4- Dynamic Programming

Melih Kandemir

University of Southern Denmark
Department of Mathematics and Computer Science (IMADA)
kandemir@imada.sdu.dk

Fall 2022

Dynamic programming (DP)

- **Dynamic:** sequential (temporal)
- **Programming** optimizing a program
  (a sequence of operation steps)

DP is applicable to problems that consist of

- optimal substructure
  - there exists a notion of optimality that can be proven
  - optimal solution can be decomposed into subproblems
- overlapping subproblems
  - subproblems recur many times
  - solutions can be cached and reused

DP suits perfectly for solving MDPs

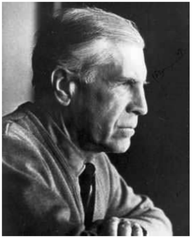
- **optimal substructure:** Bellman equation decomposes recursively (optimality principle yet to come!)
- **overlapping subproblems:** Value function stores and reuses solutions

The Newton-Leibniz duality of RL

Richard Bellman

State s_t
Action a_t
Reward r_t
Value $V(s_t)$
HJB Equation
Taylor expansion
Sutton

Lev Pontryagin



State x_t
Control u_t
Cost $g(i, u, j)$
Cost-to-go $J(x_t)$
Minimum principle
Calculus of variations
Bertsekas

Terminology (Bertsekas view)

- **State set:** $S = \{0, 1, \dots, n\}$, where 0 is the terminal state if exists
- **Policy:** A sequence $\pi = \{\mu_0, \mu_1, \dots\}$ such that $\mu_k(i) \in U(i), \forall i \in S$, where $U(i)$ is the set of control actions
- **Transition probabilities:**

$$P(i_{k+1} = j | i_k = i) = p_{ij}(\mu_k(i))$$

- **Expected cost** of a finite-horizon (episodic) problem:

$$J_N^\pi(i) = \mathbb{E}\left[\alpha^N G(i_N) + \sum_{k=0}^{N-1} \alpha^k g(i_k, \mu_k(i_k), i_{k+1}) \middle| i_0 = i\right]$$

where $g(i, u, j)$ is cost, $G(i_N)$ terminal cost, and $\alpha_k \in (0, 1]$ is a discount factor and expectation is wrt the Markov chain $\{i_0, i_1, \dots, i_N\} \sim \prod_{k=0}^{N-1} p_{i_k i_{k+1}}(\mu_k(i))$.

Cost-to-go vectors

- Optimal $N-$stage cost-to-go:

$$J_N^*(i) = \min_\pi J_N^\pi(i)$$

  and in vector form $J_N^* = (J_N^\pi(1), \ldots, J_N^\pi(n))$

- Infinite horizon problem

$$J^\pi(i) = \lim_{N \to \infty} \mathbb{E}\left[\sum_{k=0}^{N-1} \alpha^k g(i_k, \mu_k(i_k), i_{k+1}) \middle| i_0 = i \right]$$

  for which optimal cost-to-go vector is J^*.

- Stationary policy: $\pi = \{\mu, \mu, \ldots\}$ and its cost-to-go J^μ.

Dynamic programming

One-stage case

$$J_1^*(i) = \min_{\mu_0} \sum_{j=1}^{n} p_{ij}(\mu_0(i))(g(i, \mu_0(i), j) + \alpha G(j))$$

$$= \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(\mu_0(i))(g(i, u, j) + \alpha G(j))$$

Let us take a leap of faith and generalize to

$$J_k^*(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(\mu_0(i))(g(i, u, j) + \alpha J_{k-1}^*(j))$$

then starting with $J_0^*(i) = G(i)$ and solving recursively

$$J_0^* \rightarrow J_1^* \rightarrow \ldots J_k^*.$$

This is a **dynamic programming** algorithm.

Proof that DP will work for RL

Express $\mu_k = \{u, \mu_{k-1}\}, u \in U(i)$ and do

$$\begin{aligned} J_k^*(i) &= \min_{u \in U(i), \pi_{k-1}} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \alpha J_{k-1}^{\pi_{k-1}}(j)) \\ &= \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \alpha \min_{\pi_{k-1}} J_{k-1}^{\pi_{k-1}}(j)) \\ &= \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \alpha J_{k-1}^*(j)) \end{aligned}$$

■

This identity is known as the **Principle of Optimality.**

General theory

Definition

Stationary μ is **proper** if

$$\rho_\mu = \max_{i=1,\dots,n} \mathbb{P}(i_n \neq 0 | i_0 = i, \mu) < 1$$

and **improper** otherwise.

Two key assumptions:

i) There exists at least one proper μ

ii) For every improper μ, $\exists i$ s.t. $J^\mu(i) \to \infty$.

The Bellman backup operator

- One iteration of DP:

$$(TJ)(i) = \min_{u \in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + J(j))$$

where we assume $J(0) = 0$. This is the optimal cost-to-go for one-stage cost g and terminal cost J.

- Also define:

$$(T_\mu J)(i) = \sum_{j=0}^{n} p_{ij}(\mu(i))(g(i,\mu(i),j) + J(j))$$

where we assume $J(0) = 0$. This is the cost-to-go for policy μ one-stage cost g and terminal cost J.

The Bellman backup operator

- Define $n \times n$ matrix P_μ with ijth entry $p_{ij}(\mu(i))$. Then

$$T_\mu J = g_\mu + P_\mu J,$$

where $g_\mu(i) = \sum_{j=0}^{n} p_{ij}(\mu(i)) g(i, \mu(i), j)$.
- Denote k iteration DP algorithm as

$$(T^k J)(i) = (T(T^{k-1} J))(i),$$
$$(T_\mu^k J)(i) = (T_\mu(T_\mu^{k-1} J))(i)$$

with $(T^0 J)(i) = J(i)$ and $(T_\mu^0 J)(i) = J(i)$.

Preliminaries

Monotonicity lemma

For any k, stationary μ

$$\begin{aligned} J(i) \leq \bar{J}(i) &\Rightarrow (T^k J)(i) \leq (T^k \bar{J})(i) \\ &\Rightarrow (T_\mu^k J)(i) \leq (T_\mu^k \bar{J})(i) \end{aligned}$$

Constant offset lemma

For any k, J, stationary μ and $r \in \mathbb{R}_+$

$$(T^k(J + re))(i) \leq (T^k J)(i) + r,$$
$$(T_\mu^k(J + re))(i) \leq (T_\mu^k J)(i) + r.$$

where e is a vector of ones. Reverse inequalities if $r < 0$.

Main results

Proposition

Assume the two assumptions above hold. Then

(a) $J = TJ \iff J = J^*$.

(b) $\lim_{k\to\infty} T^k J = J^*, \ \forall J$.

(c) Stationary μ is optimal $\iff T_\mu J^* = TJ^*$.

(d) For every proper μ and every J

$$\lim_{k\to\infty} T_\mu^k J = J^\mu,$$

$$J^\mu = T_\mu J^\mu$$

and J^μ is the unique solution of this equation.

Value iteration

Synchronous update

repeat
    $J' := TJ$
until $J' = J$

- Equivalently $T^k J$ as $k = 1, 2, \ldots$.
- Requires infinite iterations to converge.
- Converges in $O(n)$ if P^{μ^*} is acyclic, i.e. edge (i, j) exists if $i \neq 0$ and $p_{ij}(\mu^*(i)) > 0$ and initialized $J(i) = \infty, i \neq 0$.
- Converges to J^*.

Asynchronous update (Gauss-Seidel method)

$$(FJ)(i) = \min_{u \in U(i)} \Big[\sum_{j=0}^{n} p_{ij}(u) g(i, u, j) + \sum_{j=1}^{i-1} p_{ij}(u)(FJ)(j) + \sum_{j=i}^{n} p_{ij}(u) J(j) \Big]$$

Policy iteration

repeat
  $J_{\mu_k} := (I - P_{\mu_k})^{-1} g_{\mu_k}$ ▷ Policy evaluation
  $T_{\mu_{k+1}} J^{\mu_k} := T J^{\mu_k}$ ▷ Policy improvement
until $J^{\mu_{k+1}} = J^{\mu_k}$

- Policy improvement step in more detail

$$\mu_{k+1}(i) = \arg \min_{u \in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + J^{\mu_k}(j))$$

- Converges in finite iterations.

Policy improvement theorem

Proposition.

The policy iteration algorithm generates an improving sequence of proper policies $\mu_1, \mu_2, \ldots$, i.e.

$$J^{\mu_{k+1}} \leq J^{\mu_k}, \forall k = 1, 2, \ldots$$

and terminates at J^*.

Proof.

Given a proper μ, we get $J^\mu = T_\mu J^\mu \geq T_{\bar{\mu}} J^\mu = T J^\mu$. Due to monotonicity lemma, $J^\mu \geq T_{\bar{\mu}}^k J^\mu$ holds also for $k = 1, 2, \ldots$. Now assume $\bar{\mu}$ is not proper, $T_{\bar{\mu}}^k J^\mu \to \infty$, which contradicts monotonicity lemma. Hence $\bar{\mu}$ is proper. From main result (d), we have $\lim_{k\to\infty} T_{\bar{\mu}}^k J^{\bar{\mu}} = J^{\bar{\mu}}$. If μ is nonoptimal, $J^{\bar{\mu}}(i) < J^\mu(i)$ for some i. Otherwise $J^\mu = T J^\mu \Rightarrow J^\mu = J^* \Rightarrow \mu = \mu^*$. Hence each step either improves or equilibrium is found with optimal policy. As the number of policies is finite, the sequence terminates. ■

Multistage lookahead policy iteration

repeat
$J_{\mu_k} := (I - P_{\mu_k})^{-1} g_{\mu_k}$ ▷ Policy evaluation
$T_{\mu_{k+1}} T^{m-1} J^{\mu_k} := T^m J^{\mu_k}$ ▷ Policy improvement
until $J^{\mu_{k+1}} = J^{\mu_k}$

- **Core idea:** Plan for long horizon to determine the immediate action.
- **Important observation:** $J^{\mu_{k+1}} = T_{\mu_{k+1}}^{l \to \infty} J^{\mu} \leq T^m J^{\mu_k} \leq J^{\mu_k}$
- $T^m J^{\mu_k}$ approaches $J^{\mu_{k+1}}$ as m increases, hence choose maximum m the computation budget allows.
- The tightness of the bound will be decisive for approximate cost-to-go functions.
- Since $J^{\mu_{k+1}} \leq J^{\mu_k}$, all convergence properties of the single-stage version are inherited.

Policy iteration as actor-critic

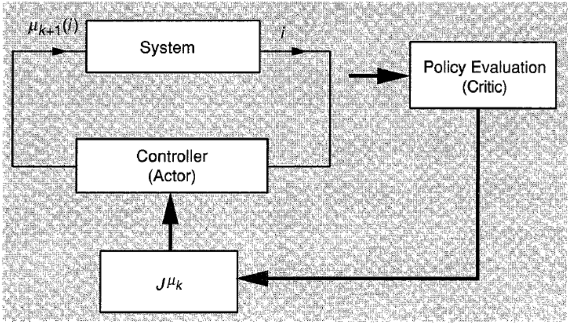


Figure: Image from Bertsekas, Neuro-dynamic programming

Discounted problems

The new operators

$$(TJ)(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \alpha J(j))$$

$$(T_\mu J)(i) = \sum_{j=1}^{n} p_{ij}(\mu(i))(g(i,u,j) + \alpha J(j))$$

Discounted problems

The corresponding lemmas.

Monotonicity lemma

For any k, stationary μ

$$J(i) \leq \bar{J}(i) \Rightarrow (T^k J)(i) \leq (T^k \bar{J})(i)$$
$$\Rightarrow (T_\mu^k J)(i) \leq (T_\mu^k \bar{J})(i)$$

Constant offset Lemma

For any k, J, stationary μ and $r \in \mathbb{R}_+$

$$(T^k(J + re))(i) = (T^k J)(i) + \alpha^k r,$$
$$(T_\mu^k(J + re))(i) = (T_\mu^k J)(i) + \alpha^k r.$$

where e is a vector of ones.

Bellman backup operator is contraction

Define maximum norm as $||J||_\infty = \max_i |J(i)|$.

Lemma (Contraction)

$\forall J, \bar{J}$ and μ:

$$||TJ - T\bar{J}||_\infty \leq \alpha ||J - \bar{J}||_\infty,$$
$$||T_\mu J - T_\mu \bar{J}||_\infty \leq \alpha ||J - \bar{J}||_\infty.$$

Proof

Denote $c = \max_{i=1,\ldots,n} |J(i) - \bar{J}(i)|$. Then

$$\begin{aligned} J(i) - c \leq \bar{J}(i) \leq J(i) + c, \qquad & i = 1, \ldots, n \\ \Rightarrow (TJ)(i) - \alpha c \leq (T\bar{J})(i) \leq (TJ)(i) + \alpha c & \\ \Rightarrow |(TJ)(i) - (T\bar{J})(i)| \leq \alpha c. & \end{aligned}$$

Second inequality follows by choosing $\mu(i)$ as the only available control at state i ■

Temporal difference based policy iteration

- Value iteration converges too slowly, especially when $\alpha \approx 1$
- Policy evaluation does not scale to large state spaces
- Best of both worlds is possible if an equivalent problem can be defined with reduced discount factor.
- This is possible only if the expectation of the one-stage cost is zero.

λ-policy iteration method

Maintain a sequence (J_k, μ_k), treat $J_k \approx J^{\mu_k}$, and do

i) $T_{\mu_{k+1}} J_k = T J_k$

ii) Calculate

$$d_k(i,j) = g(i, \mu_{k+1}(i), j) + \alpha J_k(j) - J_k(i)$$

as the one-stage cost of μ_{k+1} for an $\alpha\lambda$ discounted DP with $p_{ij}(\mu_{k+1})$. The cost-to-go is then

$$\Delta_k(i) = \sum_{m=0}^{\infty} \mathbb{E}[(\alpha\lambda)^m d_k(i_m, i_{m+1}) | i_0 = i], \qquad \forall i.$$

iii) $J_{k+1} = J_k + \Delta_k$

The policy-value iteration continuum

When $\lambda = 1$, we have

$$\begin{aligned}\Delta_k(i) &= \sum_{m=0}^{\infty} \mathbb{E}[\alpha^m d_k(i_m, i_{m+1}) | i_0 = i] \\ &= \sum_{m=0}^{\infty} \mathbb{E}[\alpha^m g(i_m, \mu_{k+1}(i_m), i_{m+1}) \\ &\qquad + \alpha^{m+1} J_k(i_{m+1}) - \alpha^m J_k(i_m) | i_0 = i] \\ &= \sum_{m=0}^{\infty} \mathbb{E}[\alpha^m g(i_m, \mu_{k+1}(i_m), i_{m+1}) | i_0 = i] - J_k(i) \\ &= J^{\mu_{k+1}}(i) - J_k(i) \Rightarrow J_{k+1} = J^{\mu_{k+1}} \Rightarrow \textbf{Policy iteration!}\end{aligned}$$

The policy-value iteration continuum

When $\lambda = 0$, we have

$$\begin{aligned} J_{k+1}(i) &= J_k(i) + \mathbb{E}[d_k(i_1, i_0)|i_0 = i] \\ &= J_k(i) + \mathbb{E}[g(i_0, \mu_{k+1}(i_0), i_1) + \alpha J_k(i_1) - J_k(i_0)|i_0 = i] \\ &= J_k(i) + \mathbb{E}[g(i_0, \mu_{k+1}(i_0), i_1) + \alpha J_k(i_1)|i_0 = i] - J_k(i_0) \\ &= \mathbb{E}[g(i_0, \mu_{k+1}(i_0), i_1) + \alpha J_k(i_1)|i_0 = i] \\ &\Rightarrow J_{k+1} = T_{\mu_{k+1}} J_k = T J_k \Rightarrow \textbf{Value iteration!} \end{aligned}$$

Theoretical properties of λ-policy iteration

Theorem [Contraction]

Consider $M_k J = (1-\lambda) T_{\mu_{k+1}} J_k + \lambda T_{\mu_{k+1}} J$ with $T_{\mu_{k+1}}$ satisfying $||T_{\mu_{k+1}} J - T_{\mu_{k+1}} \bar{J}|| \leq \beta ||J - \bar{J}||$ for $\beta < 1$ and any $(J, \bar{J})$, then

i) $||M_k J - M_k \bar{J}|| \leq \beta\lambda ||J - \bar{J}||$

ii) $M_k^m J = (1-\lambda) \Big[\sum_{i=0}^{m-1} \lambda^i T_{\mu_{k+1}}^{i+1} J_k \Big] + \lambda^m T_{\mu_{k+1}}^m J, \qquad \forall m \geq 1$

iii) J_{k+1} is the unique fixed point of M_k, i.e. $J = M_k J$, and $J_{k+1} = (1-\lambda) \sum_{m=0}^{\infty} \lambda^m T_{\mu_{k+1}}^{m+1} J_k$ holds.

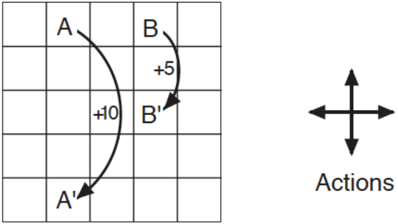
Theorem [Rate of convergence]

i) If $\alpha < 1$, then $J_k \to J^*$. Furthermore $\exists\ \bar{k}$ such that $\forall k > \bar{k}$

$$||J_{k+1} - J^*|| \leq \frac{\alpha(1-\lambda)}{1-\alpha\lambda} ||J_k - J^*||$$

ii) If $\alpha = 1$, μ proper, and $TJ_0 \leq J_0$, then $J_k \to J^*$.

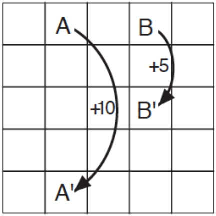
Gridworld



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Figure. Sutton and Barto, MIT Press, 2017. (Right) value function of a random policy.

Gridworld optimal solution



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

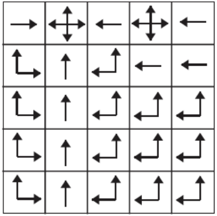
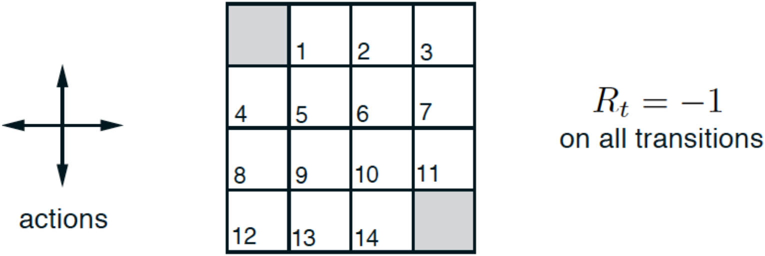


Figure. Sutton and Barto, MIT Press, 2017. (Middle:) Value function of the optimal policy. (Right:) Optimal policy.

Solving 4×4 GridWorld with policy improvement



Solving 4×4 GridWorld with policy improvement

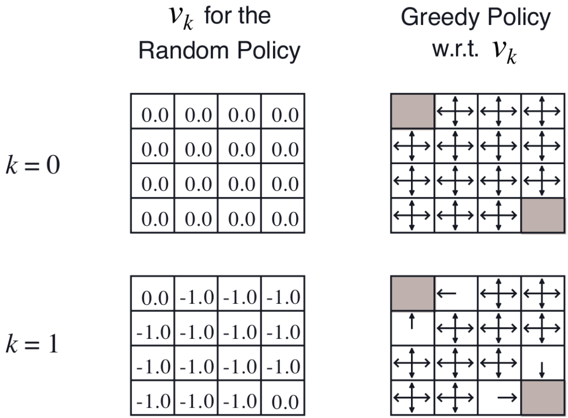


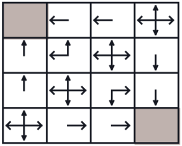
Figure. Sutton and Barto, MIT Press, 2017

Solving 4×4 GridWorld with policy improvement

All policies are optimal from $K = 3$ on.

$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0



$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

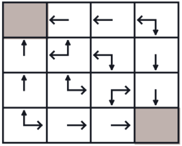


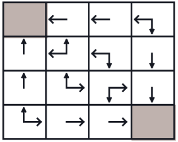
Figure. Sutton and Barto, MIT Press, 2017

Solving 4×4 GridWorld with policy improvement

All policies are optimal from $K = 3$ on.

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0



$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

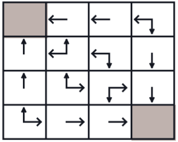


Figure. Sutton and Barto, MIT Press, 2017