Dynamic Programming & Reinforcement Learning

Lecture 4
The Poisson equation

Ger Koole

Markov decision chains

- Rewards: Visiting state x and choosing action a gives immediate reward $r(x,a)$
- Transitions: $p(y|x,a)$
- Objective: Maximise expected long-term average reward ϕ_* = expected limiting reward
- Policies: $\alpha(a|x)$ = probability of choosing a in x
- Then $p(y|x) = \sum_a \alpha(a|x)p(y|x,a)$ (law of total probability)
- and thus

$$\sum_x \pi_*(x)p(y|x) = \pi_*(y) \Leftrightarrow$$

$$\sum_{x,a} \pi_*(x)\alpha(a|x)p(y|x,a) = \sum_a \pi_*(y)\alpha(a|y)$$

- Define $\pi_*(x,a) = \pi_*(x)\alpha(a|x)$ = "state-action frequency"
- Then $\sum_x \pi_*(x)p(y|x) = \pi_*(y) \Leftrightarrow \sum_{x,a} \pi_*(x,a)p(y|x,a) = \sum_a \pi_*(y,a)$
- Also $\phi_* = \sum_{x,a} \pi_*(x)\alpha(a|x)r(x,a) = \sum_{x,a} \pi_*(x,a)r(x,a)$

Markov decision chains

We have:

$$\sum_{x,a} \pi_*(x,a)p(y|x,a) = \sum_a \pi_*(y,a)$$

$$\phi_* = \sum_{x,a} \pi_*(x,a)r(x,a)$$

Optimization:

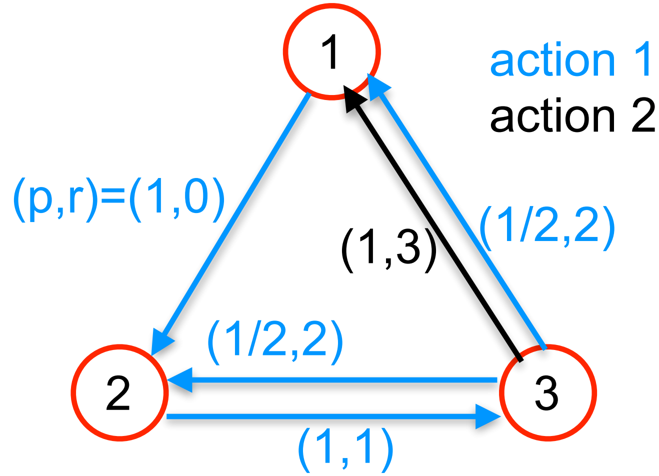
$$\max \sum_{x,a} \pi_*(x,a)r(x,a) \text{ s.t.}$$

$$\sum_{x,a} \pi_*(x,a)p(y|x,a) = \sum_a \pi_*(y,a),\ \sum_{x,a} \pi_*(x,a)=1 \text{ and } \pi_*(x,a)\geq 0$$

Optimal policy: $\alpha_*(a|x) = \pi_*(x,a) / \pi_*(x) = \pi_*(x,a) / \sum_a \pi_*(x,a)$

Typically 1 action has >0 probability per state (because of # of constraints in linear optimization)

Example



- Linear programming formulation:

$$\max\ \pi^*(2,1) + 2\pi^*(3,1) + 3\pi^*(3,2) \text{ s.t.}$$

$$0.5\pi^*(3,1) + \pi^*(3,2) = \pi^*(1,1)$$

$$\pi^*(1,1) + 0.5\pi^*(3,1) = \pi^*(2,1)$$

$$\pi^*(2,1) = \pi^*(3,1) + \pi^*(3,2)$$

$$\sum_{x,a} \pi^*(x,a)=1,\ \pi^*(x,a)\geq 0\ \forall x,a$$

- Optimal solution $(\pi^*(1,1),\pi^*(2,1),\pi^*(3,1),\pi^*(3,2))$?

Poll!

Example

- Using CPLEX at neos-server.org
- Formulated in AMPL

INPUT

LP.mod

```
var pi11 >= 0;
var pi21 >= 0;
var pi31 >= 0;
var pi32 >= 0;

maximize phi: pi21 + 2*pi31 + 3*pi32;

subject to flow_state1:
   0.5*pi31+pi32=pi11;
subject to flow_state2:
   pi11+0.5*pi31=pi21;
subject to distribution:
   pi11+pi21+pi31+pi32=1;
```

LP.run

```
solve;
display pi11, pi21, pi31, pi32;
```

OUTPUT

```
CPLEX 12.10.0.0: threads=4
CPLEX 12.10.0.0: optimal solution; objective 1.333333333
0 dual simplex iterations (0 in phase I)
pi11 = 0.333333
pi21 = 0.333333
pi31 = 0
pi32 = 0.333333
```

Disadvantages LP approach

- Computationally slow $\Rightarrow$ only small instances
- High memory requirements
  - A variable for all state-action pairs

- Next lecture: approach based on value functions

Poisson equation

- Back to Markov reward chains
- Find backward way to compute

ϕ_* = long-term average reward

- using

$V_t(x)$ = total expected reward over t time starting in x

- Needed $o(f(t))$ ("small order of f"):
- $g(t)=o(f(t))$ if $g(t)/f(t) \to 0$ as $t \to \infty$
- Example/main usage: $g(t) = o(1)$ if $\lim_{t\to\infty} g(t) = 0$
- Looks useless but is convenient

Poisson equation

$r(x) < \qquad V_t(y) \qquad >$

$< \qquad V_t(x) \qquad > \phi_*$

$t+1 \; t \qquad \ldots\ldots\ldots \qquad 2 \;\; 1 \;\; 0$

time $\rightarrow$

We have

$$V_{t+1}(x) = r(x) + \sum_y p(y\,|\,x) V_t(y)$$

but also

$$V_{t+1}(x) = V_t(x) + \sum_y \pi_t(y) r(y) = V_t(x) + \phi_* + o(1)$$

because $\pi_t \rightarrow \pi_*$ and $\phi_* = \, < \pi_*, r > \, = \pi_*^T r = \sum_y \pi_*(y) r(y)$

Combined:

$$V_t(x) + \phi_* + o(1) = r(x) + \sum_y p(y\,|\,x) V_t(y)$$

- What if $t \rightarrow \infty$?
- Does not work: V_t does not converge

Poisson equation (2)

- Solution: $V_*(x) = \lim_{t\to\infty} (V_t(x) - \phi_* t)$
- V_* = total difference between starting in x or in stationarity, |V|<∞
- Now substract $\phi_* t$ and let t→∞ to get:

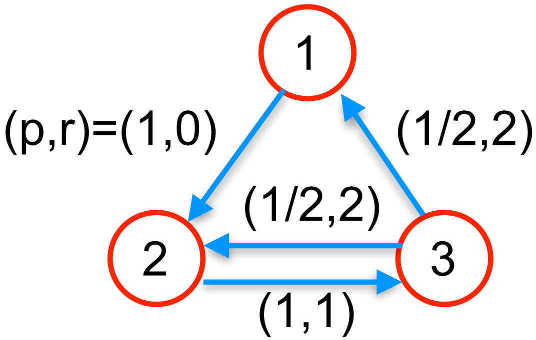
$$V_*(x) + \phi_* = r(x) + \sum_y p(y\,|\,x) V_*(y)$$

Poisson equation!

$$\Leftrightarrow V_* + \phi_* e = r + PV_*$$

- However: equation $V + \phi e = r + PV$ has no unique solution, $(V_* + ce, \phi_*)\forall c$ are all solutions
- Require also $< \pi_*, V_* > = 0$
- But: any solution to $V + \phi e = r + PV$ gives ϕ_*, unique solution by requiring $V(0) = 0$, no need to determine π_*

Example

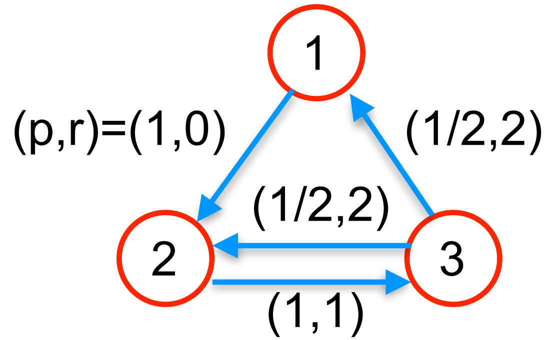


- What is the state with lowest V_*? Poll!
- Poisson equation:

$$V(1) + \phi = 0 + V(2)$$
$$V(2) + \phi = 1 + V(3)$$
$$V(3) + \phi = 2 + 0.5(V(1) + V(2))$$

- Solution? Poll!

Example



- Rewrite, $V(1) = 0$:

$$-V(2) + \phi = 0$$
$$V(2) - V(3) + \phi = 1$$
$$-0.5V(2) + V(3) + \phi = 2$$

- Solution?

```
library(expm)
A=matrix(c(-1,0,1,1,-1,1,-0.5,1,1),nrow=3,byrow=TRUE)
solve(A,c(0,1,2))
```

- $(V(2), V(3), \phi) = (1.2, 1.4, 1.2)$

Recursion

- $V_t - \min\{V_t\}e$ converges to solution of $V + \phi e = r + PV$:

```
V=c(0,0,0);P=matrix(c(0,1,0,0,0,1,0.5,0.5,0),nrow=3,byrow=TRUE)
for(n in 1:100)V=c(0,1,2)+P%*%V
V-min(V);c(0,1,2)+P%*%V-V
```

- Answer:

```
     [,1]
[1,]  0.0
[2,]  1.2
[3,]  1.4
     [,1]
[1,]  1.2
[2,]  1.2
[3,]  1.2
```

V with V(1)=0

ϕe

Backward recursion: convergence

- $V_{t+1} = r + PV_t$:

$$V_{t+1} - V_t \to \phi_* e$$

$$V_t(x) - V_t(y) \to V_*(x) - V_*(y)$$

- Convergence criterium:

iterate until

$$\text{span}\{V_{t+1} - V_t\} \leq \epsilon$$

where span $v = \max v_i - \min v_i$ for $v \in R^n$

Backward recursion

- Backward recursion algorithm:

Step 0: Take V_0 arbitrary

Step 1: Iterate $V_{t+1} = r + PV_t$ until

$$\text{span}\{V_{t+1} - V_t\} \leq \epsilon$$

- Theorem: The algorithm terminates with

$$(\phi_* - \epsilon)e \leq V_{t+1} - V_t \leq (\phi_* + \epsilon)e$$

Sidestep: Deviation matrix

- Take r = e_y
- Then $V_*(x)$ = number of visits to y from x compared to stationarity = D(x,y) = "deviation matrix"
- Π = matrix with all rows equals to $\pi_* = e\pi_*^T$
- Poisson equation for all e_x: $D + \Pi = I + PD$
- Poisson equation special case with Dr=V:

$$V_* + \phi_* e = Dr + \Pi r = Ir + PDr = r + PV_*$$

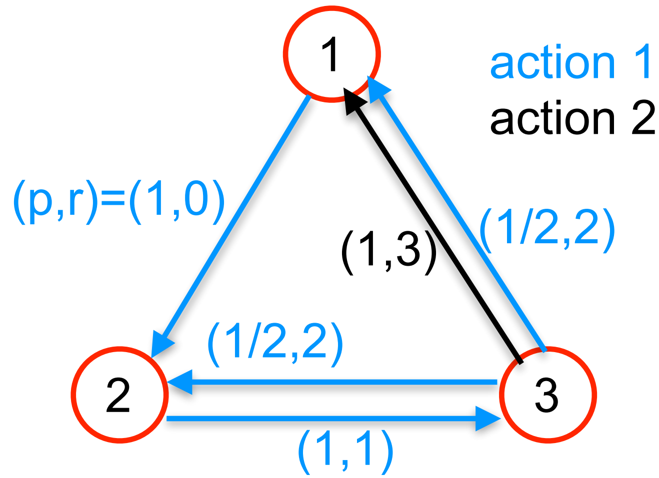
Markov decision chains: policy iteration (PI)

- Define:
  - $r_\alpha(x) = r(x, \alpha(x))$
  - $P_\alpha(x, y) = p(y \,|\, x, \alpha(x))$
- 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?

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

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

Proof: Define $J_\alpha V = r_\alpha + P_\alpha V$. Then

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

This argument can be repeated for any t

But then

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

Answers to polls

Polls

the optimum

1. What is the optimal solution of the LP formulation? (Single Choice) *

- ( ) (0.2,0.4,0.4,0)
- ( ) (0.33,0.33,0.33,0)
- (•) (0.33,0.33,0,0.33)
- ( ) (0,0,0,1)

Polls

Poisson equation

1. state with lowest V*? (Single Choice) *

- (•) 1
- ( ) 2
- ( ) 3

2. Solution Poisson equation (V,phi)? (Multiple Choice) *

- [x] (0,1.2,1.4,1.2)
- [ ] (1,1,4,18,1.4)
- [x] (1,2.2,2.4,1.2)
- [ ] (0,1,1.2,1)