

Introduction to Artificial Intelligence

Outline

Complex decisions

Chapter 17, Sections 1–3

Dietter Fox

- ◊ Decision problems
- ◊ Value iteration
- ◊ Policy iteration

Sequential decision problems

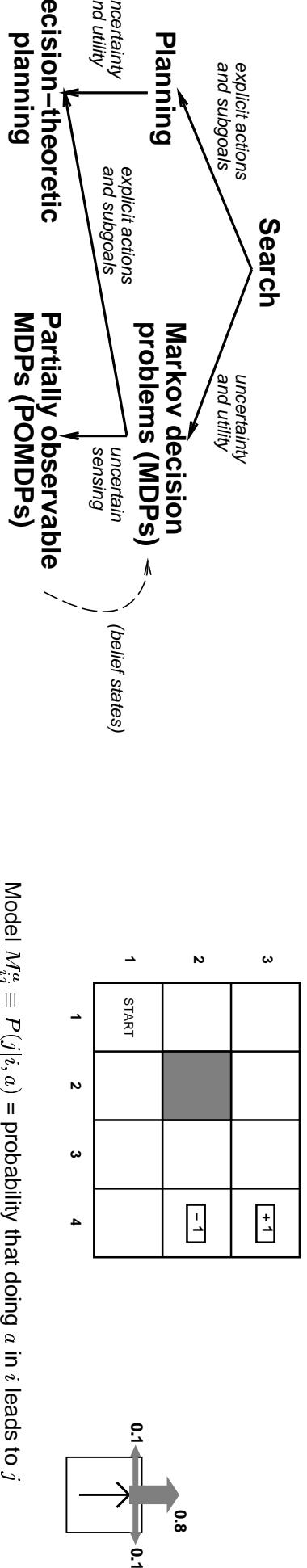
Based on AIMA Slides ©S. Russell and P. Norvig, 1998

Chapter 17, Sections 1–3 0–0

Based on AIMA Slides ©S. Russell and P. Norvig, 1998

Chapter 17, Sections 1–3 1–1

Example MDP



Model $M_{ij}^a \equiv P(j|i, a) =$ probability that doing a in i leads to j

Each state has a *reward* $R(i)$

= -0.04 (small penalty) for nonterminal states

= ±1 for terminal states

Solving MDPs

Utility

In search problems, aim is to find an optimal *sequence*

In MDPs, aim is to find an optimal *policy*
i.e., best action for every possible state
(because can't predict where one will end up)

Optimal policy and state values for the given $R(i)$:

3	→	→	→	[+1]	3	0.812	0.868	0.912	[+1]
2	↑		↑	[-1]	2	0.762		0.660	[-1]
1	↑	→	→	→	1	0.705	0.655	0.611	0.388

1 2 3 4 1 2 3 4

Based on AIMA Slides ©S. Russell and P. Norvig, 1998

Chapter 17, Sections 1–3 4–4

Bellman equation

Definition of utility of states leads to a simple relationship among utilities of neighboring states:

expected sum of rewards
= current reward
+ expected sum of rewards after taking best action

Bellman equation (1957):

$$U(i) = R(i) + \max_a \sum_j U(j) M_{ij}^a$$

$$\begin{aligned} U(1, 1) &= -0.04 \\ &+ \max\{0.8U(1, 2) + 0.1U(2, 1) + 0.1U(1, 1), \\ &+ \max\{0.9U(1, 1) + 0.1U(1, 2) \\ &+ \max\{0.9U(1, 1) + 0.1U(2, 1) \\ &+ \max\{0.8U(2, 1) + 0.1U(1, 2) + 0.1U(1, 1)\} \end{aligned}$$

One equation per state = n nonlinear equations in n unknowns

In sequential decision problems, preferences are expressed between sequences of states

Usually use an additive utility function:

$$U([s_1, s_2, s_3, \dots, s_n]) = R(s_1) + R(s_2) + R(s_3) + \dots + R(s_n)$$

(cf. path cost in search problems)

Utility of a state (a.k.a. its value) is defined to be

$$\begin{aligned} U(s_i) &= \text{expected sum of rewards until termination} \\ U(s_i) &= \text{assuming optimal actions} \end{aligned}$$

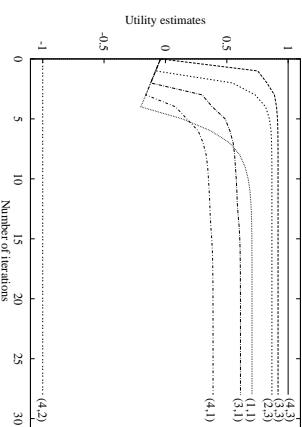
Given the utilities of the states, choosing the best action is just MEU: choose the action such that the expected utility of the immediate successors is highest.

Value iteration algorithm

Idea: Start with arbitrary utility values
Update to make them locally consistent with Bellman eqn.
Everywhere locally consistent \Rightarrow global optimality

repeat until "no change"

$$U(i) \leftarrow R(i) + \max_a \sum_j U(j) M_{ij}^a \quad \text{for all } i$$



Based on AIMA Slides ©S. Russell and P. Norvig, 1998

Chapter 17, Sections 1–3 5–5

Summary

- We can design rational agents based on **probability theory** and **utility theory**
- Sequential decision making in stochastic environments (MDPs) can be solved by computing a **policy**
- **Value iteration** is an algorithm for computing optimal policies.