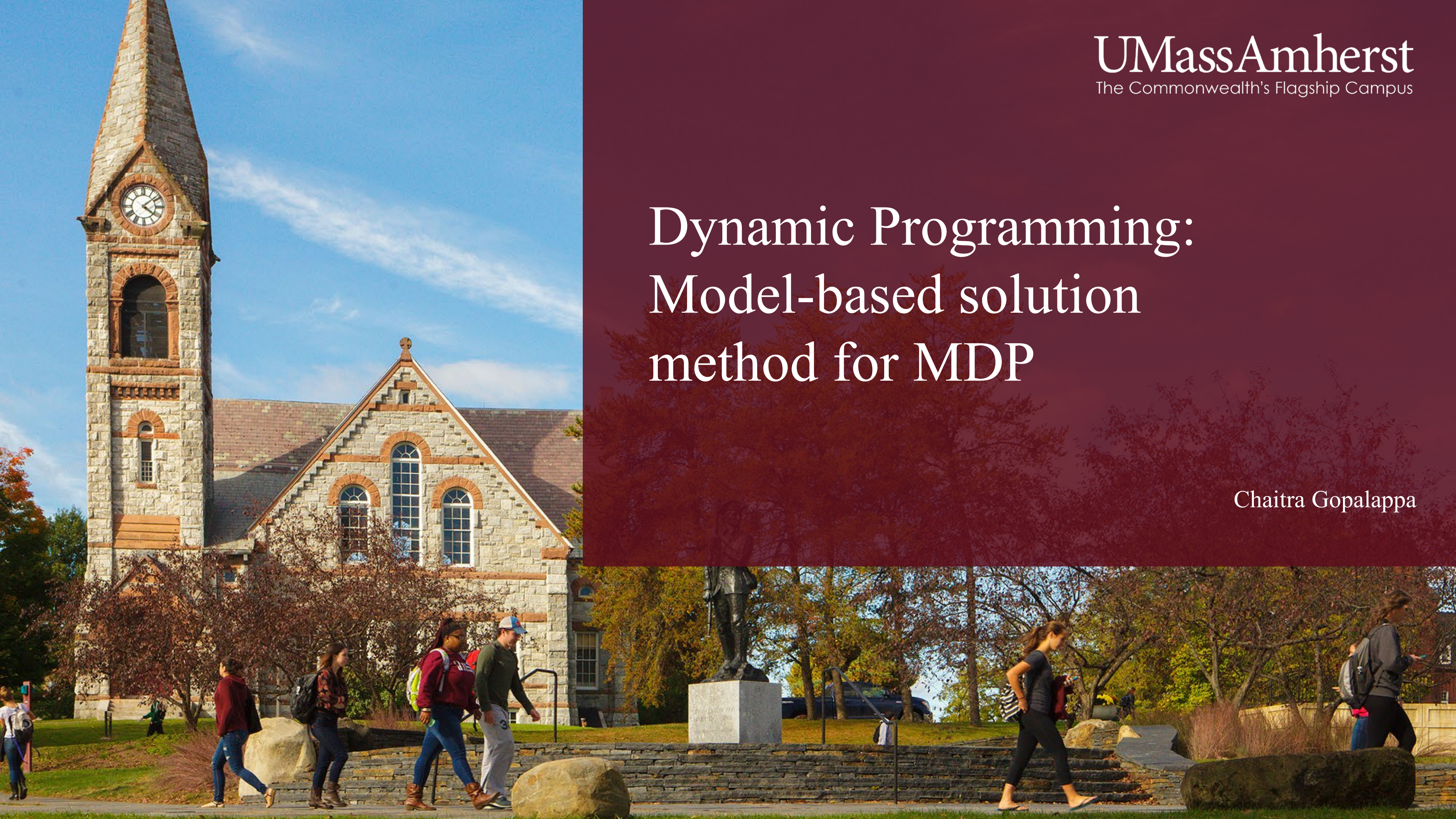
UMassAmherst
The Commonwealth's Flagship Campus

Dynamic Programming: Model-based solution method for MDP

Chaitra Gopalappa



Reviewing MDP terminology for a refresher

Caution: Unless otherwise specified, here on:

- π
  - is a policy (not steady state distribution like in Markov chain)
- Value function:
  - $v_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s]; \forall s \in S$

Objective of MDP: Find policy with maximum state-value function

- $v_*(s) = \max_{\pi} v_\pi(s), \forall s \in S$
  - $v_\pi(s)$: value function of state s under policy π
  - It is the **expected** reward when **starting** in state s and following some policy π
  - $v_*(s)$: value function of state s under optimal policy π^*
- Later-on we will use other objective functions

Expand state-value function

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right]$$

$$= \mathbb{E}_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots .. | S_t = s, \pi]$$

$$= \mathbb{E}_\pi[R_t + \gamma(R_{t+1} + \gamma R_{t+2} + \cdots ..) | S_t = s, \pi]$$

$$= \mathbb{E}_\pi[R_t + \gamma(v_\pi(S_{t+1}))\ | S_t = s, \pi]$$

$$= \sum_{s'} p(s, \pi(s), s')[r(s, \pi(s), s') + \gamma v_\pi(s')]$$

Dynamic programming
(Chapter 4, Sutton and Barto)

DP Algorithms

- Policy iteration solved as system of equations
- Policy iteration (this is the one generally referred to)
- Value iteration

Bellman equation for a "**fixed**" (deterministic) policy π

$$V(s) = \sum_{s'} p(s, \pi(s), s')[r + \gamma V(s')]$$

- $r = r(s, \pi(s), s')$; r here is deterministic
- Note: if there is randomness in value of r we rewrite the value function as
  - $V(s) = \sum_{s',r} [p(s, \pi(s), s', r)[r + \gamma V(s')]\ ]$

Bellman "optimality" equation

- Let v^* be the optimal value function for the MDP. The function v^* satisfies, for each $s \in S$. the following
  - $v^*(s) = max_{a \in A} \sum_{s',r}[p(s',r \mid s,a) \cdot (r(s,a,s') + v^*(s'))]$
  - Or if there is no randomness in rewards,
  - $v^*(s) = max_{a \in A} \sum_{s'}[p(s' \mid s,a) \cdot (r(s,a,s') + v^*(s'))]$
- Furthermore, it is the only function satisfying the property

Some basic theories

- Definition: A **stationary policy**, defined by the function π takes actions $\pi(i)$ at time n if $X_n = i$, independent of previous states, previous actions, and time n
- Property: if the state space S is finite, there exists a stationary policy that solves the problem
  - $\boldsymbol{v^*(s) = \max_{\pi} v_{\pi}(s)\,, \forall s \in \; S}$
- Definition: A function is called invariant with respect to an operation, if the operation does not vary the function.
  - If the invariant function is unique, then it is called a 'fixed point' for the operation

Some basic theories

- Definition: A **stationary policy**, defined by the function π takes actions $\pi(i)$ at time n if $X_n = i$ independent of previous states, previous actions, and time n
- Property: if the state space S is finite, there exists a stationary policy that solves the problem
  - $\boldsymbol{v^*(s) = \max_{\pi} v_{\pi}(s), \forall s \in S}$
- Definition: A function is called invariant with respect to an operation, if the operation does not vary the function.
  - If the invariant function is unique, then it is called a 'fixed point' for the operation

Note: similarity in steady state/ stationary distribution (ρ) of a Markov process, $\rho \mathrm{P} = \rho$ Multiplying by P does not change the value of ρ, so ρ is the fixed point for the MC defined by P

Fixed point theorem to solve for Bellman operators

- Solving for Bellman equation for a "**given**" (deterministic) policy π

$$V(s) = \sum_{s'} p(s, \pi(s), s')[r + \gamma V(s')]$$

Equivaent to writing: $\boldsymbol{V_\pi = B_\pi V_\pi}$

B_π: Bellman operator for a policy π;

Applying Banach fixed-point theorem, for a given policy π, we can solve for V_π by starting at some random value V

- Solving for Bellman **optimality** equation

$$v^*(s) = max_{a \in A} \sum_{s'} \left[p(s' \mid s, a) \cdot \left(r(s, a, s') + \gamma v^*(s')\right)\right]$$

Equivalent to writing: $\boldsymbol{V^*_{\pi^*} = B^*_{\pi^*} V^*_{\pi^*}}$ ***or*** $\boldsymbol{V^* = B^* V^*}$

$\boldsymbol{B^*}$: Bellman optimality operator (corresponding to optimal policy π^*;

Applying Banach fixed-point theorem we can solve for $v^*(s)$ by starting at some random value V

Policy iteration solved as system of equations

1. Initialize $\pi(s) \in \mathcal{A}(s)$; arbritrarily $\forall s \in \mathcal{S}$

2. Policy evaluation

- *Solve for $V(s)$, each $s \in \mathcal{S}$, by solving following system of equations*
  - $V(s) = \sum_{s',r}\left[p(s',r \mid s,\pi(s)) \cdot \left(r + \gamma \cdot V(s')\right)\right]; r = R(s,\pi(s),s')$
- 3. Policy improvement
- *For each $s \in \mathcal{S}$:*
  - $\pi'(s) = argmax_a \sum_{s',r}\left[p(s',r \mid s,a) \cdot \left(r + \gamma \cdot V(s')\right)\right]$
- If $\pi'(s) = \pi(s)$, stop and return $v_* = V(s), \pi_* = \pi$, else goto step 2

Policy Iteration – DP

$\{X_t, D_t\}_{t=0}^{n}$ is a MDP represented by a tuple (Ω, P, R, A); $\Omega = \{1,2\}$
$A = \{a_1, a_2\}$

$P_{a_1} = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$; $P_{a_2} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$; $R_{a_1} = \begin{bmatrix} 6 & -5 \\ 7 & 12 \end{bmatrix}$; $R_{a_2} = \begin{bmatrix} 10 & 17 \\ -14 & 13 \end{bmatrix}$; Find optimal policy

① let $\pi = [a_1, a_1]$; initializing π; λ = discounting factor

② Policy evaluation $V(s) = \sum_{s',r} [p(s', r | s, \pi(s)) \cdot (r + \lambda V(s'))]$; $r = R(s, \pi(s), s')$

$V(1) = [0.7 \;\; 0.3] \begin{bmatrix} 6 + \lambda V(1) \\ -5 + \lambda V(2) \end{bmatrix} = 2.7 + [0.7 V(1) + 0.3 V(2)] \lambda$

$V(2) = [0.4 \;\; 0.6] \begin{bmatrix} 7 + \lambda V(1) \\ 12 + \lambda V(2) \end{bmatrix} = 10 + [0.4 V(1) + 0.6 V(2)] \lambda$

Solve for $V(1)$ and $V(2) \Rightarrow V(1) = 54$; $V(2) = 64$

③ Policy improvement
$\pi'(s) = \arg\max_a \sum_{s',r} [p(s', r | s, a) \cdot (r + \lambda V(s'))]$

$\pi'(1) = \arg\max_a \left\{ [0.7 \;\; 0.3] \begin{bmatrix} 6 + \lambda 54 \\ -5 + \lambda 64 \end{bmatrix} ; [0.9 \;\; 0.1] \begin{bmatrix} 10 + \lambda 54 \\ 17 + \lambda 64 \end{bmatrix} \right.$

$= \arg\max_a \{ 54 \quad ; \quad 60.2$

$= a_2$

$\pi'(2) = \arg\max_a \left\{ [0.4 \;\; 0.6] \begin{bmatrix} 7 + \lambda 54 \\ 12 + \lambda 64 \end{bmatrix} ; [0.2 \;\; 0.8] \begin{bmatrix} -14 + \lambda 54 \\ 13 + \lambda 64 \end{bmatrix} \right.$

$= \arg\max \{ 64 \quad ; \quad 63.4$

$= a_1$

$\pi' = [a_2, a_1] \Rightarrow \pi \neq \pi' \Rightarrow$ set $\pi = \pi'$ goto step 2

② policy evaluation : $\pi = [a_2, a_1]$

$V(1) = [0.9 \;\; 0.1] \begin{bmatrix} 10 + \lambda V(1) \\ 17 + \lambda V(2) \end{bmatrix} = 10.7 + \lambda [0.9 V(1) + 0.1 V(2)]$

$V(2) = [0.4 \;\; 0.6] \begin{bmatrix} 7 + \lambda V(1) \\ 12 + \lambda V(2) \end{bmatrix} = 10 + \lambda [0.4 V(1) + 0.6 V(2)]$

Two equations, two unknowns
Solve for $V(1)$ and $V(2)$; $V(1) = 105.85$; $V(2) = 104.58$

③ policy improvement

$\pi'(1) = \arg\max_a \left\{ [0.7 \;\; 0.3] \begin{bmatrix} 6 + \lambda 105.85 \\ -5 + \lambda 104.58 \end{bmatrix} ; [0.9 \;\; 0.1] \begin{bmatrix} 10 + \lambda 105.85 \\ 17 + \lambda 104.58 \end{bmatrix} \right.$

$= \arg\max_a \{ 97.6 \quad ; \quad 105.85$

$= a_2$

$\pi'(2) = \arg\max_a \left\{ [0.4 \;\; 0.6] \begin{bmatrix} 7 + \lambda 105.85 \\ 12 + \lambda 104.58 \end{bmatrix} ; [0.2 \;\; 0.8] \begin{bmatrix} -14 + \lambda 105.85 \\ 13 + \lambda 104.58 \end{bmatrix} \right.$

$= \arg\max_a \{ 104.58 \quad ; \quad 101.95$

$= a_1$

$\pi' = [a_2, a_1] \Rightarrow \pi = \pi' \Rightarrow \pi = [a_2, a_1]$ is the optimal policy

Fixed point theorem to solve for Bellman operators

- Solving for Bellman equation for a "**given**" (deterministic) policy π

$$V(s) = \sum_{s'} p(s, \pi(s), s')[r + \gamma V(s')]$$

Equivaent to writing: $\boldsymbol{V_\pi = B_\pi V_\pi}$

B_π: Bellman operator for a policy π;

Applying Banach fixed-point theorem, for a given policy π, we can solve for V_π by starting at some random value V

Policy iteration

1. Initialize $V(s) \in \mathbb{R}, \pi(s) \in \mathcal{A}(s)$; arbritrarily $\forall s \in \mathcal{S}$; *set tolerance* θ *(=1e-6)*

2. Policy evaluation

- *Loop while* $\Delta > \theta$*:*
  - $\Delta \leftarrow 0$
  - *Loop for each* $s \in \mathcal{S}$
    - $v \leftarrow V(s)$
    - $V(s) = \sum_{s',r} p(s', r \mid s, \pi(s))[r + \gamma V(s')]$
    - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

Applies fixed-point theorem to solve Bellman equation for a *fixed* policy
Note: This can be applied to any policy

- 3. Policy improvement
- *For each* $s \in \mathcal{S}$*:*
  - $\pi'(s) = argmax_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')]$
- If $\pi'(s) = \pi(s)$, stop and return $v_* = V(s), \pi_* = \pi$, else goto step 2

Finite horizon v infinite time horizon

- Infinite time horizon:
  - e.g., What is optimal inventory ordering policy (DP assignment)
  - Discounting bounds the value function; no changes needed in formulation
  - What if discounting is not preferred? (discounting gives less value to future rewards)
    - $V(s) = \sum_{s',r}\left[p(s',r \mid s,\pi(s)) \cdot \left(R(s,\pi(s),s') + \gamma \cdot V(s')\right)\right] - \bar{V}$; $\bar{V}$ =average reward associated with the policy under evaluation (unknown); 1 more unknown than number of equation→ set one of V(s) =0 and solve for the other elements
- Finite time horizon:
  - e.g., game (win, lose); health progression over a lifetime; robotic tasks
  - Make the final state a terminal state ($\Pr(\mathit{terminal\ state}, \mathit{terminal\ state}) = 1$)
  - Make reward for $R(\mathit{terminal\ state}, a, \mathit{terminal\ state}) = 0$; you can give one time-reward for the transition, e.g., $R(:, a, win_{state}) = \mathit{high\ reward}$
  - You may choose to make discounting factor = 1; as time is sufficiently small, it bounds the value function

Fixed-point theorem

- In **policy iteration** we first did policy evaluation
  - Based on Bellman's equation, for any policy π, there exist a fixed–point solution V
  - $V(s) = \sum_{s',r}\left[p(s',r \mid s,\pi(s)) \cdot \left(R(s,\pi(s),s') + \gamma \cdot V(s')\right)\right]$
  - Therefore, for any π , we can solve for $V(s)$, by starting with some arbitrary value and iterating through until we arrive at the fixed point (solution converges)
- Applying the fixed-point theorem, we also have
  - If $\boldsymbol{v}_*$ is the optimal value function for a given MDP, then it satisfies the following equation
  - $v_*(s) = \max_{a \in \mathcal{A}} \sum_{s',r}\left[p(s',r \mid s,a) \cdot \left(R(s,a,s') + \gamma \cdot v_*(s')\right)\right]$
  - Then why not directly aim for iteratively solving for $\boldsymbol{v}_*$? (→value iteration)

Value iteration

1. Initialize $V(s) \in \mathbb{R}$ arbritrarily $\forall s \in \mathcal{S}$ *except set* $V(terminal) 0$*; set tolerance* θ *(=1e-6); set* Δ *to positive value*

2. Find optimal value function

- *Loop while* $\Delta > \theta$*:*
  - $\Delta \leftarrow 0$
  - *Loop for each* $s \in \mathcal{S}$
    - $v \leftarrow V(s)$
    - $V(s) = \max_{a \in \mathcal{A}} \sum_{s',r} \left[p(s', r \mid s, a) \cdot \left(R(s, a, s') + \gamma \cdot V(s') \right) \right]$
    - $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
- 3. Find corresponding optimal policy
  - $\pi(s) = argmax_a \sum_{s',r} \left[p(s', r \mid s, a) \cdot \left(R(s, a, s') + \gamma \cdot V(s') \right) \right]$

UMassAmherst

The Commonwealth's Flagship Campus