CS 5100 02/13

Review from last time

* Markov Decision Process
  - Decision problem

(transition model is given by transition probabilities which are Markovian)

- 0.8 proba to go to desired destination
- 0.2 probability other perpendicular locations

reward = −0.04 at all states besides the terminal one

NNEEE

$$\boxed{U(1,1)} = -0.04 + 0.8\,U(1,2) + 0.1\,U(1,1) + 0.1\,U(2,1)$$

Need!

solve a system of equations

matrix and vectors

$$x + y = 2$$
$$x - z = 3$$
$$2x + y + 3z = 1$$

$$\overset{A}{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 2 & 1 & 3 \end{pmatrix}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \overset{\vec{b}}{\begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}}$$

np.linalg.solve$(A, \vec{b})$ $\rightarrow$ solution vector $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$

limitations : solve a linear system for every path that I take (NNEEE, ...) (too many of them)

solve a linear system with n equations and n unknowns costs $O(n^3)$

→ expensive computationally

Algorithm : Find the optimal utilities without having to solve many linear systems.

$$U(s) = R(s) + \sum_{s' \in succ(s)} P(s' \mid s,a) \cdot U(s')$$

Find the optimal utility (maximizing my utility)

$$\boxed{U(s) = R(s) + \max_{a} \sum_{s' \in succ(s)} P(s'|s,a) \cdot U(s')}$$

Bellman equation

write such an equation for every state in the problem.

Harder to solve

one system of non linear equations

(because of the max in the expression)

Iterative approach

start with initial values

$U_0(s_1), U_0(s_2), \ldots, U_0(s_n)$

rewards at every state are given

Update the values of the utilities according to some update rule.

$$U_{i+1}(s_1) = g(U_i(s_1))$$

$$U_{i+1}(s) = R(s) + \max_a \sum_{s' \in sna(s)} p(s' | s, a) U_i(s')$$

$a_0 = 1$

$a_{i+1} = 2a_i + 1$

* How do I compute it?
* Will it converge to some value?
* Will it converge to the optimal values?

10 5 2.5 1.25 0.625 0.3

. . . . . 0.1 0.005 0.0001 . . .

criteria :

if the values of the utilities at different iterations are barely changing, then I am converging.

$$U_{i+1}(s) = \underbrace{R(s) + \max_{a} \sum_{s' \in \text{succ}(s)} P(s'|s,a)\, U_i(s')}_{g(U_i(s))}$$

U_{old} 1

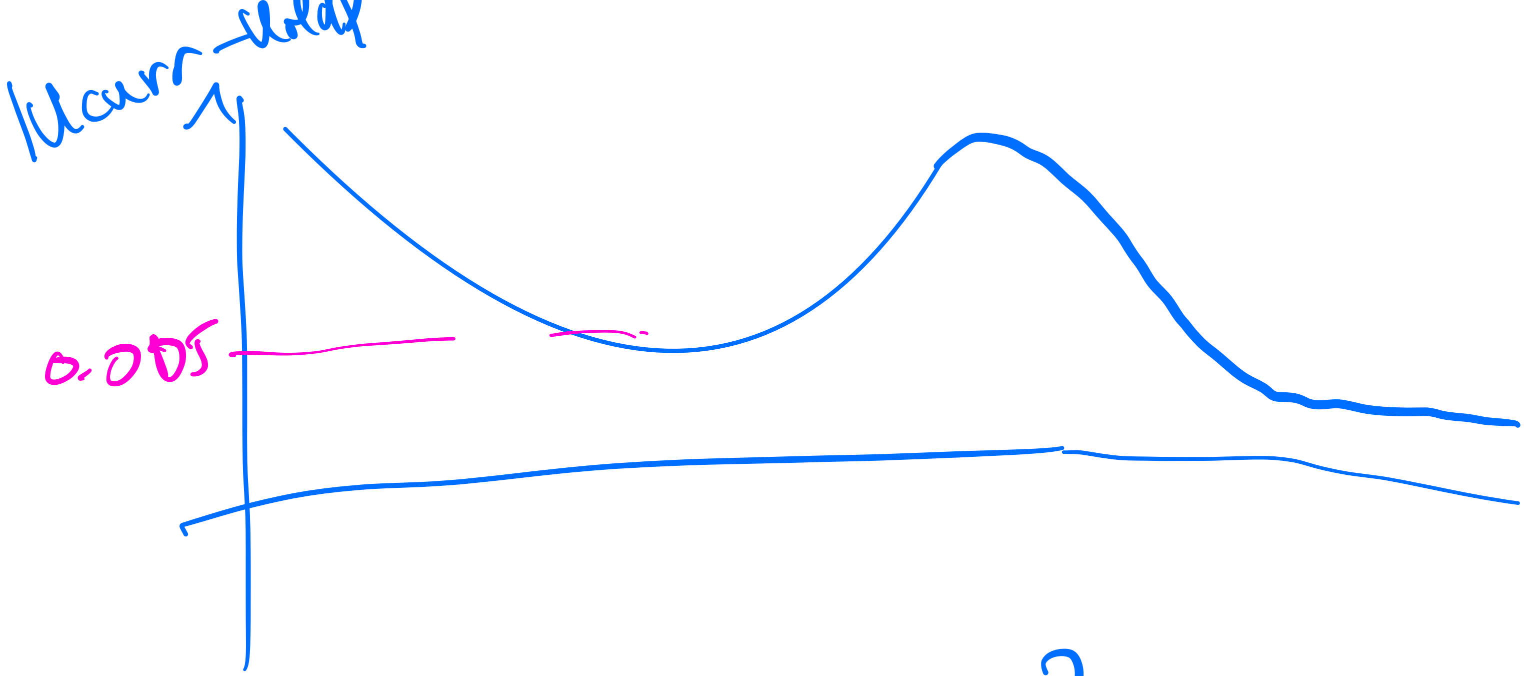
U_{curr} 1/2

while ($|U_{curr} - U_{old}| >$ ~~0.01~~ 0.0001):

$U_{old} = U_{curr}$

$U_{curr} = g(U_{curr})$

I know how to check whether the algorithm converged



* Does it always converge?
* Will it converge to the correct solution?

$$\cancel{U_{i+1}(s) = \underbrace{R(s) + \max_a \sum_{s' \in succ(s)} P(s'|s,a)\, U_i(s')}_{g(U_i(s))}}$$

U_0^{old}

$$\vec{U_0}(s) = \begin{bmatrix} U_0(s_1) \\ U_0(s_2) \\ \vdots \\ U_0(s_n) \end{bmatrix} \qquad g(\vec{U_0}(s)) = \begin{bmatrix} R(s_1) + \max_a \sum_{s'} P(s'|s,a) \cdot U_0(s') \\ \vdots \\ R(s_n) + \max_a \sum_{s'} P(s'|s_n,a)\, U_0(s') \end{bmatrix}$$

$$g(\vec{U_0}(s)) = \vec{U_1}(s)$$

$$U_{i+1}(s_j) = R(s_j) + \max_a \sum_{s'} P(s' \mid s_j, a)\underline{\underline{U_i(s')}}$$

In general

$$\boxed{\overrightarrow{U_{i+1}(s)} = g\left(\overrightarrow{U_i(s)}\right)}$$

$\overrightarrow{U_{i+1}(s)}$ is very close to $\overrightarrow{U_i(s)}$

Special Function

$$U(s) = R(s) + \max_a \sum_{s'} P(s' \mid s, a)\, U(s')$$

Bellman equation

$$\vec{U} = g(\vec{U})$$

want to converge to this equation

algorithm

$$\boxed{\overrightarrow{U_{i+1}(s)} = g\left(\overrightarrow{U_i(s)}\right)} \quad \text{vs} \quad \boxed{\vec{U} = g(\vec{U})}$$

one of the solutions is the optimal U

if $\boxed{\overrightarrow{U_i(s)} \approx \overrightarrow{U_{i+1}(s)}}$

$$\boxed{\overrightarrow{U_{i+1}(s)} \approx g\left(\overrightarrow{U_{i+1}(s)}\right)}$$

this tells me that $\overrightarrow{U_{i+1}(s)}$ is converging to a solution of $\vec{U} = g(\vec{U})$

As long $\vec{U} = g(\vec{U})$ only has one solution, we're guaranteed that $\vec{U}_{i+1}(s)$ is getting closer to the optimal solution.

Let's dive into this equation $\boxed{\vec{U} = g(\vec{U})}$ to understand how many solutions it has and specifically we want to show that it has only one solution

Bellman equation

We can show that

two utility vectors

$\underline{\vec{U}} - \underline{\vec{U}'}$

$\|\vec{x}\|_2$

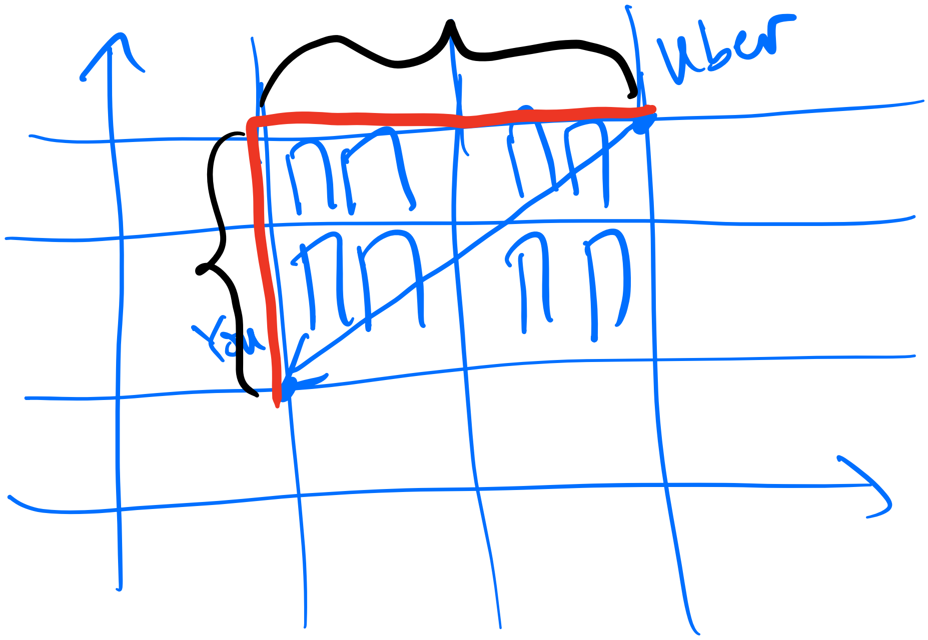
$\|\vec{U} - \vec{U}'\| \approx 0$

2-norm $\quad$ 1-norm $\quad$ inf-norm

$\sqrt{x_1^2 + \dots + x_n^2} = \|\vec{x}\|_2, \quad \|\vec{x}\|_1, \quad \|\vec{x}\|_\infty$

$|x_1| + \dots + |x_n| \quad \max |x_i|$

pythagorean



$\to U_i(s_1) \longrightarrow U(s_1)$ $\qquad U_i(s_1) \approx U(s_1)$

$\to U_i(s_2) \longrightarrow U(s_2)$ $\qquad U_i(s_2) \approx U(s_2)$

$\vdots$

$\to U_i(s_n) \longrightarrow U(s_n)$ $\qquad U_i(s_n) \approx U(s_n)$

$$\underbrace{\underbrace{|U_i(s_1) - U(s_1)|}_{\text{small}} + \underbrace{\boxed{|U_i(s_2) - U(s_2)|}}_{\text{not too small}} + \dots + \underbrace{|U_i(s_n) - U(s_n)|}_{\text{small}}}_{\text{pretty small}}$$

1000 iteration

$|V_{1000}(S_1) - V(S_1)|$ small

$|V_{1000}(S_2) - V(S_2)|$ not small (largest)

$|U_{1000}(S_3) - U(S_3)|$ not small

$|U_{1000}(S_4) - U(S_4)|$ small

$|V_{1000}$ small

$\vdots$

$|U_{1001}(S_1) - U(S_1)|$ small

$|U_{1001}(S_2) - U(S_2)|$ small

$|U_{1001}(S_3) - U(S_3)|$ not small

$\vdots$ small

$\vdots$

Infinity norm is a nice option to use in this problem.

Result

$$\|g(\tilde{U}) - g(\vec{U}')\|_\infty \le \|\vec{\tilde{U}} - \vec{U}'\|_\infty$$

g is a contraction

true for any utility vectors $\vec{\tilde{U}}, \vec{U}'$

then apply result on $\vec{U}_{i+1}$ and $\vec{U}_i$

$$\|g(\vec{U}_{i+1}) - g(\vec{U}_i)\|_\infty \le \|\vec{U}_{i+1} - \vec{U}_i\|_\infty$$

$$\|$$

$$\|\vec{U}_{i+2} - \vec{U}_{i+1}\|_\infty \le \|\vec{U}_{i+1} - \vec{U}_i\|_\infty$$

contraction

$$\boxed{\|g(\vec{\hat{U}}) - g(\vec{U}')\|_\infty \le \|\vec{\hat{U}} - \vec{U}'\|_\infty}$$

A contraction only has one solution for $\boxed{g(U) = U}$

Bellman

$U(s) = R(s) + \max \sum U$

$U_1 = R(s) + \max \sum U_1$

Assume by contradiction there are two solutions U_1, U_2 such that

$U_2 = R(s) + \max \sum U_2$

$g(U_1) = U_1$ and $g(U_2) = U_2$

such that $U_1 \neq U_2$

contraction tells me that

$$\underbrace{\| g(\vec{U_1}) - g(\vec{U_2}) \|_\infty} \leq \| \vec{U_1} - \vec{U_2} \|_\infty$$

$$\Rightarrow \| \vec{U_1} - \vec{U_2} \|_\infty \leq \underbrace{\| g(\vec{U_1}) - g(\vec{U_2}) \|_\infty}$$

$\Rightarrow$

$$\underbrace{\| \vec{U_1} - \vec{U_2} \|_\infty} \leq \| g(\vec{U_1}) - g(\vec{U_2}) \|_\infty \leq \underbrace{\| \vec{U_1} - \vec{U_2} \|_\infty}$$

Need to have $\| U_1 - U_2 \|_\infty = \| g(U_1) - g(U_2) \|_\infty$