VALUE ITERATION

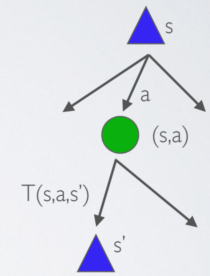
CSE 511A: Introduction to Artificial Intelligence

Some content and images are from slides created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.
All CS188 materials are available at http://ai.berkeley.edu.

1

SOLVING MDPS

- The value (utility) of a q-state *(s,a)*:
  - *Q*(s,a)* = expected utility starting in s, taking action *a*, and thereafter acting optimally.
  - Important note: The action *a* may not be the optimal action to take at state s.
- The value (utility) of a state s:
  - *V*(s)* = expected utility starting in s and acting optimally
  - *V*(s)* = $\max_a$ *Q*(s,a)*
- Optimal policy $\boldsymbol{\pi}$*
  - $\boldsymbol{\pi}$*(s) = optimal action to take at state s.
  - $\boldsymbol{\pi}$*(s) = argmax_a *Q*(s,a)*
    i.e., it is the action *a* that has the largest *Q*(s,a)* over all possible actions



2

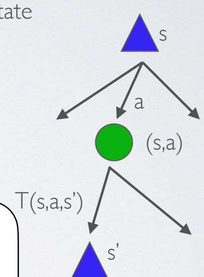
SOLVING MDPS

- Fundamental operation: compute the (expectimax) value of a state
  - Expected utility under optimal action
  - Average sum of discounted rewards
  - Note that this is just like what expectimax computed
- Recursive definition of value:

$$V^*(s) = \max_a Q^*(s,a)$$

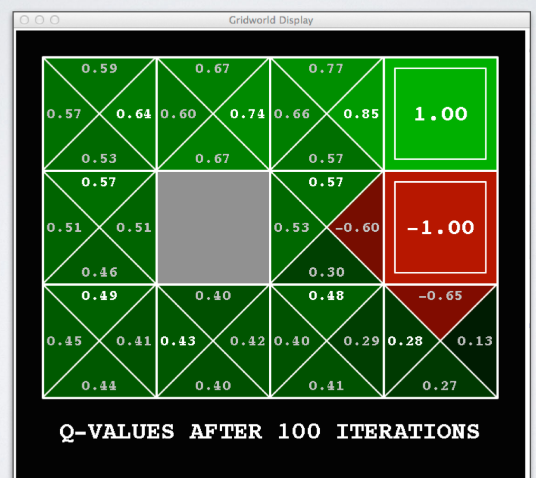
$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$



3

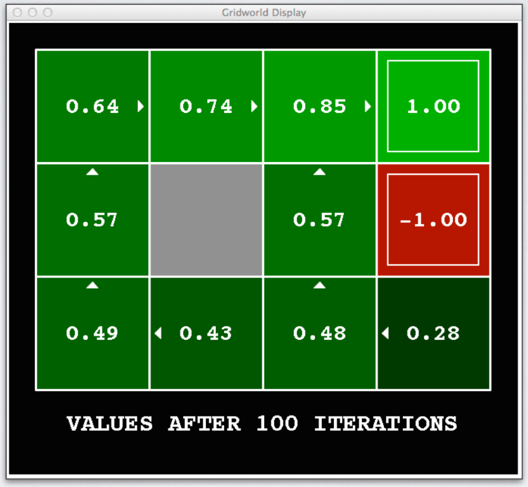
Q*(S,A) VALUES



Noise = 0.2
Discount = 0.9
Living reward = 0

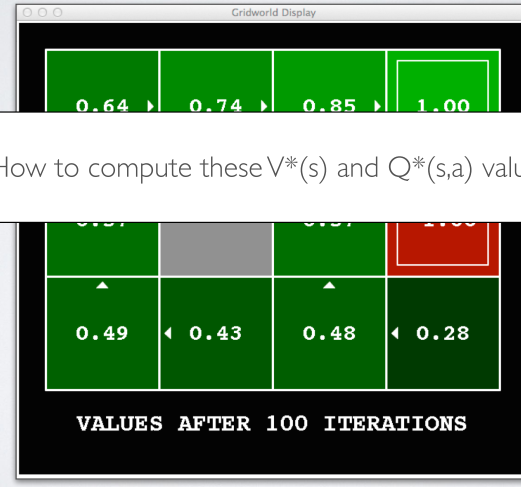
4

V*(S) VALUES



Noise = 0.2
Discount = 0.9
Living reward = 0

V*(S) VALUES



How to compute these V*(s) and Q*(s,a) values?

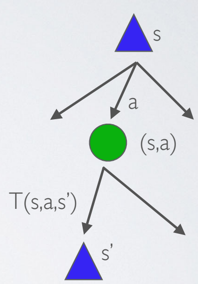
Noise = 0.2
Discount = 0.9
Living reward = 0

SOLVING MDPS

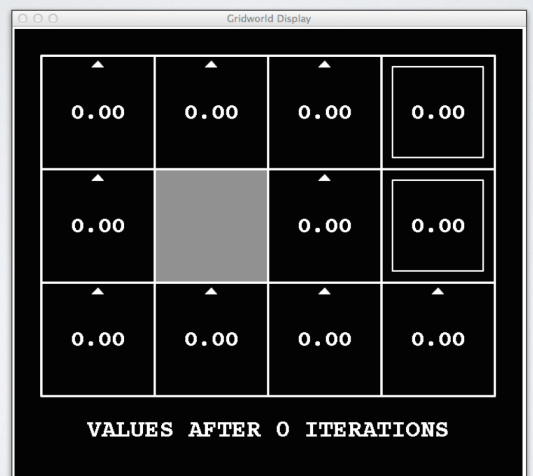
- Iteratively update *V(s)* for all states until they converge
  - Let $V_k(s)$ denote the value of *V(s)* in iteration *k*
  - Initialize $V_0(s) = 0$ for all states *s*
  - Iteratively update values for all states *s* using:

$$\boxed{V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]}$$

  - Stop when values for two subsequent iterations don't change for all states *s*
    - In practice, you can stop when the largest change is less than a very small value (e.g., 0.0001)



VALUE ITERATION



Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

Noise = 0.2
Discount = 0.9
Living reward = 0

VALUE ITERATION

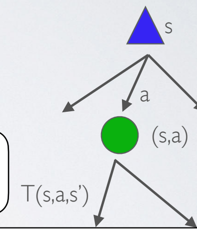
Noise = 0.2
Discount = 0.9
Living reward = 0

21

SOLVING MDPS

- Iteratively update *V(s)* for all states until they converge
  - Let $V_k(s)$ denote the value of *V(s)* in iteration *k*
  - Initialize $V_0(s) = 0$ for all states s
  - Iteratively update values for all states s using:

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$



How to calculate *Q(s,a)*?

22

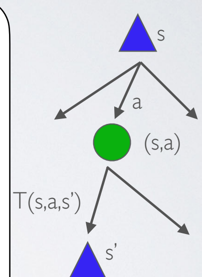
SOLVING MDPS

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$
$$= \max_a Q_{k+1}(s,a)$$

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

or

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$



23

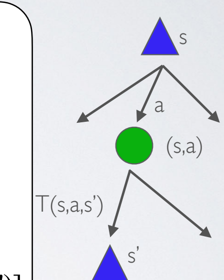
SOLVING MDPS

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$
$$= \max_a Q_{k+1}(s,a)$$

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

or

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$



How to extract converged policy?

24

SOLVING MDPS

$$V_{k+1}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$
$$= \max_a Q_{k+1}(s,a)$$

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

or

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$
$$\pi^*(s) = \arg\max_a Q^*(s,a)$$