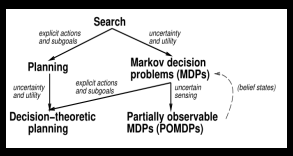
Case Study on MDP

Dr Pradipta Biswas, PhD (Cantab)
Assistant Professor
Indian Institute of Science
https://cambum.net/index.htm

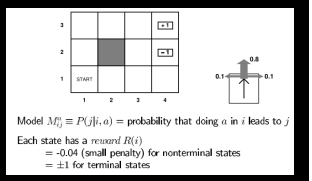
Sequential Decision Problems

- Sequential decision problems vs. episodic ones
- Fully observable environment
- Stochastic actions



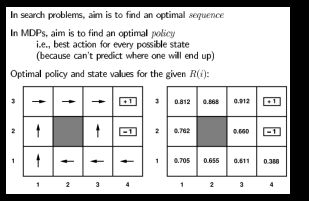
Markov Decision Process (MDP)

1. S_0. Initial state.
2. T(s, a, s'). Transition Model. Yields a 3-D table filled with probabilities. Markovian.
3. R(s). Reward associated with a state. Short term reward for state.



Policy, π

- π(s). Action recommended in state s. This is the solution to the MDP.
- π*(s). Optimal policy. Yields MEU (maximum expected utility) of environment histories.
- Figure 17.2



Utility Function (Long term reward for state)

- Additive. $U_h([s_0, s_1, \ldots]) = R(s_0) + R(s_1) + \ldots$
- Discounted. $U_h([s_0, s_1, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots \quad 0 <= \gamma <= 1$
- γ = gamma

Discounted Rewards Better

- However, there are problems because infinite state sequences can lead to +infinity or –infinity.
  - Set R_{max}, $\gamma < 1$
  - Guarantee that agent will reach goal γ^l
  - Use average reward/step as basis for comparison

Value Iteration

- MEU principle.
  $\pi^*(s) = \text{argmax}_a \sum_{s'} T(s, a, s') * U(s')$
- Bellman Equation.
  $U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U(s')$
- Bellman Update. $U_{i+1}(s) =$
  $R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U_i(s')$

Value Iteration

- Convergence guaranteed!
- Unique solution guaranteed!

Policy Iteration

Figure 17.7

1. Policy Evaluation. Given a policy π_i, calculate the utility of each state.
2. Policy Improvement. Use a one step look ahead to calculate a better policy, π_{i+1}

Bellman Equation (Standard Policy Iteration)

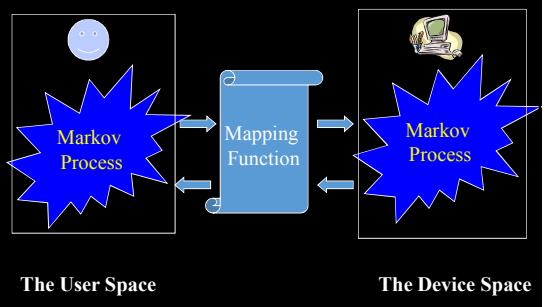
- Old version.
  $U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U(s')$
- New version.
  $U_i(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') * U_i(s')$
- This is a linear equation. With n states, there are n^3 equations to solve.

Bellman Update (Modified Policy Iteration)

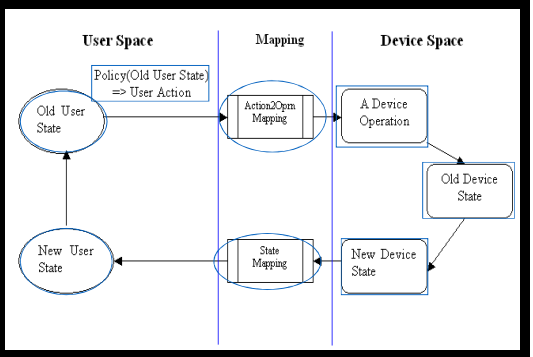
- Old method. $U_{i+1}(s) =$
  $R(s) + \gamma \max_a \sum_{s'} T(s, a, s') * U_i(s')$
- New method. $U_{i+1}(s) =$
  $R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') * U_i(s')$
- Run k updates for an estimation of the utilities. This is called *modified policy iteration*

The Cognitive Model

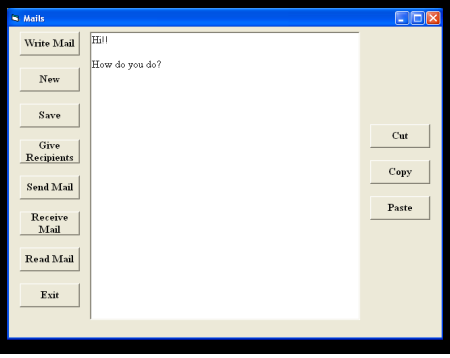
Assumptions for the Cognitive Model



Performance



Sample Application

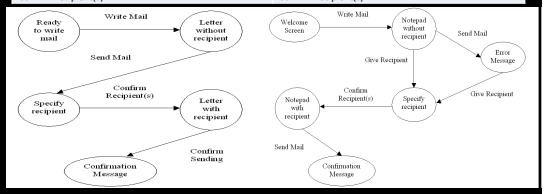


State Spaces

User Space	Device Space
States	
Ready to write mail	Welcome Screen
Letter without recipient Specify recipient	Notepad without recipient Specify recipient
Letter with recipient	Notepad with recipient
Confirmation Message	Confirmation Message
Actions	
Write Mail	Write Mail
Send Mail	Send Mail
Confirm Recipient(s)	Confirm Recipient(s)

State Spaces

User Space	Device Space
States	
Ready to write mail	Welcome Screen
Letter without recipient	Notepad without recipient
Specify recipient	Specify recipient
Letter with recipient	Notepad with recipient
Confirmation Message	Confirmation Message
Actions	
Write Mail	Write Mail
Send Mail	Send Mail
Confirm Recipient(s)	Confirm Recipient(s)



Execution of the Model

	Device Space	User Space
Iteration 1		
State	Welcome Screen	Ready to write mail
Action	Write Mail	WriteMail
State	Notepad without recipient	Letter without recipient
Action	SendMail	SendMail
State	ErrorMsg	
New Action Learned	GiveRecipients	
State	Specify recipient	Specify recipient
Action	ConfirmRecipient	ConfirmRecipient
State	Notepad with recipient	Letter with recipient
New Action Learned	SendMail	
Action	SendMail	Confirm Sending
State	Confirmation	Confirmation

Examplle of Learninng

	Device Space	User Space
Iteration 2		
State	Welcome Screen	Ready to write mail
Action	Write Mail	WriteMail
State	Notepad without recipient	Letter without recipient
Action	GiveRecipients	GiveRecipients
State	Recipient	Recipient
Action	ConfirmRecipient	ConfirmRecipient
State	Notepad with recipient	Letter with recipient
Action	SendMail	Confirm Sending
State	Confirmation	Confirmation

Features of Cognitive Model

- Learn a new state or a new operation.
- Support the label matching principle.
- Take instruction during execution.
- Model the practice effect.
- Has user-friendly interfaces for development and execution.