Section 2

Function approximation

Supervised learning $y = f(x)$, find the function to approximate to map $x \to y$

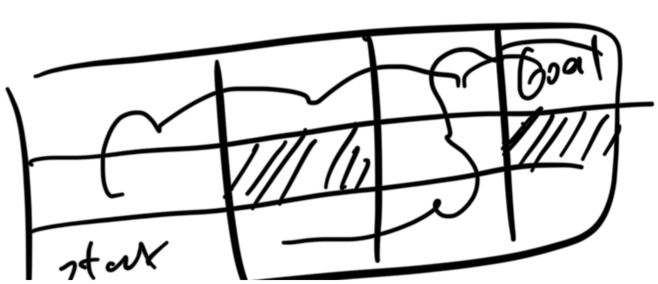
Unsupervised learning: Clustering

Using x, find function

Reinforcement learning:

We're going to be given X and Z,

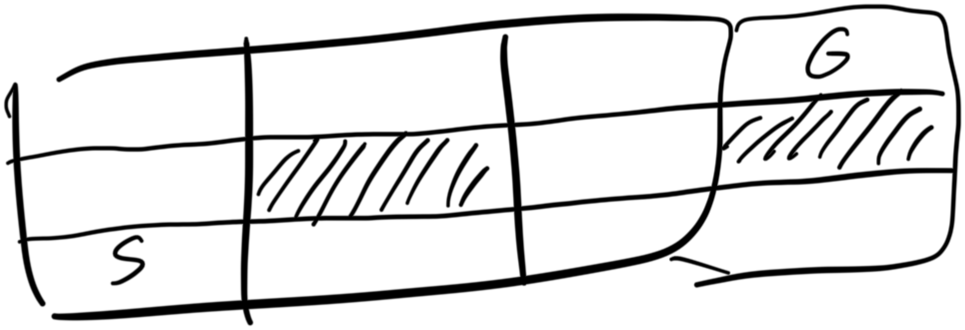
find f that generates y's.



Shortest Path

RRUUR

V V R R R



Actions executes (0.8)

Move at angle to left or right (0.2)

Reliability for U U R R R

$\boxed{.32776}$

$.8^5 = .32768$

$.1^4 \cdot .8 = .00008$

$+$

0.32776

4 often go wrong but 1 goes right

Markov Decision Process | Markovian property:

States: S

only present matters
rules are stationary

transition function

Model $T(s,a,s') \sim Pr(s' | s,a)$

Action: $A(s)$, A

Reward: $R(s)$, $R(s,a)$, $R(s,a,s')$

Policy: $\pi(s) \to a$, π^* ↳ optimal / max reward

↳ (command, order)

$\langle s, a, r \rangle$

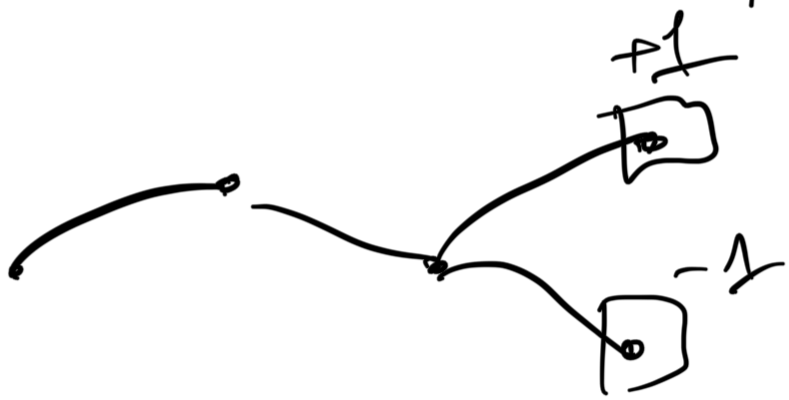
$R : z$ for $y : f(x, r)$

$f : \pi$

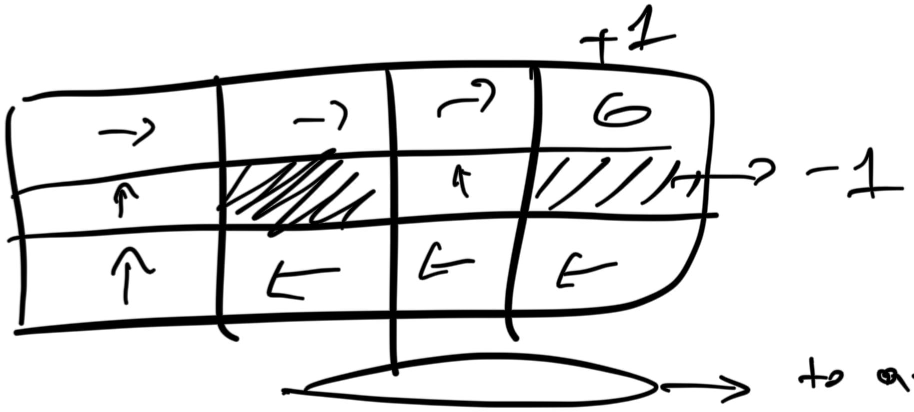
$y : a$

$x : s$

delayed reward, minor changes matter



Temporal Credit Assignment Problem



to avoid possibly going into -1.

$R(s)$: $-.04$

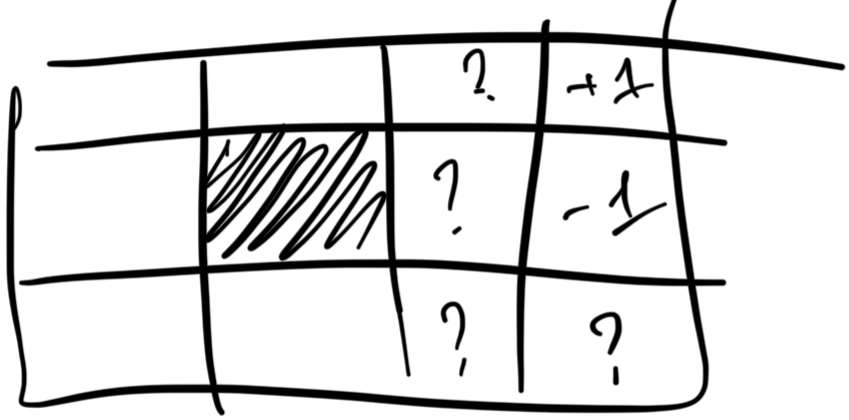
(except for goal +1 or trail -1)

Quiz

$R(s)$: $+2$

←

←



$R(s) = -2$

Sequence of Rewards

- Infinite Horizons

- Utility of sequences

$$U(s_0, s_1 \dots s_n) = \sum_{t=0}^{\infty} R(s_t)$$

quiz:

+1 +1 +1 +1 +1 +1 ... ∞

...
+1 +1 +2 +1 +2 ∞

neither is better. <u>sum</u> of utility is $\underline{\infty}$
for both.

if our rewards are always +1,
then our movement <u>doesn't really matter</u>.

$$\sum_{t=0}^{\infty} \gamma^t \cdot R(S_t) \qquad 0 \leq \gamma < 1$$

rewards get <u>smaller</u> over time. if it's
zero, only
<u>immediate</u>
rewards

$$\frac{R_{max}}{1-\gamma} = \sum_{t=0}^{\infty} \gamma^t \cdot R_{max}$$

discounted sums.

discounted → geometric
infinite → finite

$\left(\sum \gamma^t\right) R_{max}$

$x = \left(\gamma^0 + \gamma^1 + \gamma^2 + \gamma^3 \ldots\right)$

$x = \gamma^0 + \gamma \cdot x$

$x = \dfrac{1}{1-\gamma} \cdot R_{max}$

<u>Policies</u> : $\quad \underset{x}{\operatorname{argmax}} = \underbrace{E}_{\text{expected value}} \left[\sum_t \gamma^t R(s_t) \mid \pi \right]$

$\pi^* =$

$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi, s_0 = s \right]$

$\cancel{R(s)}$

Reward: immediate feedback

delayed feedback

Utility: long-term feedback

(Reward now, and reward of future)

$$\pi^* = \arg\max_a \sum_{s'} T(s,a,s') \cdot U(s')$$

$$U(s) \equiv U^{\pi^*}(s) \rightarrow \text{the utility of a state.}$$

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') \cdot U(s')$$

Bellman Equation

N of states

n equations, n unknowns.

max is non-linear

- start w/ arbitrary utilities

- update based on neighbors.
- repeat until convergence

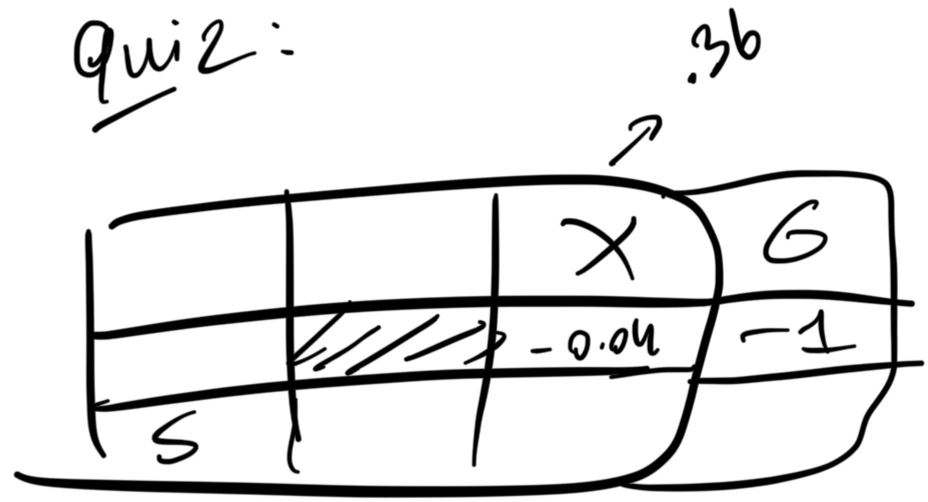
We use all values from U's from previous to update current.

$$\hat{U}_{t+1}(s) = \underbrace{R(s)}_{\to \text{truth}} + \gamma \cdot \max_a \sum_{s'} T(s,a,s') \cdot \hat{U}_t(s')$$

Value Iteration.

---

quiz:



$\gamma = 1/2$, $R(s) = -0.04$, $U_0(s) = 0$

$U_1(x) = 0$

we go to the right

$\cancel{U_1 x}$ $- .04 + \frac{1}{2} \cdot \left[0 + 0 + 0.8 \right] = \underline{\underline{.36}}$

$\cancel{U_2 x}$ $- 0.04 + \frac{1}{2} \left(0.36 + 0.8 - 0.04 \right) = \underline{\underline{0.376}}$

<u>finding policies:</u>

Start with π_0

evaluate given π_t calculate $U_t = U_t^{\pi}$.

Improve π_{t+1} : $\arg\max \sum T(s,a,s') \cdot U_t(s')$

$$U_t(s) = R(s) + \gamma \sum_{s'} T(s, \pi_t(s), s') \cdot U_t(s')$$

this is liner unlike max avg.

Policy iteration

$$V(s) = \max_a \left[R(s,a) + \gamma \cdot \sum_{s'} T(s,a,s') \cdot V(s') \right]$$

$$s \xrightarrow[a]{R(s,a)} s'$$

$$s_1 \quad a_1 \quad R(s_1,a_1) \quad s_2 \quad a_2 \quad R(s_2,a_2) \quad s_3$$

$$V(s_1) : \max_{a_1} \Big[R(s_1,a_1) + \gamma \cdot \sum_{s_2} \cdot T(s_1,a_1,s_2) \cdot$$

$$\max_{a_2} \Big[R(s_2,a_2) + \gamma \cdot \sum_{s_3} \cdot T(s_2,a_2,s_3) \ldots \Big] \Big]$$

$$q(s,a) : R(s,a) + \gamma \cdot \Sigma_{s_1} \cdot T(s,a,s') \cdot \max_{a'} \cdot q(s',a')$$

	V	q	C
V	$V(s) : V(s)$	$V(s) : \max_a q(s,a)$	$V(s) : \max_a \big(R(s,a) + C(s,a)\big)$
q	$q(s,a) : R(s,a) + \gamma \cdot \Sigma_{s_1} \cdot T(s,a,s') \, V(s')$	$q(s,a) : q(s,a)$	$q(s,a) : R(s,a) + C(s,a)$
C	$C(s,a) : \gamma \cdot \Sigma_{s_1} \cdot T(s,a,s') + V(s')$	$C(s,a) \; \gamma \cdot \Sigma_{s_1} T(s,a,s') \max_{a} q(s',a')$	$C(s,a) : C(s,a)$