MDP_ValueIteration_results

November 4, 2020

0.0.1 MS-E2191 - Graduate Seminar on Operations Research

Value iteration for solving Markov Decision Processes

0.1 1 Craft beer company

Using provided documentation and presentation material, fill in the missing 4 parts and solve the example from the presentation using Value Iteration. Use discount factor 0.9.

We use MDPtoolbox, which is a package available for Python, R and Matlab. Its documentation can be found here https://cran.r-project.org/web/packages/MDPtoolbox/MDPtoolbox.pdf

Preparations

[ ]:
```
STUDENT_NAME = ## FILL YOUR NAME ##
```

[1]:
```
# Run cell to install and add package
install.packages("MDPtoolbox")
library(MDPtoolbox)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘lpSolve’, ‘linprog’


Loading required package: Matrix

Loading required package: linprog

Loading required package: lpSolve
```

Solving the problem See documentation for examples

[63]:
```
# Transition probabilities
P <- array(0, c(2,2,2))
P[,,1] <- matrix(c(0.5, 0.5, 0.4, 0.6), 2, 2, byrow=TRUE) # for action 1
P[,,2] <- matrix(c(0.8, 0.2, 0.7, 0.3), 2, 2, byrow=TRUE) # for action 2
```

```
# Rewards
# Only the expected rewards
R<- matrix(c(6, 4, -3, -5), 2, 2, byrow=TRUE)

# or all rewards
R <- array(0, c(2,2,2))
R[,,1] <- matrix(c(9, 3, 3, -7), 2, 2, byrow=TRUE) # for action 1
R[,,2] <- matrix(c(4, 4, 1, -19), 2, 2, byrow=TRUE) # for action 2
```

[60]:
```
P[,,1] # for action 1
```

A matrix: 2 × 2 of type dbl

0.5	0.5
0.4	0.6

[61]:
```
R
```

A matrix: 2 × 2 of type dbl

6	4
-3	-5

[66]:
```
# Solve using Value Iteration
mdp_value_iteration(P, R, 0.9)
```

```
[1] "MDP Toolbox: iterations stopped, epsilon-optimal policy found"

$V
[1] 11.700189  1.810138

$policy
[1] 2 2

$iter
[1] 5

$time
Time difference of 0.001009941 secs

$epsilon
[1] 0.01

$discount
[1] 0.9
```

With 5 iterations, we wind the same optimal policy as in the presentation: choose action 2 for each initial state.

The algorithm doesn't converge to exactly same value function, since here the discount factor is used. When we use discount factor 1.0, we get the same values as in the presentation.

0.2 2 Forest management

Let's consider forest managemenet problem, where we can either keep the forest as it is or cut the forest. For every three year, we choose either Keep or Cut action. Every time we get 1 unit of money from cutting part of the forest and 0 from leaving it as it is. In the last year, we get either 3 units of money by cutting the rest of the forest or 3 units by conserving.

We start growing the forest after a wildfire. It may burn again with probability p.

Task Solve the problem using value iteration, similarly to the first exercise.

A Start with discount factor 0.9. How different values of discount factor change the policy? How different values of discount factor affect the convergence of Value Iteration?

B With discount factor 0.9, the probability of wildfire inceases to 0.3. How does the optimal policy change?

Values

[81]:
```
S = 10 # number of steps
r1 = 3 # reward for conservation
r2 = 3 # reward for cutting the rest of the forest
p = 0.05 # wildfire probability

values = mdp_example_forest(S, r1, r2, p) # generates values for this example
P = values$P # transition probabilities
R = values$R # rewards
```

[68]:
```
R
```

A matrix: 10 × 2 of type dbl

R1	R2
0	0
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
3	3

Solving the problem

[82]:
```
mdp_value_iteration(P, R, 0.9)
```

```
[1] "MDP Toolbox: iterations stopped, epsilon-optimal policy found"

$V
 [1]  4.184232  5.040975  6.043012  7.214986  8.585716 10.188909 12.063988
 [8] 14.257063 16.822063 19.822063
```

```
$policy
 [1] 1 1 1 1 1 1 1 1 1 1

$iter
[1] 16

$time
Time difference of 0.003672123 secs

$epsilon
[1] 0.01

$discount
[1] 0.9
```

A Discount factor represents our attitude towards valueing current profits more than upcoming profits. With low discount factor, we value future profits little.

Policy Here, when the discount factor gets smaller, we tend to choose action 2 more (cut forest and sell it). With higher values, we tend to choose action 1 more (keep the forest as it is). We value the short-term profits more.

Convergence With small values of discount factor, the number of iterations is small. The maximum number of iterations is obtained at around 0.8. With values > 0.9, the number drops again. With smaller values of discount factor, the values of previous iterations gets small so the computation gets easier.

B When the risk for wildfires is higher, the optimal policy is to cut (and sell) the forest early on as the risk of forest burning gets higher when years pass.

```
[ ]:
```