Utilities and MDP: A Lesson in Multiagent System

Henry Hexmoor

SIUC

Utility

- Preferences are recorded as a utility function

$$u_i : S \rightarrow R$$

S is the set of observable states in the world

u_i is utility function

R is real numbers

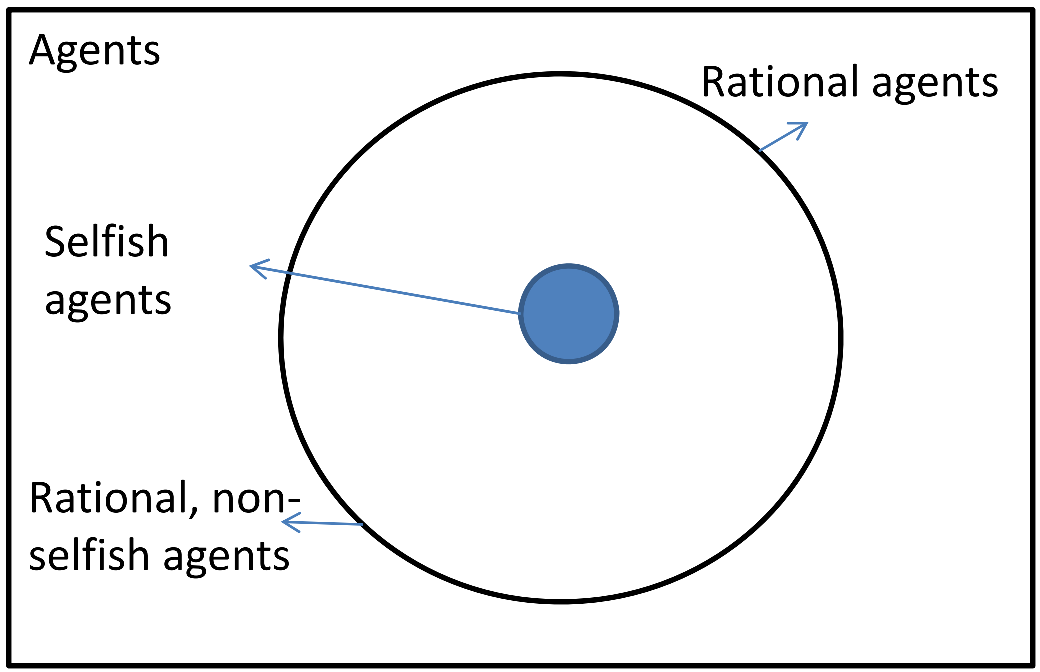
- States of the world become ordered.

Properties of Utilites

- Reflexive: $u_i(s) \geq u_i(s)$
- Transitive: *If* $u_i(a) \geq u_i(b)$ *and* $u_i(b) \geq u_i(c)$ *then* $u_i(a) \geq u_i(c)$.
- Comparable: a,b *either* $u_i(a) \geq u_i(b)$ *or* $u_i(b) \geq u_i(a)$.

Selfish agents:

- A rational agent is one that wants to maximize its utilities, but intends no harm.



Utility is not money:

- while utility represents an agent’s preferences it is not necessarily equated with money. In fact, the utility of money has been found to be roughly logarithmic.

Marginal Utility

- Marginal utility is the utility gained from next event

Example:

getting A for an A student.

versus A for an B student

Transition function

Transition function is represented as

$$T(s, a, s')$$

Transition function is defined as the probability of reaching S' from S with action 'a'

Expected Utility

- Expected utility is defined as the sum of product of the probabilities of reaching s’ from s   with action ‘a’ and utility of the final state.

$$E[u_i, s, a] = \sum_{s' \in S} T(s, a, s') u_i(s')$$

Where S is set of all possible states

Value of Information

- Value of information that current state is t and not s:

$$\Delta E = E[u_i, t, \pi_i(t)] - E[u_i, t, \pi_i(s)]$$

here $E[u_i, t, \pi_i(t)]$ represents updated, new info

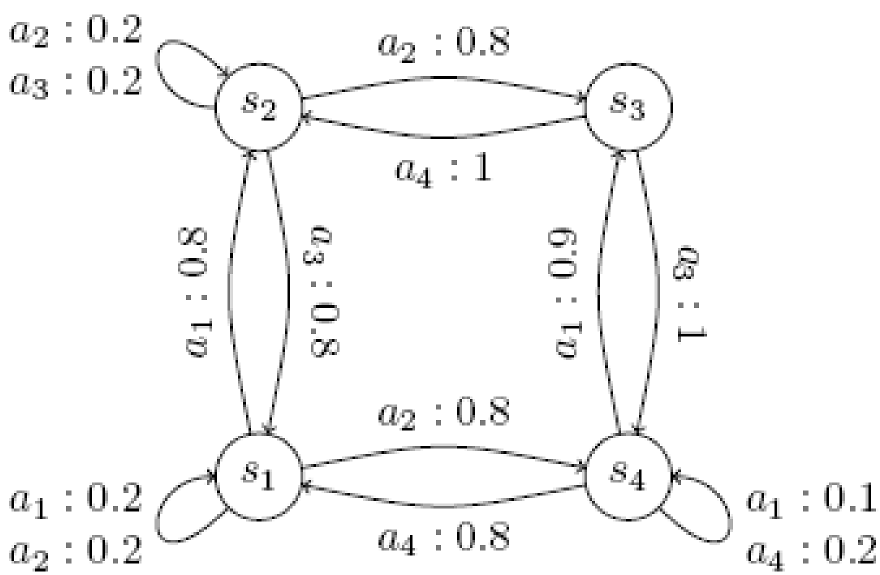
$E[u_i, t, \pi_i(s)]$ represents old value

Markov Decision Processes: MDP

- Graphical representation of a sample Markov decision process along with values for the transition and reward functions. We let the start state be s1. E.g.,

s	$r(s)$
s_1	0
s_2	0
s_3	1
s_4	0

s_i	a	s_j	$T(s_i, a, s_j)$
s_1	a_1	s_1	0.2
s_1	a_1	s_2	0.8
s_1	a_2	s_1	0.2
s_1	a_2	s_4	0.8
s_2	a_2	s_2	0.2
s_2	a_2	s_3	0.8
s_2	a_3	s_2	0.2
s_2	a_3	s_1	0.8
s_3	a_4	s_2	1
s_3	a_3	s_1	1
s_4	a_1	s_4	0.1
s_4	a_1	s_3	0.9
s_4	a_4	s_4	0.2
s_4	a_4	s_1	0.8



Reward Function: r(s)

- Reward function is represented as

$$r : S \rightarrow R$$

Deterministic Vs Non-Deterministic

- Deterministic world: predictable effects

  Example: only one action leads to T=1 , else Φ

- Nondeterministic world: values change

Policy: π

- Policy is behavior of agents that maps states to action
- Policy is represented by π

Optimal Policy

- Optimal policy is a policy that maximizes expected utility.
- Optimal policy is represented as π^*

$$\pi_i^*(s) = \arg_{a \in A} \max E[u_i, s, a]$$

Discounted Rewards: $\gamma(0-1)$

- Discounted rewards smoothly reduce the impact of rewards that are farther off in the future

$$\gamma^0 r(s_1) + \gamma^1 r(s_2) + \gamma^2 r(s_3) + \ldots$$

Where $\gamma(0-1)$ represents discount factor

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s, a, s') u(s')$$

Bellman Equation

$$u(s) = r(s) + \gamma \max_a \sum_{s'} T(s, a, s')u(s')$$

Where

r(s) represents immediate reward

T(s,a,s')u(s') represents

future, discounted rewards

Brute Force Solution

- Write n Bellman equations one for each n states, solve …
- This is a non-linear equation due to $\max_{a}$

Value Iteration Solution

- Set values of u(s) to random numbers
- Use Bellman update equation

$$u^{t+1}(s) \longleftarrow r(s) + \gamma \max_a \sum_{s'} T(s,a,s') u^t(s')$$

- Converge and stop using this equation when

$$\Delta u < \frac{\in (1-\gamma)}{\gamma}$$

where Δu max utility change

Value Iteration Algorithm

$$VALUE - ITERATION \quad (T, r, \gamma, \in)$$

do

$u \leftarrow u'$

$\delta \leftarrow \phi$

$for \quad s \in S$

$$do \quad u'(s) \longleftarrow r(s) + \gamma \max_a \sum_{s'} T(s, a, s') u(s')$$

$$if \quad |u'(s) - u(s)| > \delta$$

$$then \quad \delta \leftarrow |u'(s) - u(s)|$$

$$until \quad \delta < \frac{\in (1 - \gamma)}{\gamma}$$

$return \quad u$

$\gamma = 0.5$ *and* $\in = 0.15.$ The algorithm stops after t=4

	Time (t)				
	0	1	2	3	4
$u(s_1)$	0	0	0	$.5(.8).45 = .18$	$.5(.09 + .378) = .23$
$u(s_2)$	0	0	$.5(.8)1 = .4$	$.5(.88)1 = .44$	$.5(.18 + .98) = .57$
$u(s_3)$	0	1	1	$1 + .5(1).45 = 1.2$	$1 + .5(.47) = 1.2$
$u(s_4)$	0	0	$.5(.9)1 = .45$	$.5(.9 + .045) = .47$	$.5(1.1 + .047) = .57$

s	$\pi^*(s)$
s_1	a_2
s_2	a_2
s_3	a_3
s_4	a_1

MDP for one agent

- Multiagent: one agent changes , others are stationary.
- Better approach $\rightarrow T(s, \vec{a}, s')$

  $\vec{a}$ is a vector of size ‘n’ showing each agent’s action. Where ‘n’ represents number of agents
- Rewards:
  - Dole out equally among agents
  - Reward proportional to contribution

Observation model

- noise + cannot observe world …
- Belief state $\vec{b} = <P_1, P_2, P_3, \ldots, P_n>$

- Observation model O(s,o) = probability of observing 'o', being in state 's'.

$$\forall_{s'} \quad \overrightarrow{b'}(s') = \alpha O(s', o) \sum_{s} T(s, a, s') \vec{b}(s)$$

Where α is normalization constant

Partially observable MDP

$$T(\vec{b}, a, \vec{b}') = \begin{cases} \sum_{s'} O(s', o) \sum_{s} T(s, a, s') \vec{b}(s) & \text{if * holds} \\ 0 & \text{otherwise} \end{cases}$$

* - $\forall_{s'} \quad \vec{b}'(s') = \alpha O(s', o) \sum_{s} T(s, a, s') \vec{b}(s)$ is true for $\vec{b}, a, \vec{b}$

new reward function

$$\rho(\vec{b}) = \sum_{s} \vec{b}(s) r(s)$$

- Solving POMDP is hard.