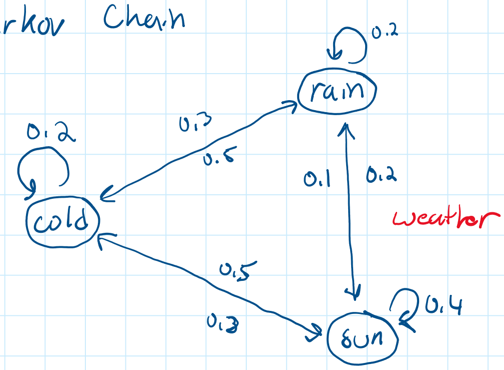
CS5080_Lecture_3

Tuesday, January 27, 2026 4:45 PM

Markov Chain



$\mathcal{S} = \{ cold, rain, sun \}$ STATES

Transitions from state to state

table T =

to \ from	COLD	RAIN	SUN	
cold	0.2	0.5	0.3	→ adds to 1
rain	0.5	0.3	0.2	
sun	0.5	0.1	0.4	

Markov property (simplifying assumption)

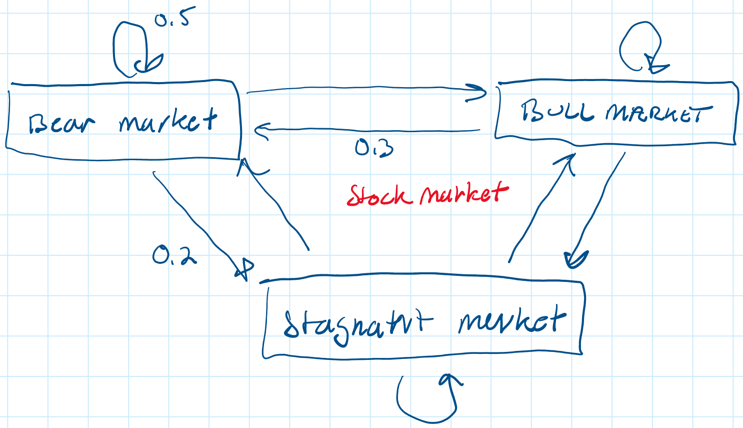
- The current state encapsulates all effects of history
- Next transition depends only on the current state

History of the chain

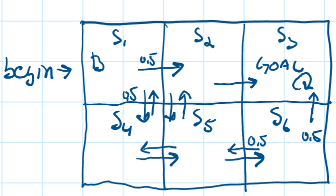
C R R R S S C C S . . .

here ? what will it be tomorrow

depends on current state only



"MAZE" or some game



Equi-probable actions in states

This in chap 2 of text

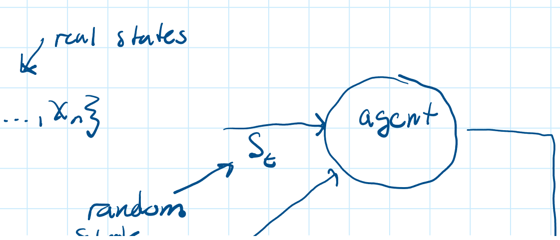
(FINITE)

Markov decision process (MDP)

- $\mathcal{S}$ = a set of states = $\{ x_1, x_2, \ldots, x_n \}$
- bring in a set of agents

$S_t \in \mathcal{S}$tates

$A_t \in \mathcal{A}$ctions

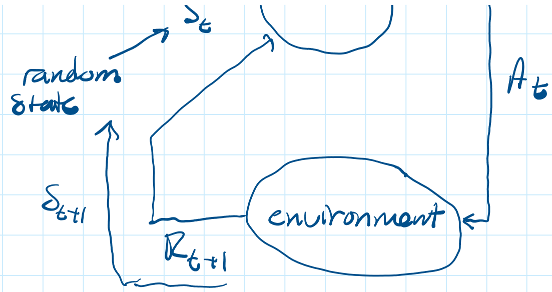


- bring in a set of agents

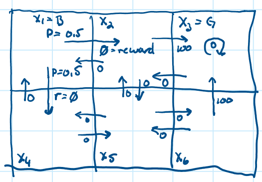
- agent can perform a set of actions $= \{a_1, a_2, \dots, a_m\}$

- Whenever an agent performs an action it gets an immediate reward R_{t+1}

  it transitions to state S_{t+1}



"MAZE" as MARKOV CHAIN → "MAZE" as MDP



100 immediate reward only on reaching goal state.

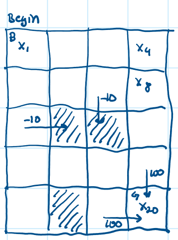
$S = \{x_1, \dots, x_6\}$

$A = \{\uparrow, \downarrow, \rightarrow, \leftarrow\}$

The agent performs an action A_t in state S_t to make the state transitions happen

example → chess, taking other pieces versus going for the win if rewards are given for taking pieces vs winning

A more complex maze

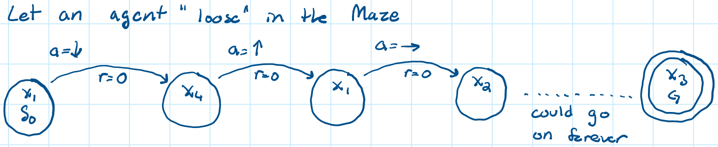


= Barrier state → -10 reward negative reward

given negative reward for entering a barrier state

The actual goal is agent starts in state B (or any other state) and transitions to state G most efficently

Let an agent "loose" in the Maze



This is call an episode of activity

Episodic environment

Tradjectory of agents activities

writing an episode text description

$x_1 \downarrow 0 \quad x_4 \uparrow 0 \quad x_1$ ......

$S_0\ A_0\ R_1\ S_1\ A_1\ R_2\ S_2$ .... State → Action → Reward

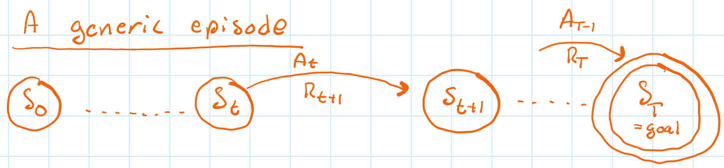
Policy (a good policy to be learned by an agent)

describes what action should be performed by the agent in which state

Time stamps need not be equally spaced

agent learns an "optimal" policy by training
at the end state it needs to compute what it did and learn

A generic episode



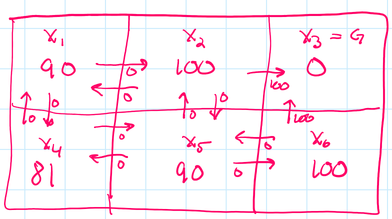
reinforment learning usually involves learning two values:

1) Values of a state $V(S)$, $S \in$ States
2) Value of agent perform action A in state S
   $q(S, A)$

Some algorithms try to learn both
$V(S)$ and $q(S, A)$

RL Algorthm type 1

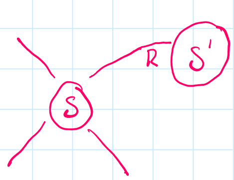
Suppose we have an RL algor that learns values of states thus



$V(S)$

based on these $V(S)$ values can we come up with a policy?

How to obtain a policy from $V(S)$ values

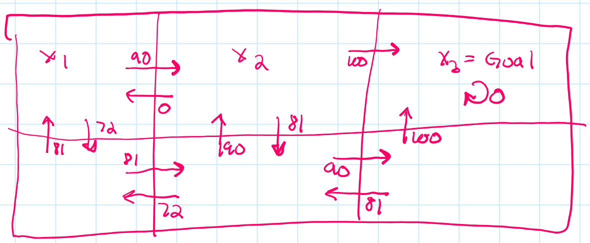


From S go to neighbor state

From S go to neighbor state
where $R' + V(s')$ is best
break ties randomly

RL Algorithm type 2

Suppose we have an algor that gives us the following $q(S,A)$ values



values of actions
not immediate rewards

Can we create a policy from these $q(S,A)$ values?

Actor – Critic methods

compute both $V(s)$ and $q(S,A)$