Reinforcement Learning Algorithms in Markov Decision Processes
AAAI-10 Tutorial

Introduction

Csaba Szepesvári   Richard S. Sutton

University of Alberta
E-mails: {**szepesva,rsutton**}**@.ualberta.ca**

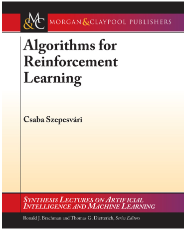
Atlanta, July 11, 2010

Outline

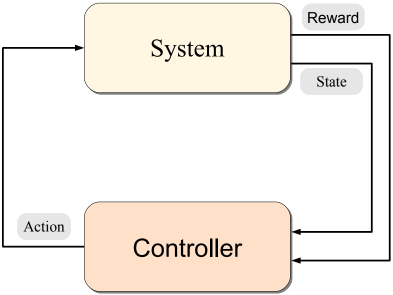
Presenters

Richard S. Sutton is a professor and iCORE chair in the Department of Computing Science at the University of Alberta. He is a fellow of the AAAI and co-author of the textbook Reinforcement Learning: An Introduction from MIT Press. His research interests center on the learning problems facing a decision-maker interacting with its environment, which he sees as central to artificial intelligence.

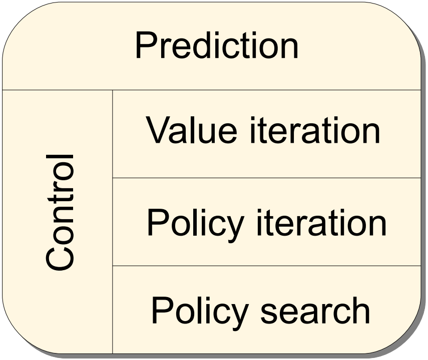
Csaba Szepesvári, an Associate Professor at the Department of Computing Science of the University of Alberta, is the coauthor of a book on nonlinear approximate adaptive controllers and the author of a recent book on reinforcement learning. His main interest is the design and analysis of efficient learning algorithms in various active and passive learning scenarios.



Reinforcement learning



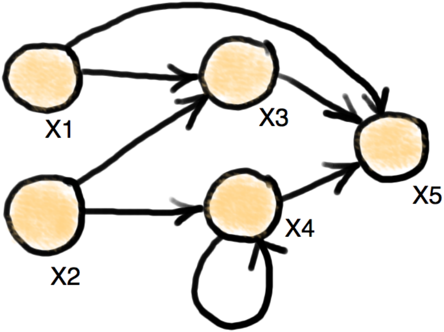
Preview of coming attractions



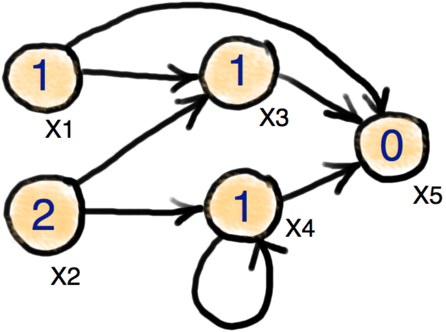
The structure of the tutorial

- Markov decision processes
  - Generalizes shortest path computations
  - Stochasticity, state, action, reward, value functions, policies
  - Bellman (optimality) equations, operators, fixed-points
  - Value iteration, policy iteration
- Value prediction
  - Temporal difference learning unifies Monte-Carlo and bootstrapping
  - Function approximation to deal with large spaces
  - New gradient based methods
  - Least-squares methods
- Control
  - Closed-loop interactive learning: exploration vs. exploitation
  - Q-learning
  - SARSA
  - Policy gradient, natural actor-critic

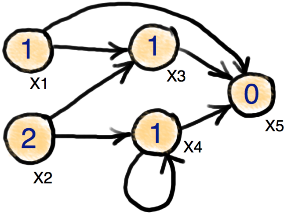
How to get to Atlanta?



How to get to Atlanta?



Value iteration



function VALUEITERATION(x^*)

for $x \in \mathcal{X}$ **do** $V[x] \leftarrow 0$

$V' \leftarrow V$

repeat

  for $x \in \mathcal{X} \setminus \{x^*\}$ **do**

    $V[x] \leftarrow 1 + \min_{y \in \mathcal{N}(x)} V(y)$

  end for

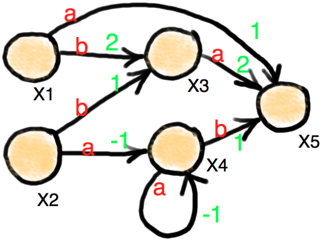
until $V \neq V'$

return V

function BESTNEXTNODE(x, V)

return $\arg\min_{y \in \mathcal{N}(x)} V(y)$

Rewarding excursions



function ValueIteration

for $x \in \mathcal{X}$ **do** $V[x] \leftarrow 0$

$V' \leftarrow V$

repeat

  for $x \in \mathcal{X} \setminus \{x^*\}$ **do**

    $V[x] \leftarrow \max_{a \in \mathcal{A}(x)} \{ r(x,a) + \gamma V(f(x,a)) \}$

  end for

until $V \neq V'$

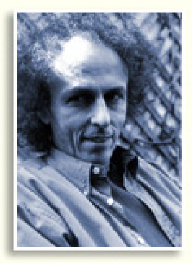
return V

function BestAction(x, V)

return $\operatorname{argmax}_{a \in \mathcal{A}(x)} \{ r(x,a) + \gamma V(f(x,a)) \}$

Uncertainty

"Uncertainty is the only certainty there is, and knowing how to live with insecurity is the only security." (John Allen Paulos, 1945–)

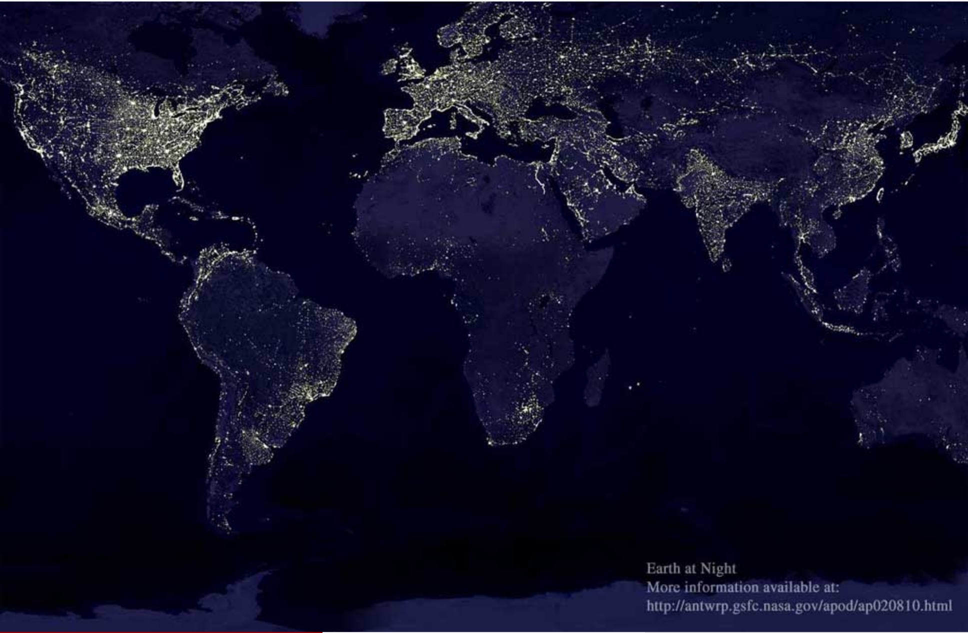


- Next state might be uncertain
- The reward detto
- Advantage: Richer model, robustness
- A transition from X after taking action A:

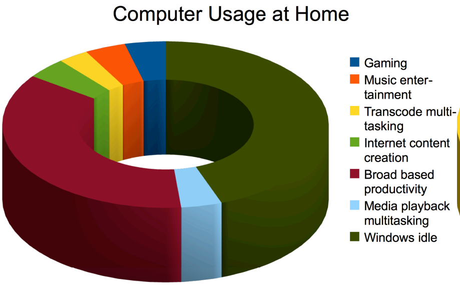
$$
\begin{aligned}
Y &= f(X, A, D), \\
R &= g(X, A, D)
\end{aligned}
$$

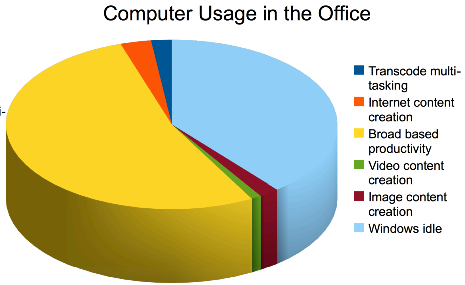
- D – random variable; "disturbance"
- f – transition function
- g – reward function

Power management



Computer usage data





Source: http://www.amd.com/us/Documents/43029A_Brochure_PFD.pdf

Power management

- **Advanced Configuration and Power Interface (ACPI)**
- First released in December 1996, last release in June 2010
- Platform-independent interfaces for hardware discovery, configuration, power management and monitoring

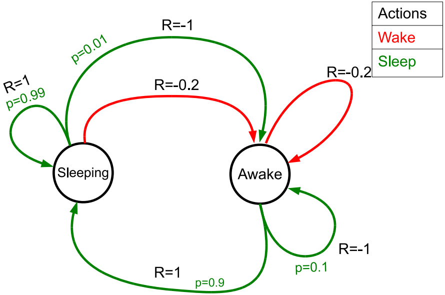
Power mgmt – Power states

- G0 (S0): Working
- G1, Sleeping subdivides into the four states S1 through S4
  - S1: All processor caches are flushed, and the CPU(s) stop executing instructions. Power to the CPU(s) and RAM is maintained; devices that do not indicate they must remain on may be powered down
  - S2: CPU powered off
  - S3: Commonly referred to as Standby, Sleep, or Suspend to RAM. RAM remains powered
  - S4: Hibernation or Suspend to Disk. All content of main memory is saved to non-volatile memory such as a hard drive, and is powered down
- G2 (S5), Soft Off: G2 is almost the same as G3 Mechanical Off, but some components remain powered so the computer can ”wake” from input from the keyboard, clock, modem, LAN, or USB device.
- G3, Mechanical Off: The computer's power consumption approaches close to zero, to the point that the power cord can be removed and the system is safe for dis-assembly (typically, only the real-time clock is running off its own small battery).

Power mgmt – Device, processor, performance states

- Device states
  - D0 Fully-On is the operating state
  - D1 and D2 are intermediate power-states whose definition varies by device.
  - D3 Off has the device powered off and unresponsive to its bus.
- Processor states
  - C0 is the operating state.
  - C1 (often known as Halt) is a state where the processor is not executing instructions, but can return to an executing state essentially instantaneously. All ACPI-conformant processors must support this power state. Some processors, such as the Pentium 4, also support an Enhanced C1 state (C1E or Enhanced Halt State) for lower power consumption.
  - C2 (often known as Stop-Clock) is a state where the processor maintains all software-visible state, but may take longer to wake up. This processor state is optional.
  - C3 (often known as Sleep) is a state where the processor does not need to keep its cache coherent, but maintains other state. Some processors have variations on the C3 state (Deep Sleep, Deeper Sleep, etc.) that differ in how long it takes to wake the processor. This processor state is optional.
- Performance states: While a device or processor operates (D0 and C0, respectively), it can be in one of several power-performance states. These states are implementation-dependent, but P0 is always the highest-performance state, with P1 to Pn being successively lower-performance states, up to an implementation-specific limit of n no greater than 16.
  P-states have become known as SpeedStep in Intel processors, as PowerNow! or Cool'n'Quiet in AMD processors, and as PowerSaver in VIA processors.
  - P0 max power and frequency
  - P1 less than P0, voltage/frequency scaled
  - Pn less than P(n-1), voltage/frequency scaled

An oversimplified model

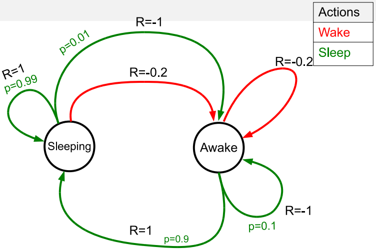


Note

The transitions can be represented as

$$\begin{aligned} Y &= f(x, a, D), \\ R &= g(x, a, D). \end{aligned}$$

Value iteration



function VALUEITERATION

for $x \in \mathcal{X}$ **do** $V[x] \leftarrow 0$

$V' \leftarrow V$

repeat

  for $x \in \mathcal{X} \setminus \{x^*\}$ **do**

    $V[x] \leftarrow \max_{a \in \mathcal{A}(x)} \mathbb{E}\left[g(x,a,D) + \gamma\, V(f(x,a,D))\right]$

  end for

until $V \neq V'$

return V

function BESTACTION(x, V)

return $\operatorname{argmax}_{a \in \mathcal{A}(x)} \mathbb{E}\left[g(x,a,D) + \gamma\, V(f(x,a,D))\right]$

How to gamble if you must?

The safest way to double your money is to fold it over once and put it in your pocket. ("Kin" Hubbard, 1868–1930)

- State $X_t \equiv$ wealth of gambler at step t, $X_t \geq 0$
- Action: $A_t \in [0, 1]$: the fraction of X_t put at stake
- $S_t \in \{-1, +1\}$, $\mathbb{P}\left(S_{t+1} = 1\right) = p$, $p \in [0, 1]$, i.i.d., random variables
- Fortune at next time step:

$$X_{t+1} = (1 + S_{t+1} A_t) X_t.$$

- Goal: maximize the probability that the wealth reaches w^*.
- How to put this into our framework?

How to gamble if you must? – Solution

- $X_t \in \mathcal{X} = [0, w^*]$, $\mathcal{A} = [0, 1]$
- Let $f : \mathcal{X} \times \mathcal{A} \times \{-1, +1\} \to \mathcal{X}$ be

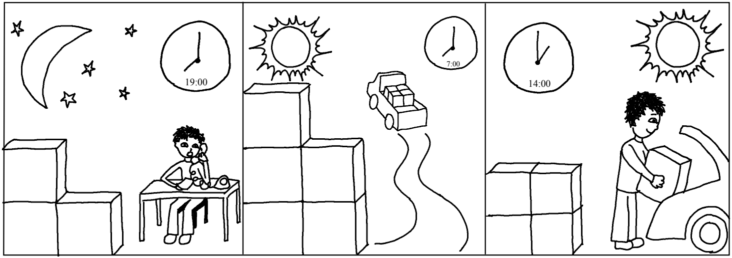
$$f(x, a, s) = \begin{cases} (1 + s\,a)x \wedge w^*, & \text{if } x < w^*; \\ w^*, & \text{otherwise.} \end{cases}$$

- Let $g : \mathcal{X} \times \mathcal{A} \times \{-1, +1\} \to \mathcal{X}$ be

$$g(x, a, s) = \begin{cases} 1, & \text{if } (1 + s\,a)x \geq w^* \text{ and } x < w^*; \\ 0, & \text{otherwise.} \end{cases}$$

- What is the optimal policy?

Inventory control



- $\mathcal{X} = \{0, 1, \ldots, M\}$; X_t size of the inventory in the evening of day t
- $\mathcal{A} = \{0, 1, \ldots, M\}$; A_t number of items ordered in the evening of day t

Dynamics:

$$X_{t+1} = ((X_t + A_t) \wedge M - D_{t+1})^+.$$

Reward:

$$\begin{aligned} R_{t+1} = &-K\,\mathbb{I}_{\{A_t>0\}} - c\,((X_t + A_t) \wedge M - X_t)^+ \\ &- h\,X_t \qquad + p\,((X_t + A_t) \wedge M - X_{t+1})^+. \end{aligned}$$

Other examples

- Engineering, operations research
  - Process control
    - Chemical
    - Electronic
    - Mechanical systems $\Rightarrow$ ROBOTS
  - Supply chain management
- Information theory
  - optimal coding
  - channel allocation
  - sensing, sensor networks
- Finance
  - portfolio management
  - option pricing
- Artificial intelligence
  - The whole problem of acting under uncertainty
  - Search
  - Games
  - Vision: Gaze control
  - Information retrieval

Controlled Markov processes

$$X_{t+1} = f(X_t, A_t, D_{t+1}) \qquad \text{State dynamics}$$
$$R_{t+1} = g(X_t, A_t, D_{t+1}) \qquad \text{Reward}$$
$$t = 0, 1, \ldots .$$

- $X_t \in \mathcal{X}$ – state at time t
- $\mathcal{X}$ – set of states
- $A_t \in \mathcal{A}$ – action at time t
- $\mathcal{A}$ – set of actions
- Sometimes, $\mathcal{A}(x)$: admissible actions
- $R_{t+1} \in \mathbb{R}$ – reward $\Rightarrow \mathbb{R}$
- $D_t \in \mathcal{D}$ – disturbance; i.i.d. sequence
- $\mathcal{D}$ – disturbance space

Return

Definition (Return)

Return, or total discounted return is:

$$\mathcal{R} = \sum_{t=0}^{\infty} \gamma^t R_{t+1},$$

where $0 \leq \gamma \leq 1$ is the so-called discount factor. The return depends on how we act!

The goal of control

Goal

Maximize the expected total discounted reward, or expected return, irrespective of the initial state:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,|\, X_0 = x\right] \to \max!, \quad x \in \mathcal{X}.$$

Alternate definition

Definition (Markov decision process)

Triplet: $(\mathcal{X}, \mathcal{A}, \mathcal{P}_0)$, where

- $\mathcal{X}$ – set of states
- $\mathcal{A}$ – set of actions
- $\mathcal{P}_0$ – state and reward kernel

  $\mathcal{P}_0(U|x,a)$ is the probability that (X_{t+1}, R_{t+1}) lands in $U \subset \mathcal{X} \times \mathbb{R}$ given that $X_t = x$, $A_t = a$

Connection to previous definition

Assume that

$$\begin{aligned} X_{t+1} &= f(X_t, A_t, D_{t+1}) \\ R_{t+1} &= g(X_t, A_t, D_{t+1}) \\ t &= 0, 1, \ldots . \end{aligned}$$

Then

$$\mathcal{P}_0(U|\,x,a) = \mathbb{P}\left(\, [f(x,a,D), g(x,a,D)] \in U \, \right),$$

Here, D has the same distribution as $D_1, D_2, \ldots$.

“Classical form”

Finite MDP (as is often seen in AI publications):

$$(\mathcal{X}, \mathcal{A}, \mathcal{P}, r)$$

- $\mathcal{X}, \mathcal{A}$ are finite.
- $\mathcal{P}(x, a, y)$ is the probability of landing at state y given that action a was chosen in state x
- $r(x, a, y)$ is the expected reward received when making this transition.

Policies, values

Note

From now on we assume that $\mathcal{A}$ is countable.

Definition (General policy)

Maps each history to a distribution over $\mathcal{A}$.
Deterministic policy: $\pi = (\pi_0, \pi_1, \ldots)$, where $\pi_0 : \mathcal{X} \to \mathcal{A}$ and $\pi_t : (\mathcal{X} \times \mathcal{A} \times \mathbb{R})^{t-1} \times \mathcal{X} \to \mathcal{A}$, $t = 1, 2, \ldots$.
Following the policy: $A_t = \pi_t(X_0, A_0, R_1, \ldots, X_{t-1}, A_{t-1}, R_t, X_t)$.

Stationary policies

Definition (Stationary policy)

The map depends on the last state only.

- Deterministic policy: $\pi = (\pi_0, \pi_0, \ldots)$.
  Following the policy: $A_t = \pi_0(X_t)$.
- Stochastic policy: $\pi = (\pi_0, \pi_0, \ldots)$, $\pi_0 : \mathcal{X} \to M_1(\mathcal{A})$.
  Following the policy: $A_t \sim \pi_0(\cdot|X_t)$.

The value of a policy

Definition (Value of a state under π)

The expected return given that the policy is started in state x:

$$V^{\pi}(x) = \mathbb{E}\left[\mathcal{R}^{\pi} | X_0 = x\right].$$

V^{π} – value function of π.

Definition (Action-value of a state-action pair under π)

The expected return given that the process is started from state x, the first action is a after which the policy π is followed:

$$Q^{\pi}(x,a) = \mathbb{E}\left[\mathcal{R}^{\pi} | X_0 = x, A_0 = a\right].$$

Q^{π} – action-value function of π

Optimal values

Definition (Optimal values)

The optimal value of a state is the value of the best possible expected return that can be obtained from that state:

$$V^*(x) = \sup_{\pi} V^{\pi}(x).$$

Similarly, the optimal value of a state-action pair is $Q^*(x,a) = \sup_{\pi} Q^{\pi}(x,a)$.

Definition (Optimal policy)

A policy π is called *optimal* if $V^{\pi}(x) = V^*(x)$ holds for all states $x \in \mathcal{X}$.

The fundamental theorem and the Bellman (optimality) operator

Theorem

Assume that $|\mathcal{A}| < +\infty$. Then the optimal value function satisfies

$$V^*(x) = \max_{a \in \mathcal{A}} \left\{ r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V^*(y) \right\}, \qquad x \in \mathcal{X}.$$

and if policy π is such that in each state x it selects an action that maximizes the r.h.s. then π is an optimal policy.

A shorter way to write this is

$$V^* = T^* V^*,$$

$$(T^* V)(x) = \max_{a \in \mathcal{A}} \left\{ r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V(y) \right\}, \quad x \in \mathcal{X}.$$

Action evaluation operator

Definition (Action evaluation operator)

Let $a \in \mathcal{A}$ and define

$$(T_a V)(x) = r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V(y), \quad x \in \mathcal{X}.$$

Comment

$$T^* V\,[x] = \max_{a \in \mathcal{A}} T_a V\,[x].$$

Policy evaluation operator

Definition (Policy evaluation operator)

Let π be a stochastic stationary policy. Define

$$
\begin{aligned}
(T^\pi V)(x) &= \sum_{a \in \mathcal{A}} \pi(a|x) \left\{ r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V(y) \right\} \\
&= \sum_{a \in \mathcal{A}} \pi(a|x) T_a V(x), \quad x \in \mathcal{X}.
\end{aligned}
$$

Corollary

T^π is a contraction, and V^π is the unique fixed point of T^π.

Greedy policy

Definition (Greedy policy)

Policy π is greedy w.r.t. V if

$$T^{\pi} V = T^{*} V,$$

or

$$\sum_{a \in \mathcal{A}} \pi(a|x) \left\{ r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V(y) \right\} = \max_{a \in \mathcal{A}} \left\{ r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) V(y) \right\}$$

holds for all states x.

A restatement of the main theorem

Theorem

Assume that $|\mathcal{A}| < +\infty$*. Then the optimal value function satisfies the fixed-point equation* $V^* = T^* V^*$ *and any greedy policy w.r.t.* V^* *is optimal.*

Action-value functions

Corollary

Let Q^* be the optimal action-value function. Then,

$$Q^* = T^* Q^*$$

and if π is a policy such that

$$\sum_{a \in \mathcal{A}} \pi(a|x) Q^*(x,a) = \max_{a \in \mathcal{A}} Q^*(x,a)$$

then π is optimal. Here,

$$T^* Q\,(x,a) = r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) \max_{a' \in \mathcal{A}} Q(y,a'), \quad x \in \mathcal{X}, a \in \mathcal{A}.$$

Finding the action-value functions of policies

Theorem

Let π be a stationary policy, T^π be defined by

$$T^\pi Q\,(x,a) = r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathcal{P}(x,a,y) \sum_{a' \in \mathcal{A}} \pi(a'|y)\, Q(y,a'), \quad x \in \mathcal{X}, a \in \mathcal{A}.$$

Then Q^π is the unique solution of

$$T^\pi Q^\pi = Q^\pi.$$

Value iteration – a second look

function ValueIteration
for $x \in \mathcal{X}$ **do** $V[x] \leftarrow 0$
$V' \leftarrow V$
repeat
  for $x \in \mathcal{X} \setminus \{x^*\}$ **do**
    $V[x] \leftarrow T^* V\,[x]$
  end for
until $V \neq V'$
return V

function BestAction(x, V)
return $\operatorname*{argmax}_{a \in \mathcal{A}(x)} T_a V\,[x]$

Value iteration

Note

- *If V_t is the value-function computed in the t^{th} iteration of value iteration then*

$$V_{t+1} = T^* V_t.$$

- *The key is that T^* is a contraction in the supremum norm and Banach's fixed-point theorem gives the key to the proof the theorem mentioned before.*

Note

One can also use $Q_{t+1} = T^* Q_t$, or value functions with post-decision states. What is the advantage?

Policy iteration

function POLICYITERATION(π)

repeat

  $\pi' \leftarrow \pi$

  $V \leftarrow$ GETVALUEFUNCTION(π')

  $\pi \leftarrow$ GETGREEDYPOLICY(V)

until $\pi \neq \pi'$

return π

What if we stop early?

Theorem (e.g., Corollary 2 of Singh and Yee 1994)

Fix an action-value function Q and let π be a greedy policy w.r.t. Q. Then the value of policy π can be lower bounded as follows:

$$V^{\pi}(x) \geq V^{*}(x) - \frac{2}{1-\gamma} \|Q - Q^{*}\|_{\infty}, \quad x \in \mathcal{X}.$$

Books

- Bertsekas and Shreve (1978)
- Puterman (1994)
- Bertsekas (2007a,b)

References

Bertsekas, D. P. (2007a). *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Belmont, MA, 3 edition.

Bertsekas, D. P. (2007b). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, Belmont, MA, 3 edition.

Bertsekas, D. P. and Shreve, S. (1978). *Stochastic Optimal Control (The Discrete Time Case)*. Academic Press, New York.

Puterman, M. (1994). *Markov Decision Processes — Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY.

Singh, S. P. and Yee, R. C. (1994). An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233.