Sequential decisions under uncertainty

KTH/EES PhD course

Lecture 3

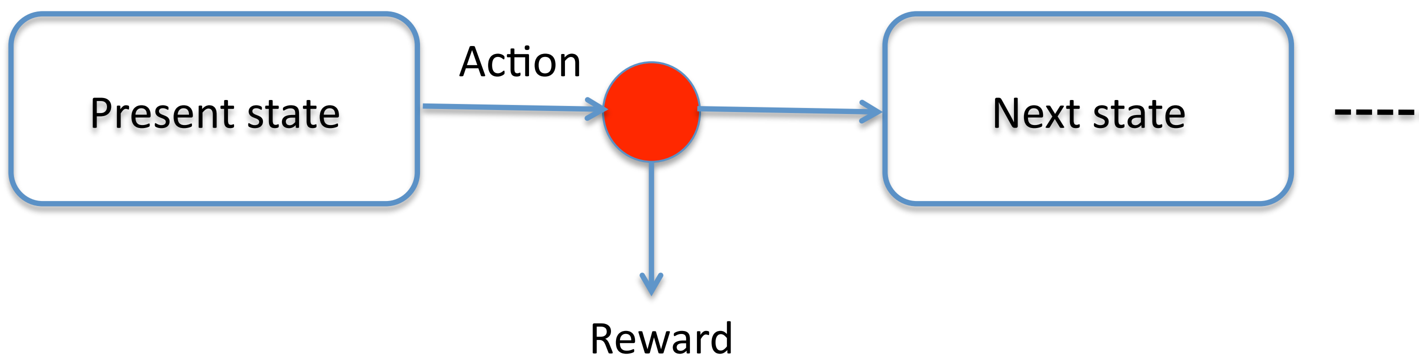
Lecture 3

- Finite-horizon Markov Decision Processes
  - Two deterministic examples
  - Optimal monotone policies

- Infinite-horizon MDPs with discount

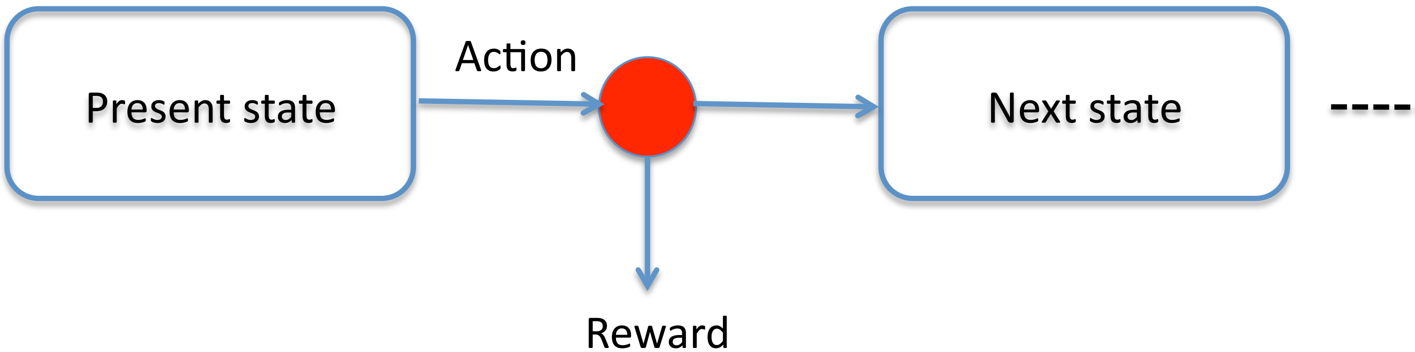
Finite-horizon Markov Decision Processes

States, actions, time horizon



- Set of states: S
- Set of actions available in state *s*: $A_s, \quad A = \cup_{s \in S} A_s$
- These sets are finite, countably infinite, or compacts subsets of a Euclidian space (finite dimension)
- Time horizon *N*: $t \in \{1, \ldots, N\}$

Rewards and transitions



- Reward when selecting at time *t* action *a* in state *s*: $r_t(s,a)$
  It could also depend on the next state: $r_t(s,a,s')$
- Reward at time *N*: $r_N(s)$
- Probability to move from state *s* to *s'* when selecting at time *t* action *a*: $p_t(s'|s,a)$

Algorithm: Optimal MD policy

1. For $t = N, \quad u_N(s) = r_N(s), \forall s \in S$
2. Until $t = 1$

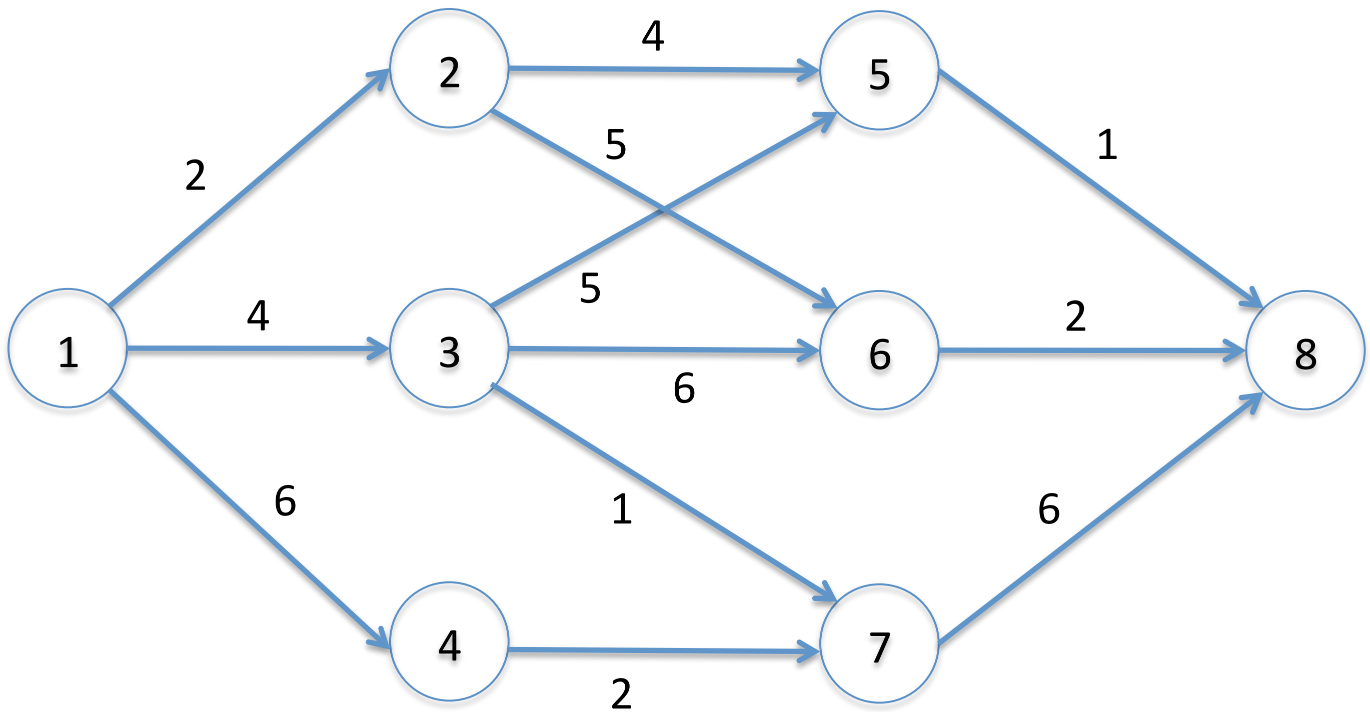
$$t - 1 \rightarrow t$$

$$\forall s_t \in S:$$

$$u_t(s_t) = \max_{a \in A_{s_t}} \left[r_t(s_t, a) + \sum_{j \in S} p_t(j|s_t, a) u_{t+1}(j) \right]$$

$$A^\star_{s_t,t} = \arg\max_{a \in A_{s_t}} \left[r_t(s_t, a) + \sum_{j \in S} p_t(j|s_t, a) u_{t+1}(j) \right]$$

Ex1: routing



Find the max-weight path from source 1 to destination 8

DP formulation

- States (positions): 1, 2, 3, 4, 5, 6, 7, 8
- Actions: from a state, the possible next states
- Rewards: edge weights
- Transitions: deterministic
- Max total reward from state *s*: $u^\star(s)$
- Bellman's equations lead to:

$$u^\star(8) = 0$$

$$u^\star(5) = 1 \quad u^\star(6) = 2 \quad u^\star(7) = 6$$

$$u^\star(2) = 7 \quad u^\star(3) = 8 \quad u^\star(4) = 8$$

$$u^\star(1) = 12$$

Ex2: optimization

- Objective: $\min \quad g_1(x_1) + \ldots + g_N(x_N)$

$$s.t. \quad x_1 + \ldots + x_N = B$$

- Time horizon N
- State: remaining "budget"
- Reward at time i: $g_i(x_i)$
- Example: $g_i(u) = u^2$

$$u_N^\star(b) = b^2$$

$$u_{N-1}^\star(b) = \min_{x \le b}(x^2 + (b-x)^2) = b^2/2$$

$$\ldots$$

$$u_1^\star(b) = b^2/N$$

Optimality of monotone policies

- Do optimal policies have specific structures?
- Example: are they monotone?

$$S = \mathbb{N} \qquad (s \leq s' \Rightarrow a_s \leq a_{s'})?$$
$$A = \mathbb{R}_+$$

Super-additive functions

- $f : X \times Y \to \mathbb{R}$ is super-additive iff

  $X, Y \subset \mathbb{R}^n$

$$f(x^+, y^+) + f(x^-, y^-) \geq f(x^+, y^-) + f(x^-, y^+)$$

when $x^+ \geq x^-, \quad y^+ \geq y^-$

Montone and optimal policy

$$S = \mathbb{N}, A = \mathbb{R}_+$$

$$q_t(k|s,a) = \sum_{j \geq k} p_t(j|s,a)$$

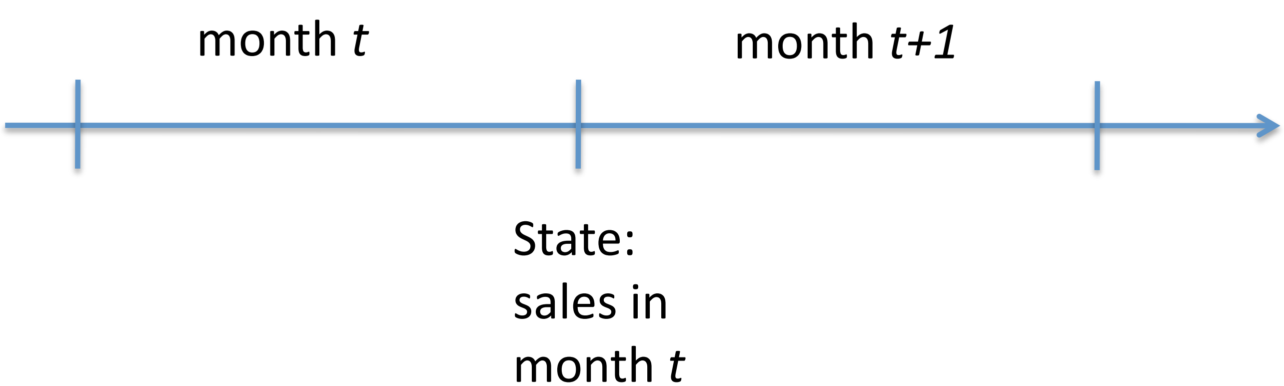
Theorem If

1. $r_t(s,a)$ is nondecreasing in s, and super-additive
2. $q_t(k|s,a)$ is nondecreasing in s, and super-additive

Then there exists an optimal nondecreasing policy

Example

- Optimal adaptive pricing
- States: monthly sales
- Actions: setting the price for the upcoming month
- Rewards: sales



Example

- Rewards: $r_t(s,a)$ expected sales in month *t* if the previous month's sales was *s*, and the price is *a*
- Assumptions:

$r_t(s,a)$ increasing in s
super-additive?

$q_t(k|s,a)$ increasing in s
super-additive?

Infinite-horizon Markov Decision Processes with discount

Model

- Policies: $\pi = (\pi_1, \pi_2, \ldots) \in HR$

  $\pi_t : H_t \to \mathcal{P}(A)$

- Assumptions:
  - Stationary rewards and transitions: $r(s,a), \quad p(j|s,a)$
  - Bounded rewards
  - Finite or countable state space

- Discounted reward:

$$\forall \pi \in HR, \quad v_\lambda^\pi(s) = \lim_{N\to\infty} E^\pi[\sum_{t=1}^{N} \lambda^{t-1} r(X_t, Y_t)]$$

Objective

- Value function:

$$v_\lambda^\star(s) = \sup_{\pi \in HR} v_\lambda^\pi(s)$$

Optimality of MR policies

Theorem Let $\pi = (\pi_1, \pi_2, \ldots) \in HR$

For all $s \in S$, there exists $\pi' = (\pi'_1, \pi'_2, \ldots) \in MR : \forall t, \forall a$

$$P^{\pi'}[X_t = j, Y_t = a | X_1 = s] = P^{\pi}[X_t = j, Y_t = a | X_1 = s]$$

Corollary $\forall \pi \in HR, \quad \exists \pi' \in MR : \; v_\lambda^\pi(s) = v_\lambda^{\pi'}(s)$

Bellman's equations

- The value function *should* satisfy:

$$\forall s \in S, \quad v(s) = \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

- (Non-linear) operator:

$$\forall s \in S, \quad Lv(s) = \max_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

$$\forall s \in S, \quad \mathcal{L}v(s) = \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

- Bellman's equations: $\mathcal{L}v = v$

Solution to Bellman's equations

- Bellman's equations have a unique solution
- A consequence of fixed point theorem and of the following result

Theorem L and $\mathcal{L}$ are contraction mappings.

Notation

- For $d : S \rightarrow \mathcal{P}(A)$

$$r_d(s) = \sum_{a \in A} q_{d(s)}(a) r(s,a)$$

$$(P_d v)(s) = \sum_{a \in A} q_{d(s)}(a) \sum_{j \in S} p(j|s,a) v(j)$$

- For $d : S \rightarrow A$

$$r_d(s) = r(s, d(s))$$

$$(P_d v)(s) = \sum_{j \in S} p(j|s, d(s)) v(j)$$

Stationary policies

- For $\pi = (\pi_1, \pi_2, ...) \in MR$

$$v_\lambda^\pi = r_{\pi_1} + \lambda P_{\pi_1} r_{\pi_2} + ... + \lambda^{n-1} P_{\pi_1} ... P_{\pi_{n-1}} r_{\pi_n} + ...$$
$$= r_{\pi_1} + \lambda P_{\pi_1} v_\lambda^{\pi'}$$

  where $\pi' = (\pi_2, \pi_3, ...)$

- Stationary policy: $\pi = (\pi_1, \pi_1, ...)$

$$v_\lambda^\pi = r_{\pi_1} + \lambda P_{\pi_1} v_\lambda^\pi$$

The value function of a stationary policy is the unique fixed point of the linear operator $L_{\pi_1} = r_{d_1} + \lambda P_{\pi_1}$

Stationary policies

- Stationary policy: $\pi = (\pi_1, \pi_1, ...)$

$$v_\lambda^\pi = r_{\pi_1} + \lambda P_{\pi_1} v_\lambda^\pi$$

$Id - \lambda P_{\pi_1}$ invertible, and $v_\lambda^\pi = (Id - \lambda P_{\pi_1})^{-1} r_{\pi_1}$

Optimality of Bellman's equations

- Bellman's equations provide a characterization of the value function

Theorem $v^\star = v^\star_\lambda$

Summary $v_\lambda^\star(s) = \sup_{\pi \in HR} v_\lambda^\pi(s)$

- $v_\lambda^\star(s)$ is the unique solution of Bellman's equations

$$\forall s \in S, \quad v(s) = \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v(j)\}$$

- Optimal stationary policies: $\pi = (\pi_1, \pi_1, ...) \in MD$

$$\forall s \in S, \quad \pi_1(s) \in \arg\max_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_\lambda^\star(j)\}$$

- ε-optimal stationary policies: $\pi = (\pi_1, \pi_1, ...) \in MD$

$$\forall s \in S, \quad r(s, \pi_1(s)) + \lambda \sum_{j \in S} p(j|s, \pi_1(s)) v_\lambda^\star(j)$$

$$\geq \sup_{a \in A_s} \{r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_\lambda^\star(j)\} - \epsilon$$

Solving Bellman's equations

- Value iteration
- Policy iteration
- Q-learning

Value iteration

- Algorithm

1. Fix $v_0 \in V$ $(V = \{v : S \to \mathbb{R}\})$. Fix $\epsilon > 0$.
2. Do until $\|v_{n+1} - v_n\| \leq \epsilon(1-\lambda)/2\lambda$: $v_{n+1} = \mathcal{L}v_n$

$$v_{n+1}(s) = \sup_{a \in A_s} \left(r(s,a) + \sum_{j \in S} p(j|s,a)\lambda v_n(j)\right)$$

- Convergence: it does (contraction mapping)
- When it stops, we have an ε-optimal stationary policy: e.g.

$$d(s) \in \arg\max_{a \in A_s} \left(r(s,a) + \sum_{j \in S} p(j|s,a)\lambda v_n(j)\right)$$

Policy iteration

- Algorithm

1. Fix $d_0 : S \to A$. Set $n = 0$.
2. Compute the value function v_n of $\pi_n = (d_n, d_n, ...)$:

$$v_n = (Id - \lambda P_{d_n})^{-1} r_{d_n}.$$

3. Do until $d_{n+1} = d_n$: update the policy as follows:

$$\forall s, d_{n+1}(s) \in \arg\max_{a \in A_s} (r(s,a) + \sum_j p(j|s,a)\lambda v_n(j))$$

$n \to n+1$.