

Partially observable Markov decision problems

David Keil

University of Connecticut, October 2002

Sources: S. Russell and P. Norvig, *Artificial intelligence: A modern approach* (Prentice Hall, 1995)

		Agent type	
		Deterministic	Stochastic
Environment type	Accessible	Reflex	Solves MDPs
	Inaccessible	Policy-based non-Markov	Solves POMDPs

DK

Overview

- POMDP: Partially observable Markov decision process
- POMDPs are problems in inaccessible stochastic environments with Markov property
- Markov property: probability that system will be in a given state at next step depends only on current state, not on past history (L. Lipsky)

Policy search

- Policy: a mapping from states to actions
- Policy is as opposed to *action sequence*
- Agents that precompute action sequences cannot respond to new sensory information
- Agent that follows a policy incorporates sensory information about state into action determination

RN95, p. 499-500

Stochastic vs. deterministic problems

- Deterministic version of decision problem assures single result of an action
- In stochastic version, an action has a given effect with a probability value
- Example: In a maze, action “move-north” may have probability 0.8 to move north, 0.1 east, 0.1 west

RN95, p. 499

Transition models

- Definition: Set of probability values for given state transitions under given actions
- M_{ij}^a denotes probability of transition from state i to state j on action a
- Transition model is only needed for stochastic problems

Accessible vs. inaccessible environments

- In accessible environment, agent percept identifies current state
- Hence if environment is stochastic, probabilities of state transitions depend only on current state, not on past history of interaction
- Markov decision problems (MDPs) are associated with stochastic problems in accessible environments

Reward vs. utility

- Reward is obtained immediately upon entering a state
- Utility of a state is expected longterm cumulative reward
- Utility can provide guide to rational decisions

RN95, pp. 502-503

Policy search in POMDPs

- *Value iteration* algorithm calculates utility of each state, from which optimal actions can be computed
- *Policy iteration* chooses a policy, calculates utility of state under that policy, then repeats at predecessor states

RN95, pp. 502-505

Value of information vs. reward value

- In POMDPs, utility of an action may be influenced by
 - future reward brought closer by an action
 - future percepts made possible by an action
- Value of information obtained in future percepts must be part of a state-action pair's utility

RN95, pp. 501-502