Advanced Topics on Artificial Intelligence

Alban Grastien

The Australian National University

Second Semester, 2020

Second Approach

From Value Iteration to Policy Iteration

General Idea of Policy Iteration

- Start with some policy
- Find a better policy
- Iterate until no better policy can been found (then, we have the optimal policy)

Value Function of a Policy

Given a policy π, the value function V_π is:

$$V_\pi(s) = \begin{cases} 0 & \text{if } s \in G \\ Q_\pi(s, \pi(s)) & \text{otherwise.} \end{cases}$$

$$Q_\pi(s, a) = \Sigma_{s' \in S}\left(P(s, a, s') \cdot (C(s, a, s') + \gamma V_\pi(s)) \right)$$

* Dead-end states need to be handled separately.

Value Function of a Policy

Given a policy π, the value function V_π is:

$$V_\pi(s) = \begin{cases} 0 & \text{if } s \in G \\ Q_\pi(s, \pi(s)) & \text{otherwise.} \end{cases}$$

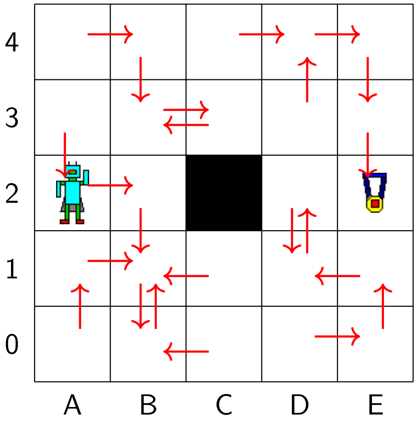
$$Q_\pi(s, a) = \Sigma_{s' \in S}\Big(P(s, a, s') \cdot (C(s, a, s') + \gamma V_\pi(s))\Big)$$

Compare to Bellman Equations:

$$V^*(s) = \begin{cases} 0 & \text{if } s \in G \\ \min_{a \in A(s)} \; Q^*(s, a) & \end{cases}$$

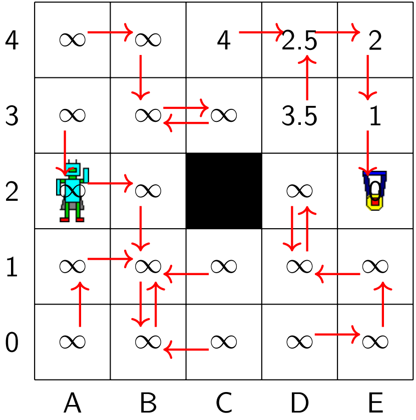
$$Q^*(s, a) = \Sigma_{s' \in S}\Big(P(s, a, s') \cdot (C(s, a, s') + \gamma V^*(s'))\Big)$$

Little Robot: Policy Value



- Choose a random policy
- Compute the value of the states with this policy

Little Robot: Policy Value



- Choose a random policy
- Compute the value of the states with this policy

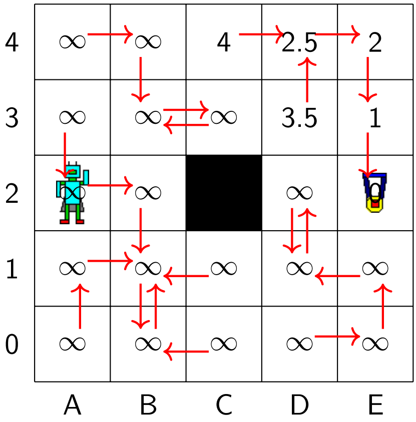
Improving a Policy

Theorem

Let π be a policy and let V_π be the value function of this policy.

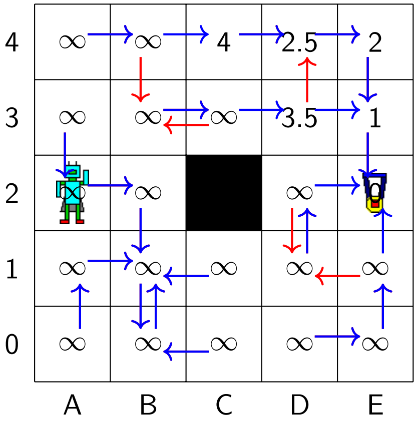
Then π_{V_π} (the policy built from V_π) is better than π unless π is the optimal policy

Little Robot: Computing a new policy



- Choose a random policy
- Compute the value of the states with this policy

Little Robot: Computing a new policy



- Choose a random policy
- Compute the value of the states with this policy
- New policy (in blue)

Policy Iteration

Policy-Iteration

- $t := 0$
- Choose an arbitrary policy π^t
- **repeat**
  - $t \mathrel{+}= 1$
  - $V^{t-1} := V_{\pi^{t-1}}$
  - $\pi^t := \pi_{V^{t-1}}$
- **until** $\pi^t = \pi^{t-1}$
- **return** π^t

Advantages and Drawbacks

Advantage:

- Terminates and computes the optimal policy

Drawbacks:

- Fairly slow (requires to compute the value of many policies)
    - Termination requires to compute this value precisely!
- Estimates the last actions first

Approximate Computation of the Value of a Policy

Approximate-Value-of-Policy

- Input: policy π
- $t := 0$
- Choose an arbitrary value function V^t
- **repeat**
  - $t \mathrel{+}= 1$
  - **for all** $s \in S$
    - $V^t(s) := C(\pi(s)) + \Sigma_{s' \in S} \left(P(s, \pi(s), s') \times V^{t-1}(s') \right)$
- **until** some condition is satisfied
- **return** V^t

Approximate Computation of the Value of a Policy

Comments

- Approximate values of policy can prevent PI from terminating
- Rather than starting with an arbitrary value function V^0, could start with the value function for the previous policy