Dynamic Programming & Reinforcement Learning

Lecture 5
The Bellman equation

Ger Koole

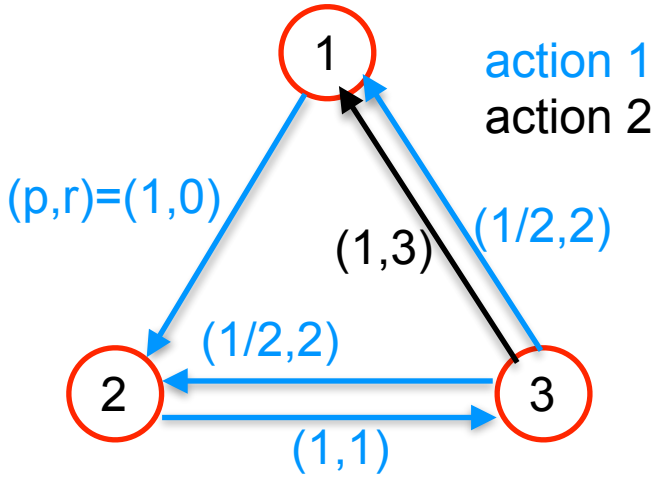
Markov decision chains: policy iteration (PI)

- Define:
  - $r_\alpha(x) = r(x, \alpha(x))$
  - $P_\alpha(x, y) = p(y \mid x, \alpha(x))$
- Policy iteration:

0. Fix a policy α
1. Find a solution (V_α, ϕ_α) of $V + \phi e = r_\alpha + P_\alpha V$
2. $\tilde{\alpha} = \arg\max_{\alpha'} \{ r_{\alpha'} + P_{\alpha'} V_\alpha \}$
3. If $\alpha = \tilde{\alpha}$: terminate with (V_α, ϕ_α) optimal

   Else: $\alpha = \tilde{\alpha}$ and go to step 1

Example



- Step 0: take $\alpha = (1,1,1)$
- Step 1: ((0,1.2,1.4),1.2) is a solution
- Step 2 in state 3:
  $\alpha'(3) = \arg\max\{2 + 0.5(V_\alpha(1) + V_\alpha(2)),3 + V_\alpha(1)\} = 2$
- Step 3: $\alpha = (1,1,2)$ and go to step 1
- Step 1: ((0,1.33,1.66),1.33) is a solution
- Step 2 in state 3: $\alpha'(3) = 2$
- Step 3: $\alpha = \alpha'$ and termination

Why does PI work?

$u \geq v$ for vector u, v if $u(x) \geq v(x) \forall x$

Theorem: If $r_{\alpha'} + P_{\alpha'} V_\alpha \geq r_\alpha + P_\alpha V_\alpha$ then

$$\phi_{\alpha'} \geq \phi_\alpha$$

Proof: Define $J_\alpha V = r_\alpha + P_\alpha V$. Thus $J'_\alpha V_\alpha \geq J_\alpha V_\alpha$. Then

$J'_\alpha V_\alpha \geq J_\alpha V_\alpha$ & convex combinations preserve $\geq$

$P'_\alpha \phi_\alpha e = \phi_\alpha e$

$$J^2_{\alpha'} V_\alpha \geq J_{\alpha'} J_\alpha V_\alpha = J_{\alpha'}(\phi_\alpha e + V_\alpha) \geq r_\alpha + P_\alpha V_\alpha + \phi_\alpha e = J^2_\alpha V_\alpha$$

Poisson equation

$r_\alpha + P_\alpha V_\alpha + P_\alpha \phi_\alpha e = J_\alpha(\phi_\alpha e + V_\alpha)$

This argument can be repeated for any t

But then

$$\phi_{\alpha'} = \lim_{t \to \infty} \frac{J^t_{\alpha'} V_\alpha}{t} \geq \lim_{t \to \infty} \frac{J^t_\alpha V_\alpha}{t} = \phi_\alpha$$

Bellman equation

- For the optimal policy α_* it holds that

$$r_{\alpha_*} + P_{\alpha_*} V_{\alpha_*} = \max_{\alpha'} \{ r_{\alpha'} + P_{\alpha'} V_{\alpha_*} \}$$

- For any policy and thus also α_* it holds that

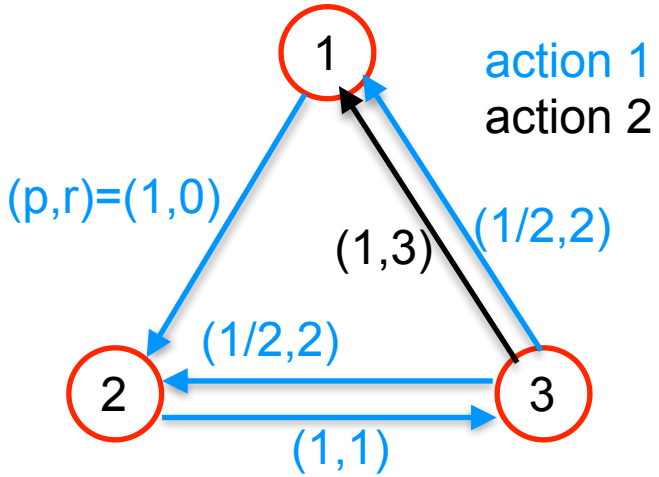
$$V_{\alpha_*} + \phi_{\alpha_*} e = r_{\alpha_*} + P_{\alpha_*} V_{\alpha_*}$$

- Combined: $(V_{\alpha_*}, \phi_{\alpha_*})$ is solution of

$$V + \phi e = \max_{\alpha} \{ r_{\alpha} + P_{\alpha} V \}$$

- *Optimality equation* or *Bellman equation*

Example



- Bellman equation:

$$V(1) + \phi = V(2)$$

$$V(2) + \phi = 1 + V(3)$$

$$V(3) + \phi = \max\{2 + \frac{1}{2}V(1) + \frac{1}{2}V(2), 3 + V(1)\}$$

- $(V, \phi) = ((0, 4/3, 5/3), 4/3)$ is indeed a solution
- But how to solve such a non-linear equation?
- Policy iteration or value iteration

Backward recursion

- Markov decision chains: replace recursion by

$$V_{t+1} = \max_{\alpha} \{ r_\alpha + P_\alpha V_t \}$$

- Per state instead of as a vector:

$$V_{t+1}(x) = \max_{a} \{ r(x,a) + \sum_{y} p(y \mid x,a) V_t(y) \}$$

- Maximizing policy for t big is long-run optimal policy
- $(V_t - V_t(0)e, V_{t+1}(x) - V_t(x))$ for any x and t big enough is solution to Bellman equation
- aka value iteration (VI)

Q-values

- Can be helpful to determine:

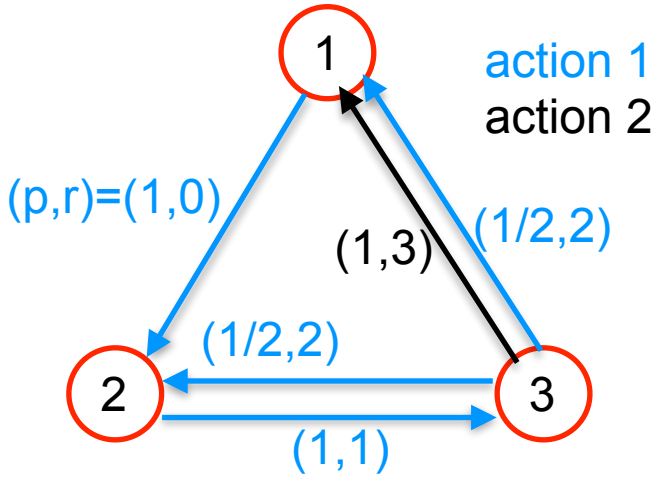
$$Q_{t+1}(x,a) = r(x,a) + \sum_y p(y \mid x,a) V_t(y)$$

- Recursion:

$$Q_{t+1}(x,a) = r(x,a) + \sum_y p(y \mid x,a) V_t(y) = r(x,a) + \sum_y p(y \mid x,a) \max_{a'} Q_t(y,a')$$

- Advantage: easy to determine optimal policy
- Disadvantage: additional state variable
- Using if you do not know r and/or P (RL)

Example




```
V=c(0,0,0);r1=c(0,1,2);r2=c(0,1,3)
P1=matrix(c(0,1,0,0,0,1,0.5,0.5,0),nrow=3,byrow=TRUE)
P2=matrix(c(0,1,0,0,0,1,1,0,0),nrow=3,byrow=TRUE)
for(n in 1:100)V=pmax(r1+P1%*%V,r2+P2%*%V)
a=which.max(c(r1[3]+P1[3,]%*%V,r2[3]+P2[3,]%*%V))
a;V-min(V);if(a==1){r1+P1%*%V-V}else{r2+P2%*%V-V}
```

```
[1] 2
```

a

```
     [,1]
[1,]    0
[2,]    2
[3,]    3
```

V

```
     [,1]
[1,]    2
[2,]    2
[3,]    0
```

ϕe

Not a constant vector!
What is wrong??

Poll!

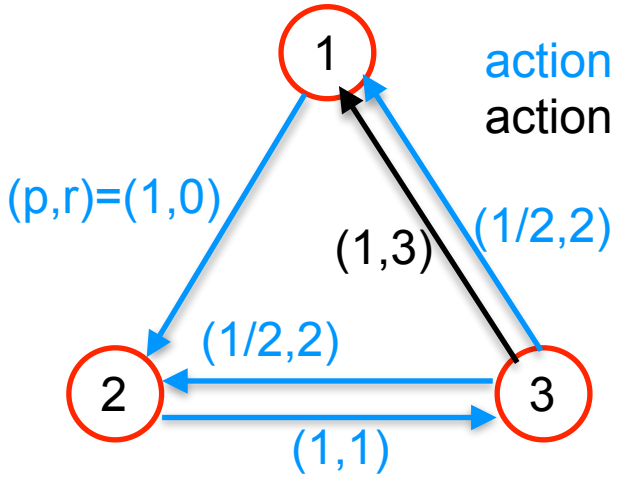
Periodicity

- Consider a periodic Markov (decision) chain
  - Solution Poisson/Bellman equation? 👍
  - Convergence backward recursion? 👎
- Repair convergence by introducing aperiodicity: replace P by $P' = \delta P + (1-\delta)I$ for some $\delta \in (0,1)$
- P' is aperiodic and we have convergence
- If (V'_*, ϕ'_*) solution of $V + \phi e = \max_\alpha \{r_\alpha + P'_\alpha V\}$

  then $(V_*, \phi_*) = (\delta V'_*, \phi'_*)$ solution of

$$V + \phi e = \max_\alpha \{r_\alpha + P_\alpha V\}$$

Example




```
V=c(0,0,0);r1=c(0,1,2);r2=c(0,1,3)
P1=0.5*(matrix(c(0,1,0,0,0,1,0.5,0.5,0),nrow=3,byrow=TRUE)+diag(3))
P2=0.5*(matrix(c(0,1,0,0,0,1,1,0,0),nrow=3,byrow=TRUE)+diag(3))
for(n in 1:100)V=pmax(r1+P1%*%V,r2+P2%*%V)
a=which.max(c(r1[3]+P1[3,]%*%V,r2[3]+P2[3,]%*%V))
a;V-min(V);if(a==1){r1+P1%*%V-V}else{r2+P2%*%V-V}
```

```
[1] 2
```

a

```
          [,1]
[1,] 0.000000
[2,] 2.666667
[3,] 3.333333
```

$V', \delta = 0.5 \Rightarrow V_* = (0,4/3,5/3)$

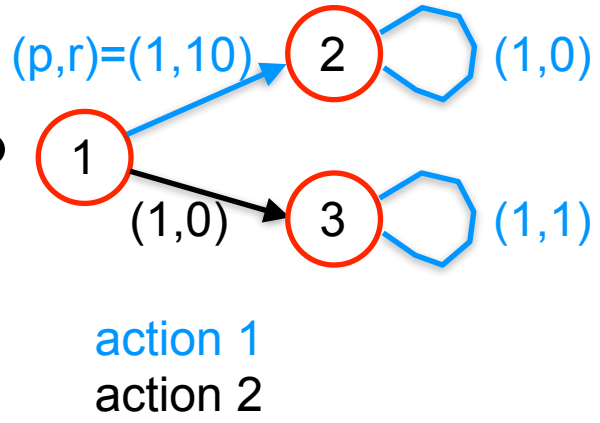
```
          [,1]
[1,] 1.333333
[2,] 1.333333
[3,] 1.333333
```

ϕe

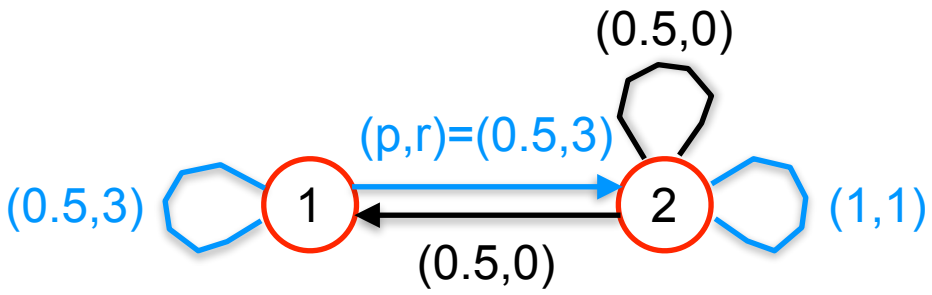
Non-communicating chains

- Theorem: For communicating chains the (maximal) long-run average reward is independent of initial state or distribution (and thus of form ϕe)
- Example non-communicating chain:
- What is its average reward?

Poll!



Final example

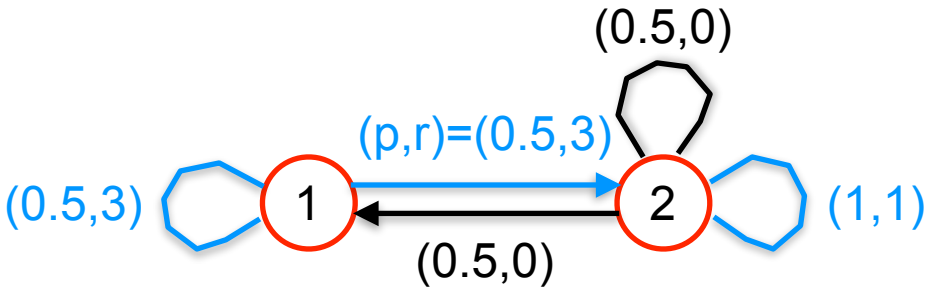


action 1
action 2

Find the average optimal solution of this problem using 3 methods:

- Policy iteration (by hand)
- By solving the Bellman equation (by hand)
- Value iteration (in some tool)

Policy iteration



0. Fix a policy α
1. Find a solution (V_α, ϕ_α) of $V + \phi = r_\alpha + P_\alpha V$
2. $\tilde{\alpha} = \arg\max_{\alpha'} \{r_{\alpha'} + P_{\alpha'} V_\alpha\}$
3. If $\alpha = \tilde{\alpha}$: terminate with (V_α, ϕ_α) optimal
   Else: $\alpha = \tilde{\alpha}$ and go to step 1

0. $\alpha = (1,1)$

1. $V(1) + \phi = 3 + 0.5(V(1) + V(2)); V(2) + \phi = 1 + V(2)$: $(0, -4), 1$

2. $\tilde{\alpha}(2) = \arg\max_a \{1 + V(2), 0.5(V(1) + V(2))\} = \arg\max\{-3, -2\} = 2$

3. $\alpha = (1,2)$ and goto 1

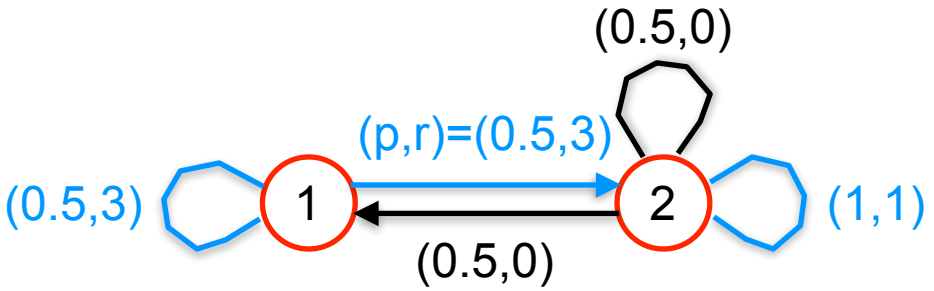
1. $V(1) + \phi = 3 + 0.5(V(1) + V(2)); V(2) + \phi = 0.5(V(1) + V(2))$:
$(0, -3), 1.5$

2. $\tilde{\alpha}(2) = \arg\max_a \{1 + V(2), 0.5(V(1) + V(2))\} = \arg\max\{-2, -1.5\} = 2$

3. $\alpha = (1,2)$ and stop

Bellman equation

$$V + \phi e = \max_{\alpha} \{ r_\alpha + P_\alpha V \}$$

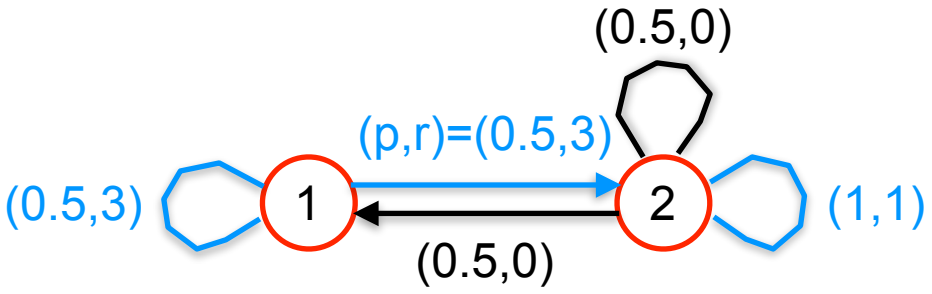


$(0, -3), 1.5$ indeed solution of

$$V(1) + \phi = 3 + 0.5(V(1) + V(2))$$
$$V(2) + \phi = \max\{1 + V(2), 0.5(V(1) + V(2))\}$$

Value iteration

$$V_{t+1} = \max_{\alpha}\{r_\alpha + P_\alpha V_t\}$$



iterate:

$$V_{t+1}(1) = 3 + 0.5(V_t(1) + V_t(2))$$
$$V_{t+1}(2) = \max\{1 + V_t(2), 0.5(V_t(1) + V_t(2))\}$$

```
V=c(0,0)
P1=matrix(c(0.5,0.5,0,1),nrow=2,byrow=T)
P2=matrix(rep(0.5,4),nrow=2)
for(n in 1:100)V=pmax(c(3,1)+P1%*%V,c(3,0)+P2%*%V)
a=which.max(c(c(3,1)[2]+P1[2,]%*%V,c(3,0)[2]+P2[2,]%*%V))
V-min(V);if(a==1){c(3,1)+P1%*%V-V}else{c(3,0)+P2%*%V-V}
```

```
     [,1]
[1,]    3
[2,]    0
     [,1]
[1,]  1.5
[2,]  1.5
```

Assignment 2

- Implement the following exercise in python/C/C++ using standard functions and packages (no MDC packages)
- You keep inventory of 2 different items with the following features and parameters:
  - Every time unit (a day) the demand of each item can be either 0 or 1, both with probability 0.5, independent of each other
  - You can order any combination of the 2 items
  - Orders arrive by the end of the day, and have a fixed cost of 5
  - Holding costs are 1 per item & unit of time for the first item, 2 for the second item
  - Lost sales/backorders should be avoided, thus you should order an item when its inventory is 1
  - The shelf space = maximum inventory level of each items is 20

a) What is an appropriate state and action space? Does the action set depend on the state?

- Consider the following order policy: when inventory of one of the items is 1, then you order such that the inventory level of both items becomes 5 (without the sales for that day)

b) Give a list of all possible transitions and their probabilities and simulate the system for a long period to determine the long-run average costs

c) Compute iteratively the stationary distribution and use this to find the long-run average costs as well.

d) Solve the average-cost Poisson equation using backward recursion (hint: b, c & d should give the same answer)

e) Now we are looking for the optimal policy. You can assume that you only order if one of the items has inventory level 1. What are the minimal average costs? Find it using value iteration

f) Characterize the optimal policy

- Same submission rules as assignment 1

Answers to polls