Dynamic Programming Lecture #7

Outline:

- Stochastic DP algorithm
- Simple example
- Repeated prisoner's dilemma
- LQ optimal control

Stochastic DP

- System:

$$x_{k+1} = f_k(x_k, u_k, w_k)$$
$$x_k \in S_k, \quad u_k \in U_k(x_k), \quad w_k \in W_k(x_k, u_k)$$

- Assume:
  - w_k is an RV on some probability space Ω_k
  - Probability function $p(w_k)$ can depend on x_k & u_k.
  - Probability function CANNOT depend on $w_0, \ldots, w_{k-1}$.
  - More precisely:

$$p_{W_k}(w_k | x_k, u_k) = p_{W_k}(w_k | x_0, ..., x_k, u_0, ..., u_k, w_0, ..., w_{k-1})$$

- Objective:

$$J^*(x_0) = \min_{\mu_0,\ldots,\mu_{N-1}} E_{w_0,\ldots,w_{N-1}} \left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\}$$

- Interpretation:
  - Total probability space $\Omega = \Omega_0 \times \ldots \times \Omega_{N-1}$.
  - Given admissible policy, value between $\{\cdot\}$ is an RV on Ω.
  - Can enumerate possibilities & probabilities to compute expected value.

Stochastic DP Algorithm

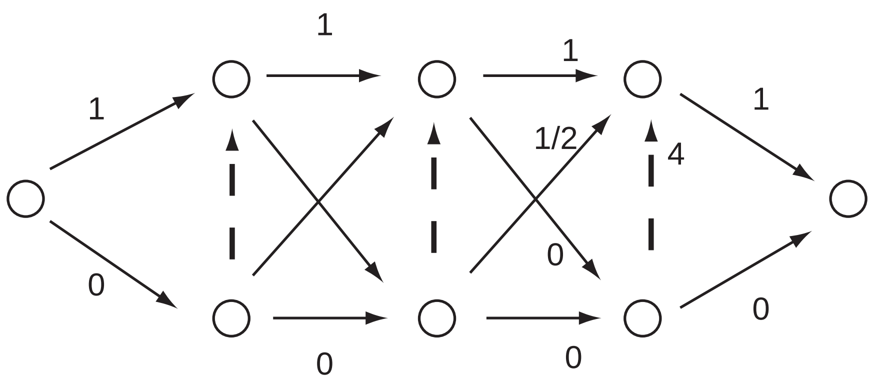
- Define

$$J_N(x_N) = g_N(x_N)$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}$$

- Theorem:
  - $J_0(x_0) = J^*(x_0)$
  - $\mu_k(x_k) = \arg\min_{u_k \in U_k(x_k)} E_{w_k} \left\{ g_k(x_k, u_k, w_k) + J_{k+1}(f_k(x_k, u_k, w_k)) \right\}$
  - $J_k(x_k) =$ optimal cost-to-go (i.e., solution to subproblem).
- Proof by induction to show $J_k(x_k) =$ optimal cost-to-go
  - Assume true for $J_{k+1}(\cdot)$.
  - Show true for $J_k(\cdot)$.
  - Start induction with $J_N(\cdot)$.
- Details of proof: Later...

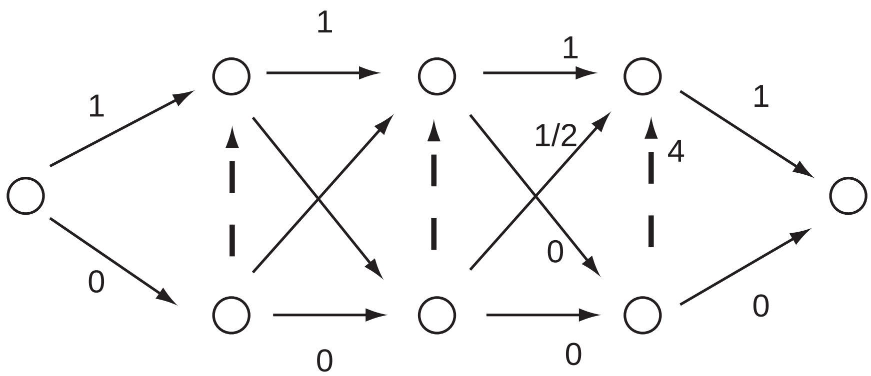
Example



- Two states per stage: $\{1, 2\}$.
- High road (=1) costly...low road (=2) cheap.
- On low road, can get a costly "bump" to high road with probability p:

$$x^+ = u + w, \quad u \in \{1, 2\}, \quad W(u = 1) = \{0\}, \quad W(u = 2) = \{0, -1\}$$

Example, cont (2)



- Apply DP with $p = 1/4$:

$$J_3(1) = 1, \quad J_3(2) = 0$$

$$J_2(1) = \min \begin{cases} 1 + J_3(1) \\ (0 + 4 + J_3(1))p + (0 + J_3(2))(1-p) \end{cases} = \min \begin{cases} 1 + 1 \\ (0 + 4 + 1)(1/4) + (0 + 0)(3/4) \end{cases} = 5/4(\text{low})$$

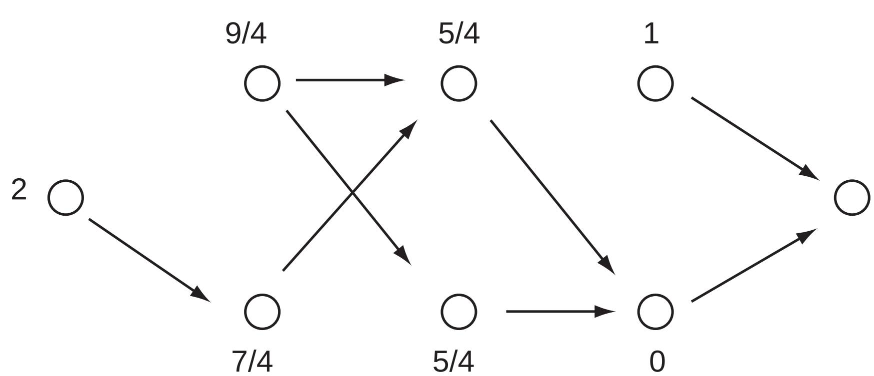
$$J_2(2) = \min \begin{cases} 1/2 + J_3(1) \\ (0 + 4 + J_3(1))p + (0 + J_3(2))(1-p) \end{cases} = \begin{cases} 1/2 + 1 \\ (0 + 4 + 1)(1/4) + (0 + 0)(3/4) \end{cases} = 5/4(\text{low})$$

$$J_1(1) = \min \begin{cases} 1 + J_2(1) \\ (0 + 4 + J_2(1))p + (0 + J_2(2))(1-p) \end{cases} = \begin{cases} 1 + 5/4 \\ (0 + 4 + 5/4)(1/4) + (0 + 5/4)(3/4) \end{cases} = 9/4(\text{high or low}$$

$$J_1(2) = \min \begin{cases} 1/2 + J_2(1) \\ (0 + 4 + J_2(1))(1/4) + (0 + J_2(2))(3/4) \end{cases} = 7/4(\text{high})$$

$$J_0 = \min \begin{cases} 1 + J_1(1) \\ (0 + 4 + J_1(1))(1/4) + (0 + J_1(2))(3/4) \end{cases} = 2(\text{low})$$

- Result: Map of cost-to-go AND optimal decision.



Repeated Prisoner's Dilemma

	C	D
C	$4,4$	$0,5$
D	$5,0$	$1,1$

- Row = "us", Column = "them"
- Reward = (us, them)
- "Dilemma": Defected is a "dominating" strategy for both players.
- Repeated PD: Play game over stages $[0, 1, ..., N-1]$.
- Define:
  - u_k = Row's action at stage k
  - w_k = Column's action at stage k
- Opponent models:
  - Tit-for-Tat:

$$w_k = \begin{cases} u_{k-1}, & \text{with probability } p; \\ D, & \text{with probability } 1-p. \end{cases}$$

  - Grim trigger:

$$w_k = \begin{cases} C, & \text{with probability } p \text{ if } u_0, ..., u_{k-1} = C; \\ D, & \text{with probability } 1-p \text{ if } u_0, ..., u_{k-1} = C; \\ D, & \text{if } u_j = D \text{ for any } j < k. \end{cases}$$

  Typically, $p = 1$.

DP for PD: Tit-for-Tat

- Set $M = \begin{pmatrix} 4 & 0 \\ 5 & 1 \end{pmatrix}$.
- Dynamics and costs:

$$x_{k+1} = u_k$$

$$g_k(x_k, u_k, w_k) = M_{u_k w_k}$$

$$g_N(x_N) = 0$$

- Stage $N-1$, $x_{N-1} = C$:

$$J_{N-1}(C) = \max \begin{cases} 4p + 0 \cdot (1-p) + J_N(C), & u_{N-1} = C; \\ 5p + 1 \cdot (1-p) + J_N(D), & u_{N-1} = D. \end{cases}$$

$$\Rightarrow$$

$$J_{N-1}(C) = 4p + 1 \quad \& \quad \mu^*_{N-1}(C) = D$$

- Stage $N-1$, $x_{N-1} = D$:

$$J_{N-1}(D) = \max \begin{cases} 0 + J_N(C), & u_{N-1} = C; \\ 1 + J_N(D), & u_{N-1} = D. \end{cases}$$

$$\Rightarrow$$

$$J_{N-1}(D) = 1 \quad \& \quad \mu^*_{N-1}(D) = D$$

DP for PD: Tit-for Tat, cont.

- Stage $N-2$, $x_{N-2} = C$:

$$J_{N-2}(C) = \max \begin{cases} 4p + 0 \cdot (1-p) + (4p+1), & u_{N-2} = C; \\ 5p + 1 \cdot (1-p) + 1, & u_{N-2} = D. \end{cases}$$

Accordingly, if

$$4p + 4p + 1 > 4p + 1 + 1 \Leftrightarrow p > 1/4$$

then

$$J_{N-2}(C) = 8p + 1 \quad \& \quad \mu^*_{N-2}(C) = C$$

- Stage $N-2$, $x_{N-2} = D$:

$$J_{N-2}(D) = \max \begin{cases} 0 + (4p+1), & u_{N-2} = C; \\ 1 + 1, & u_{N-2} = D. \end{cases}$$

Accordingly, if

$$4p + 1 > 1 + 1 \Leftrightarrow p > 1/4$$

then

$$J_{N-2}(D) = 4p + 1 \quad \& \quad \mu^*_{N-2}(D) = C$$

DP for PD: Grim trigger

- As before, for $p > 1/4$:

$$J_{N-1}(C) = 4p + 1 \quad \& \quad \mu^*_{N-1}(C) = D$$

$$J_{N-1}(D) = 1 \quad \& \quad \mu^*_{N-1}(D) = D$$

$$J_{N-2}(C) = 8p + 1 \quad \& \quad \mu^*_{N-2}(C) = C$$

- Not as before:

$$J_{N-2}(D) = \max \begin{cases} 0 + J_{N-1}(D), & u_{N-2} = C; \\ 1 + J_{N-1}(D), & u_{N-2} = D. \end{cases}$$

$$\Rightarrow$$

$$J_{N-2}(D) = 2 \quad \& \quad \mu^*_{N-2}(D) = D$$

- In fact, $\mu^*_k(D) = D$
- Next question: What is optimal control versus μ^*?

LQ Optimal Control

- Linear system (time-invariant):

$$x^+ = Ax + Bu + w, \quad E\{w\} = 0$$

  - x : state
  - u : control
  - w : "process" disturbance

- Quadratic cost:

$$\min_{\mu_0,\ldots,\mu_N} E\left\{x_N^T Q_N x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T u_k\right\}, \quad Q_N \geq 0$$

- Assumptions: $Q = Q^T > 0$, $Q_N = Q_N^T \geq 0$
- Recall: $Q > 0$:

$$x^T Q x > 0, \quad \text{for all} x \neq 0$$

- Interpretation: Want to minimize "energy" of state while not expending excessive energy of control, where

$$\mathcal{E}[f] = \sum_k f_k^T Q f_k$$

  Compare to:

$$\int i^2 R \quad \text{or} \quad \int cv^2$$

- Applications: Flutter control, vibration suppression, control law generation
- Q scales relative importance of terms & state/control energy tradeoff
- Q_N penalize size of terminal state

LQ Optimal Control, cont

- $N-1$ recursion:

$$J_N(x_N) = x_N^T Q_N x_N$$

$$\begin{aligned} & J_{N-1}(x_{N-1}) \\ = \;& \min_{u_{N-1}} E_{w_{N-1}} \left\{ x_{N-1}^T Q x_{N-1} + u_{N-1}^T u_{N-1} + J_N(Ax_{N-1} + Bu_{N-1} + w_{N-1}) \right\} \\ = \;& \min_{u_{N-1}} E \left\{ x_{N-1}^T Q x_{N-1} + u_{N-1}^T u_{N-1} + (Ax_{N-1} + Bu_{N-1} + w_{N-1})^T Q_N (Ax_{N-1} + Bu_{N-1} + w_{N-1}) \right\} \\ = \;& \min_{u_{N-1}} x^T x\text{-terms} + u^T u\text{-terms} + x^T u\text{-terms} + E \left\{ w_{N-1}^T Q_N w_{N-1} \right\} \end{aligned}$$

Take $\frac{\partial}{\partial u_{N-1}}$:

$$u_{N-1} = -(I + B^T Q_N B)^{-1} B^T Q_N A x_{N-1}$$

and substitute to produce (quadratic!)

$$J_{N-1}(x_{N-1}) = x_{N-1}^T P_{N-1} x_{N-1} + E \left\{ w_{N-1}^T Q_N w_{N-1} \right\}$$

where

$$P_{N-1} = Q + A^T Q_N A - A^T Q_N B (I + B^T Q_N B)^{-1} B^T Q_N A$$

LQ Optimal Control, cont

- $N-2$ recursion: Same analysis, but Q_N replaced by P_{N-1}.
- k^{th} recursion:

$$u_k = -(I + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x_k$$
$$P_{k-1} = Q + A^T P_k A - A^T P_k (I + B^T P_k B)^{-1} B^T P_k A, \quad P_N = Q_N$$
$$J_0(x_0) = x_0^T P_0 x_0 + \sum_{k=0}^{N-1} E\left\{ w_k^T P_{k+1} w_k \right\}$$

- $P_k \geq 0$ by definition of positive cost.
- Comments:
  - Indicative of DP: Find a recurring structure and exploit.
  - DP leads to map of cost-to-go and optimal decision.
  - Could have derived case where A & B vary with k ... today's optimal action depends on tomorrow's model.