Today

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Optimality of monotone policies

Concepts from Last Class

Definition: A function is *superadditive* if for $\mathrm{x}^+ \geq x^-$, where $\mathrm{x}^+, x^- \in \mathrm{X}$ and $\mathrm{y}^+ \geq y^-$, where , $y^+, y^- \in \mathrm{Y}$,

$$\mathrm{g}(\mathrm{x}^+, \mathrm{y}^+) + \mathrm{g}(\mathrm{x}^-, \mathrm{y}^-) \geq g(\mathrm{x}^+, y^-) + g(x^-, y^+)$$

If the reverse inequality holds then $\mathrm{g}(x, y)$ is **subadditive**.

Definition: A Markov chain has the *IFR property* if there is an ordering of states, $S \equiv \{1,2, \ldots., n\}$, such that

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

is nondecreasing in s for all k and a.

*Read about these concepts in Section 4.7 of Puterman

Special Structured Policies

Policies with a simple structure are:

- Easier for decision makers to understand
- Easier to implement
- Easier to solve the associated MDPs

A common example is a **control limit policy**

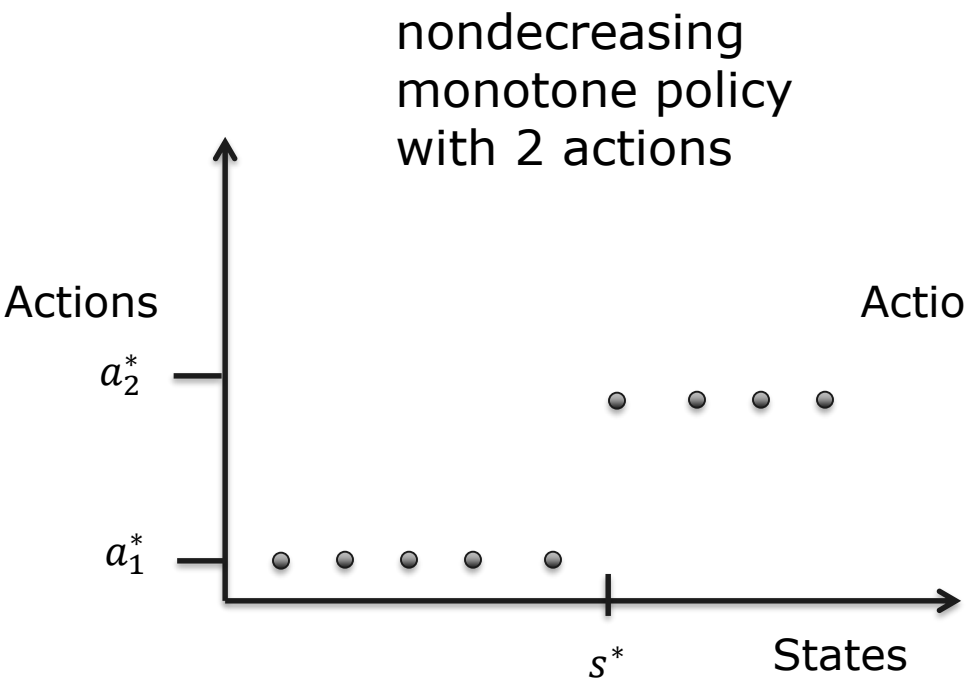
$$a_t(s_t) = \begin{cases} a_1, if\ s < s^* \\ a_2, if\ s \geq s^* \end{cases}$$

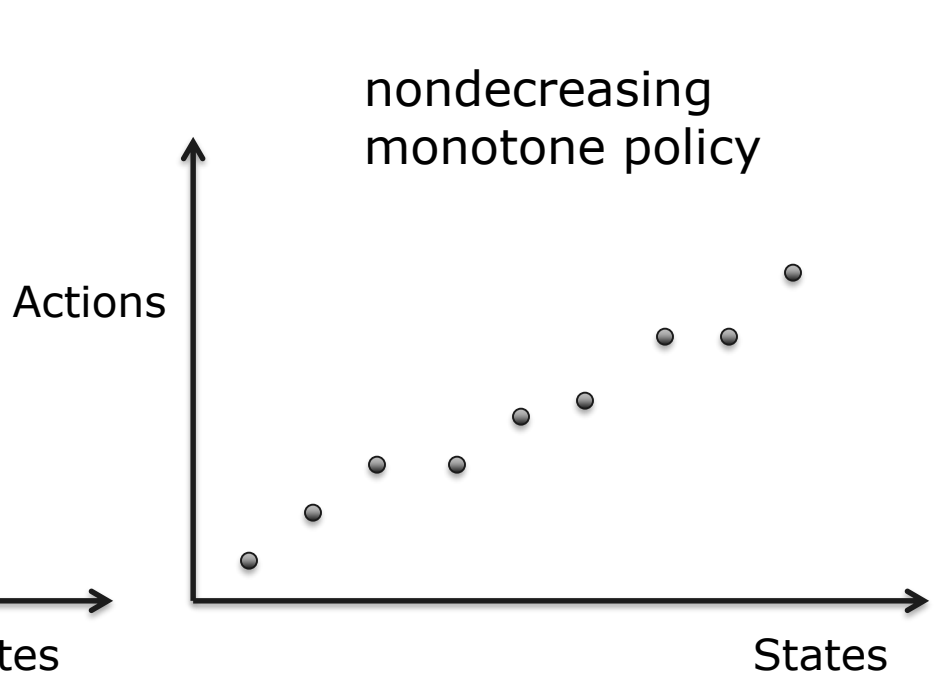
where a_1 and a_2 are alternative actions and s^* is a control limit.

Question: What conditions guarantee the existence of a control limit policy?

Examples of Monotone Policies

Definition: A policy is monotone if the *decision rule* at each stage is nonincreasing or nondecreasing with respect to the system state.





Superadditive Functions

Write the following optimality equations:

$$v_t(s) = \max_{a \in A} \{ r(s,a) + \lambda \sum_{j \in S} p(j|s,a) v_{t+1}(s) \}$$

as

$$v_t(s) = \max_{a \in A} \{ v_t(s,a) \}$$

We will prove: if $v_t(s,a)$ is **superadditive** or **subadditive** then there is an optimal policy that monotone.

Property of Superadditive Functions

The following Lemma is the key to providing sufficient conditions such that monotone policies are optimal if $v_t(s,a)$ is superadditive.

Lemma (4.7.1 Puterman): Suppose $g(x,y)$ is a superadditive function defined on $\mathrm{X} \times Y$ and for each $x \in X, \max_{y \in Y} g(x,y)$ exists. Then

$$\mathrm{f(x)} = \max\{\mathrm{y}' \in argmax_{y \in Y}\, g(x,y)\}$$

is monotone nondecreasing in x.

Proof: Completed in class. See Puterman, pp 104-105.

Definition: Increasing Failure Rate (IFR)

Many stochastic processes exhibit the following characteristic

- There is a natural ordering of states (e.g. best to worst)
- The worse the current state, the more likely it is to get worse

This property can be formalized as follows

Definition (IFR): A Markov chain has the IFR property if there is an ordering of states, $S \equiv \{1,2,\ldots,n\}$, such that

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

is nondecreasing in s for all k and a.

Property: Product of Sequences

The following Lemma is also a key property for proving conditions under which an optimal policy is monotone.

Lemma (4.7.2 Puterman): Let $\{x_j\}, \{x_j'\}$ be real valued nonnegative sequences satisfying

$$\sum_{j=k}^{\infty} x_j \geq \sum_{j=k}^{\infty} x_j'$$

For all k , with equality holding for $k = 0$. Suppose $v_{j+1} \geq v_j$ for j=0,1,…, then

$$\sum_{j=0}^{\infty} x_j v_j \geq \sum_{j=0}^{\infty} x_j' v_j.$$

provided the sums are finite.

Proof: See Puterman, pp 106.

Monotonicity: Optimal Value to Go

The following proposition provides conditions under which the optimal value function is monotone.

We need this to prove $v_t(s,a)$ is superadditive.

Proposition (4.7.3 Puterman) The optimal value to go function, $v_t(s)$, is nondecreasing (nonincreasing) in s for t=1,…, N, if the following conditions hold

1. $r_t(s,a)$ is nondecreasing (nonincreasing) in s for all $a \in A$ and $t = 1, \ldots, N-1$.
2. $q_t(k|s,a)$ is nondecreasing in s for all $k \in S, a \in A$ and $t = 1, \ldots, N$.
3. $\mathrm{R}_N(s)$ nondecreasing (nonincreasing) in s.

Proof: Completed in class. See Puterman, p 107.

Monotonicity: Optimal Policy

The following provides conditions under which the optimal policy is monotone.

Theorem (4.7.4 Puterman) Suppose for $t = 1, \ldots, N-1$

1. $r_t(s,a)$ is nondecreasing in s for all $a \in A$.

2. $q_t(k|s,a)$ is nondecreasing in s for all $k \in S, a \in A$.

3. $r_t(s,a)$ is superadditive (subadditive) on $S \times A$.

4. $q_t(k|s,a)$ is superadditive (subadditive) on $S \times A$, $\forall k$

5. $R_N(s)$ is nondecreasing in s.

Then there exist optimal decision rules, $d_t^*(s)$, which are nondecreasing (nonincreasing) in s for $t = 1, \ldots, N-1$.

Proof: See Puterman, p 107.

Another Monotonicity Result

The following provides **additional sufficient conditions** under which the optimal policy is monotone.

Theorem (4.7.5 Puterman) Suppose for $t = 1, \ldots, N-1$

1. $r_t(s,a)$ is nonincreasing in s for all $a \in A$.

2. $q_t(k|s,a)$ is nondecreasing in s for all $k \in S, a \in A$.

3. $r_t(s,a)$ is superadditive on $S \times A$.

4. $\sum_{j=0}^{\infty} p_t(j|s,a)u(j)$ is superadditive on $S \times A$, $\forall k$ for any nonincreasing $u()$

5. $R_N(s)$ is nonincreasing in s.

Then there exist optimal decision rules, $d_t^*(s)$, which are nondecreasing in s for $t = 1, \ldots, N-1$.

Proof: See Puterman, p 108.

Summary

- For certain MDPs the optimal policy can be proven to be monotone before solving the MDP
- In such cases it becomes easier to solve the problem because the set of actions that could be optimal reduces as the problem is solved:
  - **monotone backward induction** is an algorithm that takes advantage of this fact

Read section 4.7 of Puterman for a complete description of optimality of monotone policies.

Exploiting Monotonicity

To use this algorithm, states $s = 1, \ldots, M$, are ordered such that optimal actions, $a^*(s)$, are non-decreasing in s.

Complete set of possible actions

Algorithm (Monotone Backward Induction):

1. $v_N(s_N) = R_N(s_N), \quad for\ all\ S_N$, Set $\mathrm{t} = \mathrm{N} - 1$, $Set\ A_1 = A$

2. $Evaluate\ v_t(s_t)\ for\ all\ s_t = 1\ to\ M\ as$:

$$v_t(s_t) = \max_{\mathrm{a} \in A_{s_t}} \{\mathrm{r_t(s_t}, a) + \lambda \sum_{j \in S} p_t(j|s_t, a) v_{t+1}(j)\}$$

Set of optimal actions at stage t

$$A^*_{s_t} = \arg\max_{\mathrm{a} \in A_{s_t}} \{\mathrm{r_t(s_t}, a) + \lambda \sum_{j \in S} p_t(j|s_t, a) v_{t+1}(j)\}$$

$$a^*_{s_t} = \max\{a \in \mathrm{A^*_{s_t}}\}$$

New set of possible actions at state $s_t + 1$

$$A_{s_t+1} = \{a \in A | a \geq a^*_{s_t}\}$$

3. If $\mathrm{t} = 1$ then stop; otherwise $\mathrm{t} = \mathrm{t} - 1$, and return to step 2.

Example: Maintenance Problem

A tool **deteriorates** stochastically with states $S = \{0,1,2,\ldots\}$.The decision maker chooses from actions $\mathrm{A} = \{\mathrm{D}, \mathrm{R}\}$, to replace the tool (R) or defer replacement (D). The tool deteriorates by i states with probability $\mathrm{p(i)}$. **Action R has the potential to return the tool to the ideal state 0**.

Transition probabilities:

$$\mathrm{p_t}(j|s,D) = \begin{cases} 0, & j < s \\ p(j-s), & j \geq s \end{cases}$$

and $\mathrm{p_t}(j|s,R) = p(j), j \geq 0$. Rewards:

$$\mathrm{r_t}(s,a) = \begin{cases} W - h(s), & a = D \\ W - K, & a = R \end{cases}$$

Where W is a fixed reward for each epoch, K is a fixed cost of replacement, and h(s) is a state dependent nonnegative maintenance cost. The salvage value at the end of period N is $\mathrm{R_N(s)}$

Note: Theorem 4.7.5 can be used to prove there is always an optimal monotone policy for this problem (see Canvas).

In Class Assignment

Assume the manufacturing tool has states $S = \{G, B, F\}$, i.e., good (G), bad (B), failed (F) .The decision maker chooses from actions $A = \{D, R\}$, to replace the tool (R) or defer replacement (D). There are three stages t=1,2,3.

Transition probability matrix:

$$P = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0 & 0.4 & 0.6 \\ 0 & 0 & 1.0 \end{bmatrix}$$

and repair returns the state to G w.p. 1.

Rewards:

$$r_t(s, a) = \begin{cases} 1 - h(s), & a = D \\ 0, & a = R \end{cases}$$

Where W=1, K=1, $h = (0,1,2)$ and the salvage is $R_3 = (2, 1, 0)$.

Exercise: Given it can be proven an optimal monotone policy exists. Use monotone backward induction to find the optimal policy.