Aalto University
School of Science

Value iteration method for solving Markov decision processes

Ville Tuominen

Presentation *12*

30.10.2020

MS-E2191 Graduate Seminar on Operations Research
Fall 2020

Markov Decision Process

- In each timestep n the process is in state i
- Decision Maker chooses action a in each step n
- Process moves to state j with probability $P_a(i,j)$ independently from previous states and actions

Maximum value and optimal policy is

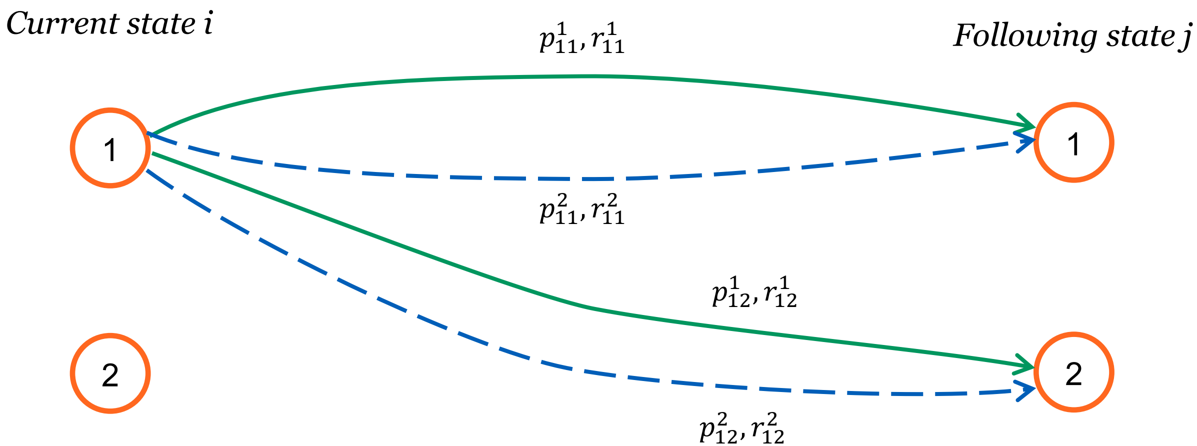
$$v_i = \max_{a \in A} \sum_{j=1}^{S} p_{ij}^a \left(r_{ij}^a + \gamma v_j \right)$$

S = states, $1, \ldots, S$
γ = discount factor

r_{ij}^a = reward when transitioning from state *i* to *j* in action *a*
p_{ij}^a = probability of transitioning from *i* to *j* in action *a*

Example problem

Toymaker's problem: Craft Beer company that has premises for 4 months (Howard, 1960)



State	Action	Transition probabilities		Reward		Expected immediate reward
i	*a*	p_{i1}^a	p_{i2}^a	r_{i1}^a	r_{i2}^a	q_i^a
1 Successful beer	1 No ads	0.5	0.5	9	3	6
	2 Ads	0.8	0.2	4	4	4
2 Unsuccesful beer	1 No research	0.4	0.6	3	-7	-3
	2 Research	0.7	0.3	1	-19	-5

Value Iteration

Recursive method using Principle of Optimality.

Total expected return in *n* steps staring from state *i* with optimal policy is

$$v_i(n+1) = \max_{a\in A} \sum_{j=1}^{S} p_{ij}^a \left(r_{ij}^a + \gamma v_j(n)\right)$$

$$= \max_{a\in A} \mathrm{q}_\mathrm{i}^a + \gamma \textstyle\sum_{j=1}^{S} p_{ij}^a v_j(n)$$

S = states, *1,…,S*
n = step, *0,…,N*
γ = discount factor

r_{ij}^a = reward when transitioning from state *i* to *j* in action *a*
p_{ij}^a = probability of transitioning from *i* to *j* in action *a*
q_i^a = expected immediate reward of state *i* in action *a*

Example problem solution

Choose initial value $v_j(0)$ (e.g. 0) and find decision $d_s(n)$ for all states *s* and steps *n* using Value Iteration.

Value and decision for each step starting from states 1 and 2

n	0	1	2	3	4
$v_1(n)$	0	6	8,2	10,22	12,22
$v_2(n)$	0	-3	-1,7	0,23	2,22
$\boldsymbol{d_1(n)}$	**-**	**1**	**2**	**2**	**2**
$\boldsymbol{d_2(n)}$	**-**	**1**	**2**	**2**	**2**

Problems

Value Iteration converges to the best alternative (optimal policy) for each state *s* when *n* grows

- When *n* is large enough?
- Not ideal for long plans (at least in the 1960s)

Discounted infinite horizon problem

In addition to policy, value v_i converges to optimal v_i^* when $\gamma < 1$ and *n* tends to infinity.

Using Error Bounds

- Value Iteration may convergence to optimal costs faster
- Possible to analyse the convergence from P_{A^*} and μ
- Discard unnecessary controls to speed up computation

Lower bound $\underline{c_n} = \frac{\gamma}{1-\gamma} \min_{i=1,..,n} v_i(n) - v_i(n-1)$

Upper bound $\overline{c_n} = \frac{\gamma}{1-\gamma} \max_{i=1,..,n} v_i(n) - v_i(n-1)$

Other methdos

There exists alternative algorithms for value iteration.

Gauss-Seidel

- Iterates one state at a time
- Faster convergence unless parallel computation can be used

Q-Learning

- When the transition probabilities are unknown
- Based on Q-factor of control and state

Policy iteration

- *Next week*

References

- *Bertsekas, D. P. (2012). Dynamic programming and optimal control (Vol. 2, 4th ed.) Approximate Dynamic Programming. Belmont, MA: Athena scientific. (p. 82-97)*
- *Howard, R. A. (1960). Dynamic programming and markov processes. John Wiley & Sons (p. 26-31)*

Homework

Instructions

1. Download Jupyter Notebook template from MyCourses
2. Log in to https://jupyter.cs.aalto.fi/
3. Launch *R: General use* server
4. Upload the notebook to your favourite folder

Homework

1. **Fill in the missing parts and solve the Craft Beer company example using *MDPtoolbox* library**
2. **Solve Forest Management problem and examine the effect of discount factor and the probability of wildfires**

Use presentation for values and given documentation for examples. Write answers to notebook cells (you can change the cell type to Markdown).

DL: 6.11. 9.00, send to ville.m.tuominen[at]aalto.fi