31.3.2015

Optimal and Learning Control for Autonomous Robots

Lecture 7

ADRL

Jonas Buchli
Agile & Dexterous Robotics Lab

FNSNF

ETH Zürich

Erratum Script

p14

$$\frac{dV^*}{dt} = V_t^* + V_{\mathbf{x}}^{*T}\mathbf{f} + \frac{1}{2}\mathrm{Tr}\left[V_{\mathbf{xx}}^* E[(\mathbf{f}+\mathbf{Bw})(\mathbf{f}+\mathbf{Bw})^T]\Delta t\right]. \quad (1.55)$$

p28

$$-\mathbf{x}^T\dot{\mathbf{S}}(t)\mathbf{x} = \min_{u\in U}\{\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} + 2\mathbf{u}^T\mathbf{P}\mathbf{x} + 2\mathbf{x}^T\mathbf{S}(t)\mathbf{A}\mathbf{x} + 2\mathbf{x}^T\mathbf{S}(t)\mathbf{B}\mathbf{u}\}. \quad (1.105)$$

Equations (1.27), (1.28), (1.29):

$$V^\mu(n,\mathbf{x}) = L_n(\mathbf{x},\mathbf{u}_n) + \alpha E_{x'\sim P_f(.|\mathbf{x},\mathbf{u}_n)}\left[V^\mu\left(n+1,\mathbf{x}'\right)\right] \quad (1.27)$$

$$V^*(n,\mathbf{x}) = \min_{\mathbf{u}}\left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha E_{\mathbf{x}'\sim P_f(\cdot|\mathbf{x},\mathbf{u}_n)}\left[V^*\left(n+1,\mathbf{x}'\right)\right]\right] \quad (1.28)$$

$$\mathbf{u}^*(n,\mathbf{x}) = \arg\min_{\mathbf{u}_n}\left[L_n(\mathbf{x},\mathbf{u}_n) + \alpha E_{\mathbf{x}'\sim P_f(\cdot|\mathbf{x},\mathbf{u}_n)}\left[V^*\left(n+1,\mathbf{x}'\right)\right]\right]. \quad (1.29)$$

Equations (1.143), (1.145):

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}(\mathbf{P}+\mathbf{B}^T\mathbf{S})\mathbf{x} \quad (1.143)$$

$$\mathbf{u}^*(\mathbf{x}) = -\mathbf{R}^{-1}(\mathbf{P}+\mathbf{B}^T\mathbf{S})\mathbf{x} \quad (1.145)$$

Class logistics

Exercise 1

- Submission:
  - Code must be submitted through website form
  - NO EMAIL SUBMISSION!
  - submit by Wed, 15.4.2015
  - USE OFFICE HOURS FOR QUESTIONS!
- Interviews:
  - Interviews on Friday, 17.4.2015, all day
  - 10 min session/group
  - explain submitted code and answers
  - pass/fail grade given
  - Doodle link for sign up for interview will be given

Office hours:
Thu, 17:30-18:30
Room: ML J37.1

Interview slot poll

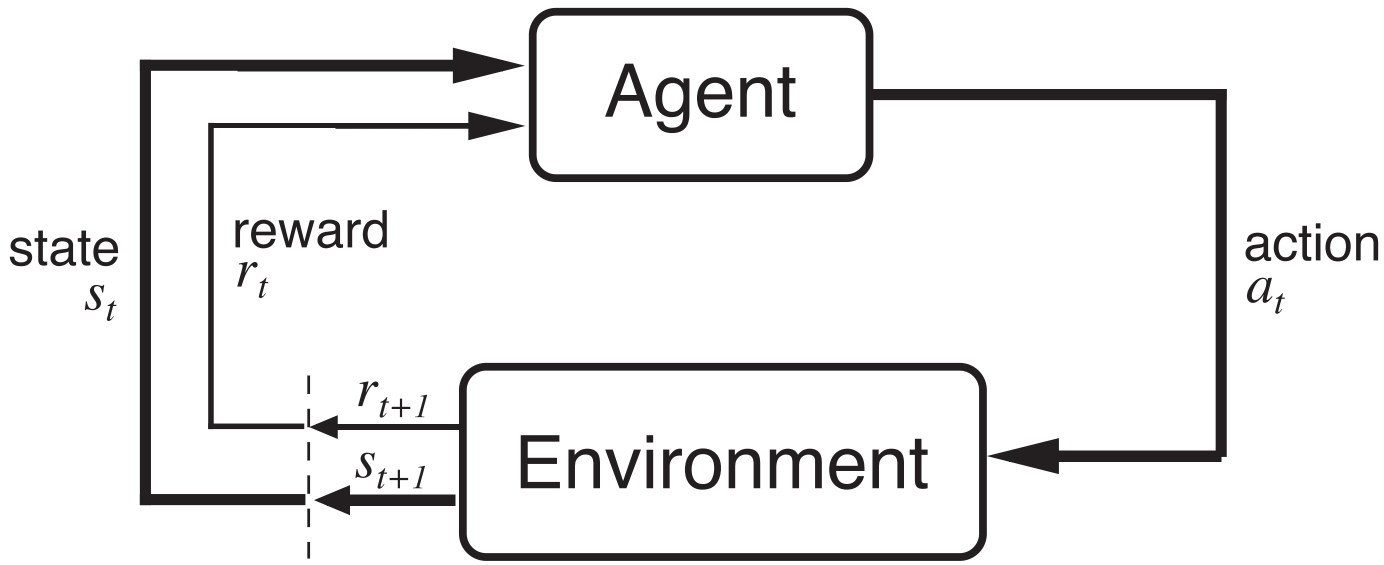
http://doodle.com/p2p2qi4vuhr3eean

Please sign up using your group ID!!!

Lecture 7 Goals

★Intro 'Classic' reinforcement learning
★Markov decision processes (MDP)
★State value function / action value function
★Policy iteration, evaluation, value iteration

The RL problem



Return:

$$R(t) = f(r(t, \dots t_e))$$

function of immediate and future reward

$$\max_{\pi} R$$

Policy: $\pi : s \to a$ can be stochastic

given state, rule how to pick action

Environment dynamics: $(s, a) \to s'$

Rewards $(s, a, s') \to r$

possibly stochastic, but observable

Given S,R,E find

$$\pi^* = \arg\max E[R]$$

Optimal policy ? 'good'

with only partial or no model of environment dynamics

Note: this lecture still *model-based* RL!

A D R L

ETH Zürich

Markov Property

Only need current state to predict next state

$$\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1})$$

Andrey Markov

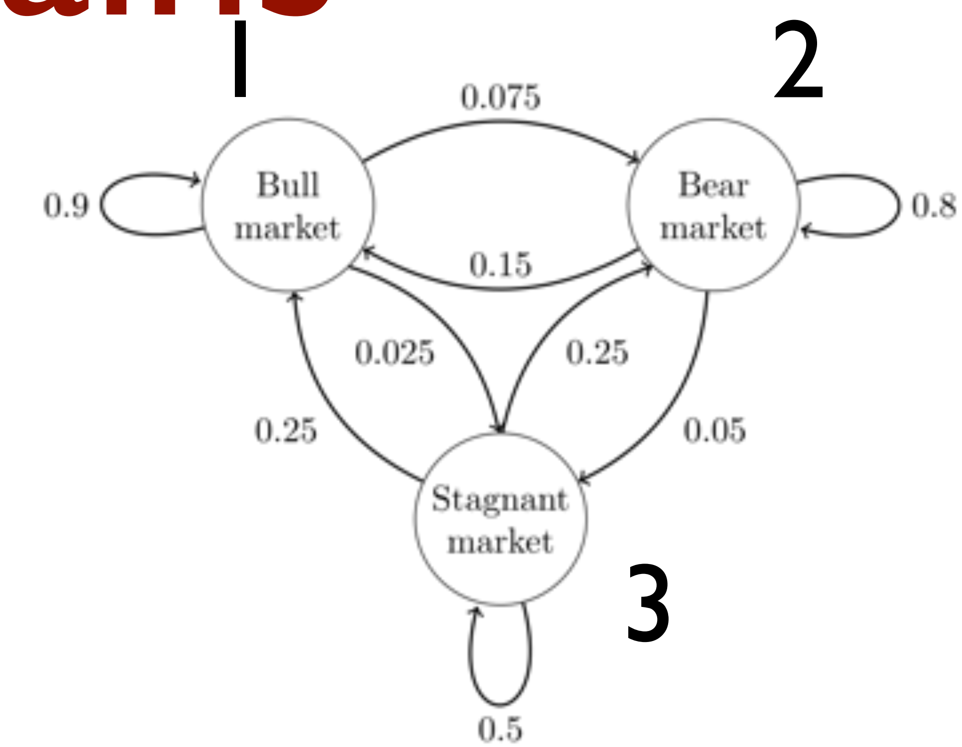
'Memoryless'

Any system with a finite memory can be described by a system with Markov property where the states are the original states + a delay line

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\}$$

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\}$$

Markov chains



'transition probability matrix'

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}.$$

Probabilities have to sum to 1

Find stationary solution by iteration:

$$\lim_{N\to\infty} P^N = \begin{bmatrix} 0.625 & 0.3125 & 0.0625 \\ 0.625 & 0.3125 & 0.0625 \\ 0.625 & 0.3125 & 0.0625 \end{bmatrix}$$

$$\begin{aligned} x^{(n+3)} &= x^{(n+2)}P = \left(x^{(n+1)}P\right)P \\ &= x^{(n+1)}P^2 = \left(x^{(n)}P^2\right)P \\ &= x^{(n)}P^3 \\ &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^3 \\ &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.7745 & 0.17875 & 0.04675 \\ 0.3575 & 0.56825 & 0.07425 \\ 0.4675 & 0.37125 & 0.16125 \end{bmatrix} \\ &= \begin{bmatrix} 0.3575 & 0.56825 & 0.07425 \end{bmatrix}. \end{aligned}$$

'Probabilities *diffuse* through the graph of possibilities'

Markov Decision Process

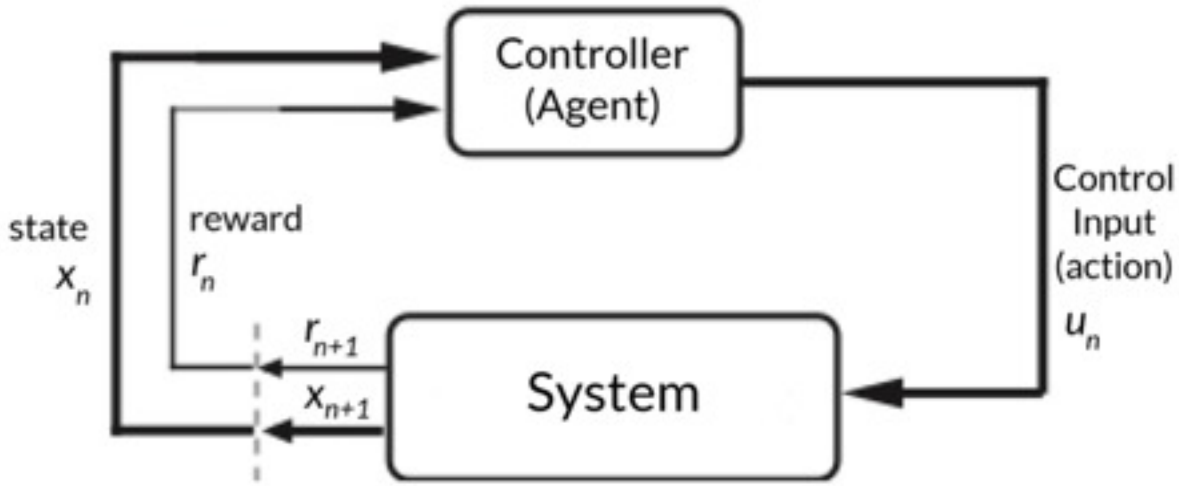
RL problem with Markov Property: MDP

Extension of Markov chains: addition of actions (allowing choice) and rewards (giving motivation)

reward dependent on current state, next state, action $R_a(s, s')$

RL	Optimal control
environment	system
agent	controller
state: s	state: x
control action: a	control action: u
reward: r	(negative) intermediate cost: L
discount factor: γ	discount factor α
stochastic policy: π	deterministic policy μ

MDP/RL



discrete state-action space

$$x \in \mathbf{X} = \{x_1, \ldots, x_n\}$$

$$u \in \mathbf{U} = \{u_1, \ldots, u_m\}$$

discrete time

$$t_n = \Delta t n$$

Policy: $\pi(x, u) \rightarrow$ stochastic $\pi(u|x)$

given state, rule how to pick action

Environment dynamics: $(x, u) \rightarrow x'$ stochastic: $P(x, u, x') = P(x_{n+1} = x' | x_n = x, u_n = u)$

transition probability

Rewards $(x, u, x') \rightarrow r$ $\quad r(x, u, x') = r(x_{n+1} = x', x_n = x, u_n = u)$

Return: $R_0 = r_0 + \alpha\, r_1 + \alpha^2\, r_2 + \cdots + \alpha^n r_n + \cdots = \sum_{k=0}^{\infty} \alpha^k\, r_k$

A D R L

$$\pi^* = \arg\max_\pi E[R_0]$$

Notation

Transition probability

$P(x, u, x') = P(x_{n+1} = x' | x_n = x, u_n = u)$

$$\mathcal{P}^u_{xx'} = Pr\{x_{n+1} = x' \mid x_n = x, u_n = u\}$$

Expected reward

$$\mathcal{R}^u_{xx'} = E\{r_n \mid x_{n+1} = x', u_n = u, x_n = x\}$$

reward function of next state, current controls, current state

a reward independent of next state $\mathcal{R}^u_x = \sum_{x'} \mathcal{P}^u_{xx'} \mathcal{R}^u_{xx'}$

expectation: weighted sum - 'sum over all possible next states'

Accumulated reward

Value: expected accumulated reward

Accumulated reward (with discount)

$$R_0 = r_0 + \alpha\, r_1 + \alpha^2\, r_2 + \cdots + \alpha^n r_n + \cdots = \sum_{k=0}^{\infty} \alpha^k\, r_k$$

Discount rate $\alpha \in [0, 1]$

Accumulated reward at time n $R_n = \sum_{k=0}^{\infty} \alpha^k\, r_{n+k}$

Optimal policy maximizes reward $\pi^* = \arg\max_\pi E[R_0]$

Remember: cost ~ -reward

The RL Problem
as optimal control problem

thus RL problem corresponds mathematically to infinite horizon discrete time optimal control problem

Differences to previous lectures (‘optimal control’)

Controller (policy): stochastic
Reward: more general probabilistic model
States & actions: discrete (but see L1, shortest path)

Important: None of these are fundamental differences! But...

model-free RL

Assume model of environment not or only very partially/locally known

model-based RL

Difference between RL and OC!

A D R L

ETH Zürich

Sample based RL?

Later!

aka model free RL

E.g. transition probability distributions not known, but have model to generate sample transitions

Transition probability distribution

$$\mathcal{P}_{xx'}^{u} = Pr\{x_{n+1} = x' \mid x_n = x, u_n = u\}$$

n

n x m

In continuous time: might have differential equation but don't know integral of this equation (e.g. non-integrable equations)

Value functions (again)

using Reward

to understand how to derive sample based RL (model free RL) need to understand the mathematical properties of the problem - understand Value functions

will let us develop sampling based RL later

State value function

expected accumulated reward starting at x and following the given policy π

$$V^{\pi}(x) = E\{R_n \mid x_n = x\} = E\left\{\sum_{k=0}^{\infty} \alpha^k r_{n+k} \mid x_n = x\right\}$$

Let's derive Bellman equation (again? because now we have stochastic reward)

Conditional probability distributions

Conditional probability distribution $X \sim P(x|y)$

random number drawn from such distribution is a function of condition $X = f(y)$

Expected value of conditioned random variable

$$E(X|y) = \sum_i P(X_i|y) X_i$$

$E(X|y) = f(y)$ is function of y

Marginal probability

$$P(X|y)$$

'marginalizing over y'

$$P(X) = \sum_i P(X|y_i)P(y_i)$$

Note relationship to expectation:

$$= \underset{y}{E}(P(X|y))$$

$$p(X) = \int_y p(X|y)p(y)dy$$

Conditional on several variables

$$P(X|z, y)$$

$$P(X|z) = \sum_i P(X|z, y_i)P(y_i)$$

Law of total expectation

$$\underset{X}{E}\{X\} = \underset{z}{E}\left\{\underset{X|z}{E}\{X \mid z\}\right\}$$

$$\underset{X}{E}\{X \mid y\} = \underset{z}{E}\left\{\underset{X|z}{E}\{X \mid z, y\} \mid y\right\}$$

use definition $E(x) = \sum_i P(x_i) x_i$

$$\underset{z}{E}\{\ldots, z\} = \sum_i P(\ldots, z_i) z_i$$

A D R L

ETH Zürich

Derivation of Bellman Equation for V (1)

$$V \quad V^{\pi}(x) = E\left\{ r_n + \alpha \sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1} \mid x_n = x \right\} : x \Big\}$$

E is linear

$$V^{\pi}(x) = E\left\{ r_n \mid x_n = x \right\} + E\left\{ \alpha \sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1} \mid x_n = x \right\}$$

using law of total expectation $\underset{X}{E}\{X \mid y\} = \underset{z}{E}\left\{ \underset{X|z}{E}\{X \mid z, y\} \mid y \right\}$

$$V^{\pi}(x) = E\left\{ r_n \mid x_n = x \right\}$$
$$+ E\left\{ E\left\{ E\left\{ \alpha \sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1} \mid x_{n+1} = x', u_n = u, x_n = x \right\} \mid u_n = u, x_n = x \right\} \mid x_n = x \right\}$$

u

x'

$\alpha^k \, r_{n+k+1}$

Derivation of Bellman Equation for V (2)

$$V^{\pi}(x) = E\left\{r_n \mid x_n = x\right\}$$
$$+ E\left\{E\left\{E\left\{\alpha \sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1} \mid x_{n+1} = x', u_n = u, x_n = x\right\} \mid u_n = u, x_n = x\right\} \mid x_n = x\right\}$$

Using Markov property: $\mathcal{R}^u_{xx'} = E\{r_n \mid x_{n+1} = x', u_n = u, x_n = x\}$

$$Pr\left\{r_{n+k+1} \mid x_{n+1} = x'\right\} = Pr\left\{r_{n+k+1} \mid x_{n+1} = x', u_n = u, x_n = x\right\}, \qquad \forall k \geq 0.$$

$$P\left(\sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1}\right) = f(x_{n+1})$$

$$V^{\pi}(x) = E\left\{r_n \mid x_n = x\right\} + E\left\{E\left\{E\left\{\alpha \sum_{k=0}^{\infty} \alpha^k \, r_{n+k+1} \mid x_{n+1} = x'\right\} \mid u_n = u, x_n = x\right\} \mid x_n = x\right\}$$

$$= E\left\{r_n \mid x_n = x\right\} + E\left\{E\left\{\alpha V^{\pi}(x') \mid u_n = u, x_n = x\right\} \mid x_n = x\right\}$$

A D R L

Derivation of Bellman Equation for V (3)

$$= E\{r_n \mid x_n = x\} + E\left\{E\left\{\alpha V^\pi(x') \mid u_n = u, x_n = x\right\} \mid x_n = x\right\}$$

rolling back the expectations

$$V^\pi(x) = E\{r_n \mid x_n = x\} + E\left\{\alpha V^\pi(x') \mid x_n = x\right\}$$

Bellman equation

$$V^\pi(x) = E\left\{r_n + \alpha V^\pi(x') \mid x_n = x\right\}$$

expectation in respect to rewards
AND state transition

cf. Bellman Equation of optimal ctrl problem (prev. Lectures)	$V^*(x) = L(x,u) + \alpha E[V^*(x')]$

Bellman Eq. and probabilities

$$V^{\pi}(x) = E\left\{r_n + \alpha V^{\pi}(x') \mid x_n = x\right\}$$

$$V^{\pi}(x) = E\left\{E\left\{r_n + \alpha V^{\pi}(x') \mid u_n = u, x_n = x\right\} \mid x_n = x\right\}$$

probability of control

expected value of reward

$$V^{\pi}(x) = \sum_u \pi(x,u) \sum_{x'} \mathcal{P}^u_{xx'} \left[\mathcal{R}^u_{xx'} + \alpha V^{\pi}(x')\right]$$

state transition probability

$$\mathcal{R}^u_x = \sum_{x'} \mathcal{P}^u_{xx'} \mathcal{R}^u_{xx'}$$

Bellman Eq. as linear problem

$$V^\pi(x) = \sum_u \pi(x,u) \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \alpha V^\pi(x')]$$

Can be written as linear problem (linear in unknown V)

$$\mathbf{V} = \mathbf{A}\mathbf{V} + \mathbf{B}$$

$$[\mathbf{A}_{i,j}] = \alpha \sum_u \pi(x_i,u) \mathcal{P}^u_{x_i x_j}$$

$$[\mathbf{B}_i] = \sum_u \pi(x_i,u) \sum_{x'} \mathcal{P}^u_{x_i x'} \mathcal{R}^u_{x_i x'}$$

$\mathbf{V}$: vector with all value functions, for all states

Action value function

Idea: Value for a given first command (not necessarily from π) and subsequently following policy π

$$Q^\pi(x,u)$$

$$Q^\pi(x,u) = E_\pi\{R_n \mid x_n = x, u_n = u\} = E\left\{\sum_{k=0}^{\infty}\alpha^k r_{n+k} \mid x_n = x, u_n = u\right\}$$

compare w. definition of V

$$V^\pi(x) = E\{R_n \mid x_n = x\} = E\left\{\sum_{k=0}^{\infty}\alpha^k r_{n+k} \mid x_n = x\right\}$$

$$V^\pi(x) = \sum_u \pi(x,u) Q^\pi(x,u)$$

Action value function

Bellman Equation Derivation

$$Q^\pi(x,u) = E_\pi\{R_n \mid x_n = x, u_n = u\} = E\left\{\sum_{k=0}^{\infty} \alpha^k r_{n+k} \mid x_n = x, u_n = u\right\}$$

$$\begin{aligned}
Q^\pi(x,u) &= E\{R_n \mid x_n = x, u_n = u\} \\
&= E\left\{r_n + \alpha \sum_{k=0}^{\infty} \alpha^k r_{n+k+1} \mid x_n = x, u_n = u\right\} \\
&= E\left\{r_n + \alpha E\left\{\sum_{k=0}^{\infty} \alpha^k r_{n+k+1} \mid x_{n+1} = x'\right\} \mid x_n = x, u_n = u\right\} \\
&= E\left\{r_n + \alpha V^\pi(x') \mid x_n = x, u_n = u\right\}
\end{aligned}$$

$$= E\left\{r_n + \alpha V^\pi(x') \mid x_n = x, u_n = u\right\}$$

using $V^\pi(x) = \sum_u \pi(x,u) Q^\pi(x,u)$

$$Q^\pi(x,u) = E\left\{r_n + \alpha \sum_{u'} \pi(x',u') Q^\pi(x',u') \mid x_n = x, u_n = u\right\}$$

$$Q^\pi(x,u) = \sum_{x'} \mathcal{P}^u_{xx'} \left[\mathcal{R}^u_{xx'} + \alpha \sum_{u'} \pi(x',u') Q^\pi(x',u')\right]$$

Bellman Equation

(note linear in Q, as previously in V)

Optimal policy

for all $x \in \mathbf{X}$

for all possible policies π

$$V^*(x) \geq V^\pi(x)$$

$$V^*(x) = \max_\pi V^\pi(x)$$

V* vs Q*

$$Q^*(x,u) = \max_{\pi} Q^{\pi}(x,u)$$

$$V^*(x) = \sum_a \pi^*(x,u) Q^*(x,u)$$

V* vs Q*

$$V^*(x) = \sum_a \pi^*(x,u) Q^*(x,u)$$

$\pi^*(x,u)$ is always between 0 an 1

$$V^*(x) = \sum_u \pi^*(x,u) Q^*(x,u) \leq \max_u Q^*(x,u)$$

$$V^*(x) = \max_u Q^*(x,u)$$

A D R L

ETH Zürich

Opt. Bellman equation

Derivation

$$V^*(x) = \max_u Q^*(x,u)$$

apply Bellman equation

$$Q^*(x,u) = E\left\{r_n + \alpha \sum_{u'} \pi^*(x',u')Q^*(x',u') \mid x_n = x, u_n = u\right\}$$

using again $V^*(x) = \sum_u \pi^*(x,u)Q^*(x,u)$ and $V^*(x) = \max_u Q^*(x,u)$

$$Q^*(x,u) = E\left\{r_n + \alpha \max_{u'} Q^*(x',u') \mid x_n = x, u_n = u\right\}$$

Optimal Bellman equation for action-value function

A D R L

ETH Zürich

Optimal Bellman Eq.

$$Q^*(x,u) = E\left\{ r_n + \alpha \max_{u'} Q^*(x',u') \mid x_n = x, u_n = u \right\}$$

$$Q^*(x,u) = \sum_{x'} \mathcal{P}^u_{xx'} \left[\mathcal{R}^u_{xx'} + \gamma \max_{u'} Q^*(x',u') \right]$$

Note: Now Bellman Equation is NOT linear anymore (because of max operator)

Optimal Bellman Equation for V

use $V^\pi(x) = E\left\{r_n + \alpha V^\pi(x') \mid x_n = x\right\}$ for opt. policy π^*

$$V^*(x) = \max_u E\left\{r_n + \alpha V^*(x') \mid x_n = x\right\}$$

Optimal Bellman equation for state-value function

$$V^*(x) = \max_{u \in \mathbf{U}} \sum_{x'} \mathcal{P}^u_{xx'}\left[\mathcal{R}^u_{xx'} + \alpha V^*(x')\right]$$

A D R L

ETH Zürich

[Sutton & Barto]

"By means of V* , the optimal expected long-term return is turned into a quantity that is locally and immediately available for each state. Hence, a one-step-ahead search yields the long-term optimal actions."

'greedy policy' in respect to V* is optimal

Q* - removes even the need for the on-step-ahead search

(and transition probabilities - will be important for model free algorithms!)

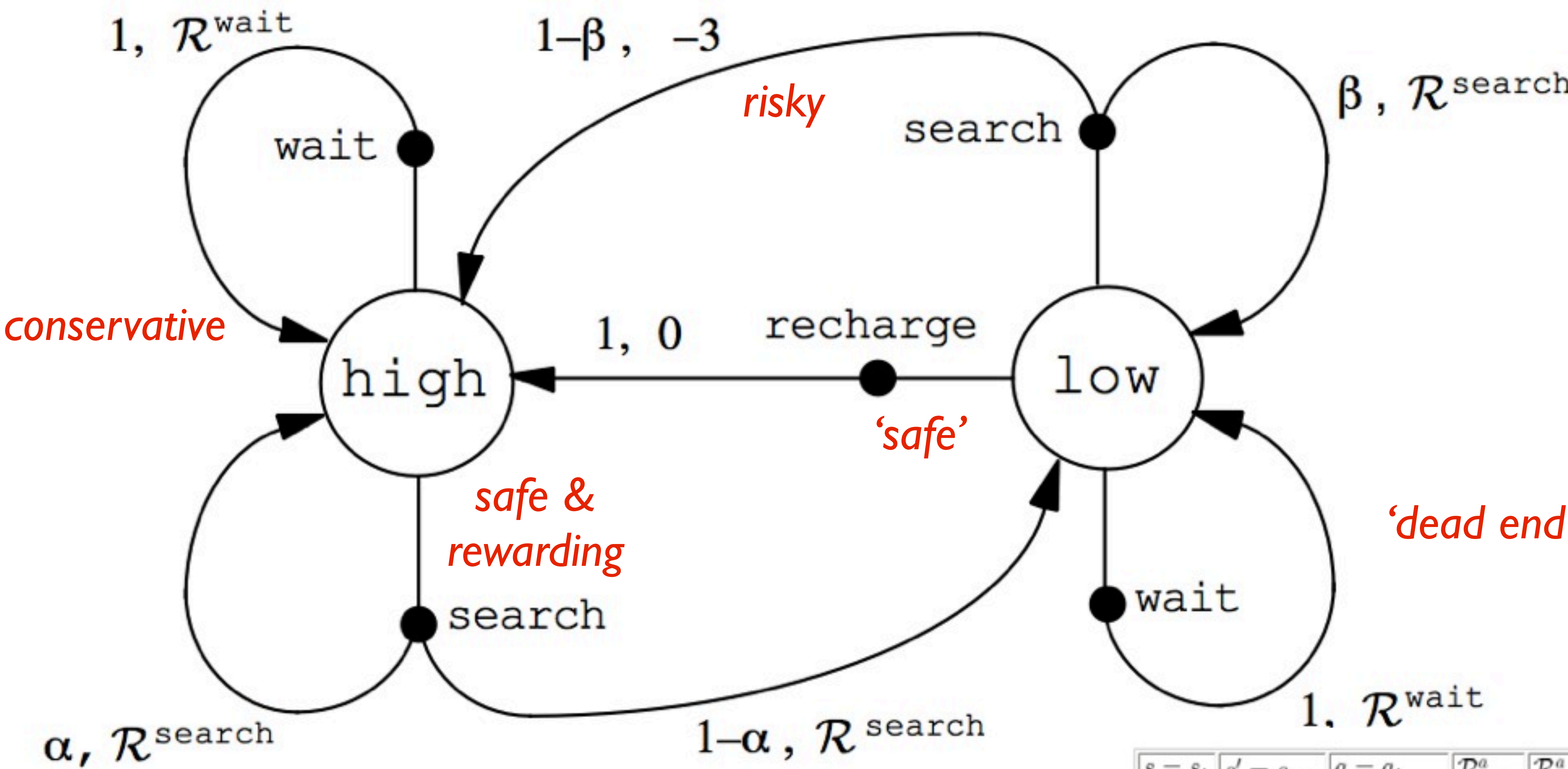
"Having makes choosing optimal actions still easier. With Q* the agent does not even have to do a one-step ahead search [...] The action-value function effectively caches the results of all one-step ahead searches"

MDP
Exact solutions

There are 3 solutions to the MDP problem:

- Value iteration
- Policy Iteration
- Generalized policy iteration

These are all model based, from GPI we will derive sample based (model-free) methods

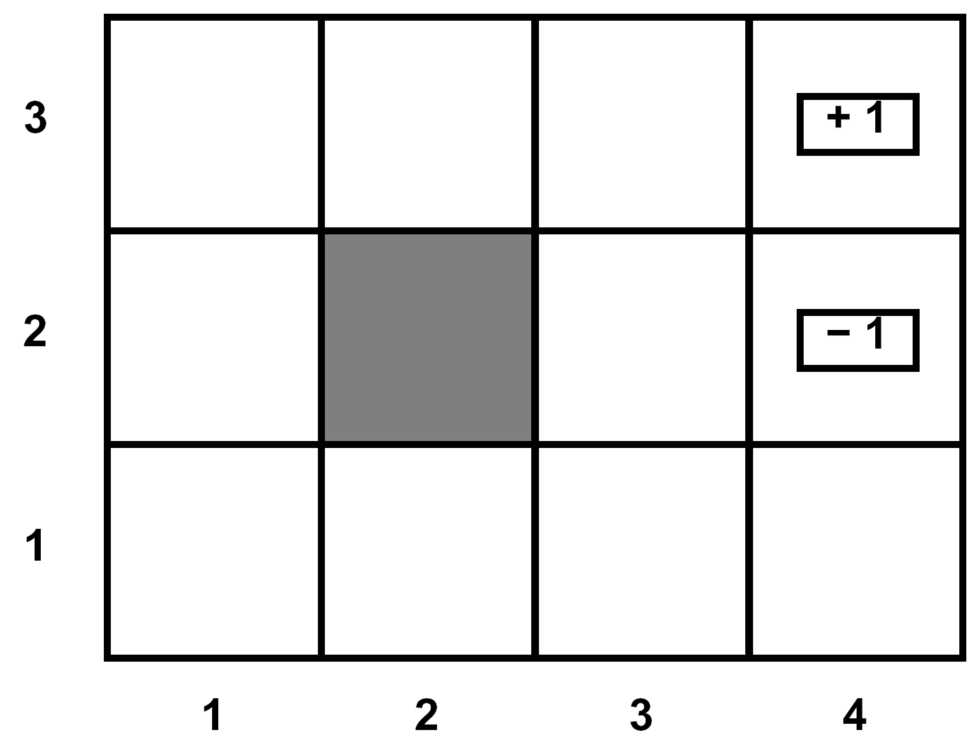


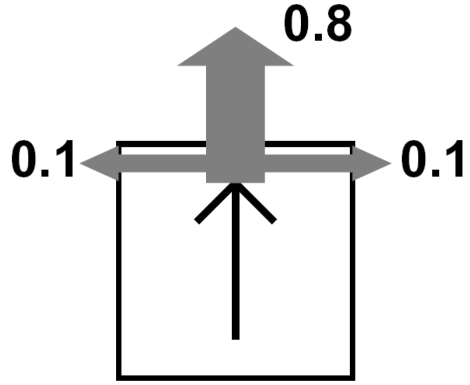
$s = s_t$	$s' = s_{t+1}$	$a = a_t$	$\mathcal{P}^a_{ss'}$	$\mathcal{R}^a_{ss'}$
high	high	search	α	$\mathcal{R}^{\text{search}}$
high	low	search	$1-\alpha$	$\mathcal{R}^{\text{search}}$
low	high	search	$1-\beta$	-3
low	low	search	β	$\mathcal{R}^{\text{search}}$
high	high	wait	1	$\mathcal{R}^{\text{wait}}$
high	low	wait	0	$\mathcal{R}^{\text{wait}}$
low	high	wait	0	$\mathcal{R}^{\text{wait}}$
low	low	wait	1	$\mathcal{R}^{\text{wait}}$
low	high	recharge	1	0
low	low	recharge	0	0.

A D R L

Canonical Example: Grid World

- The agent lives in a grid
- Walls block the agent's path
- The agent's actions do not always go as planned:
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- Big rewards come at the end

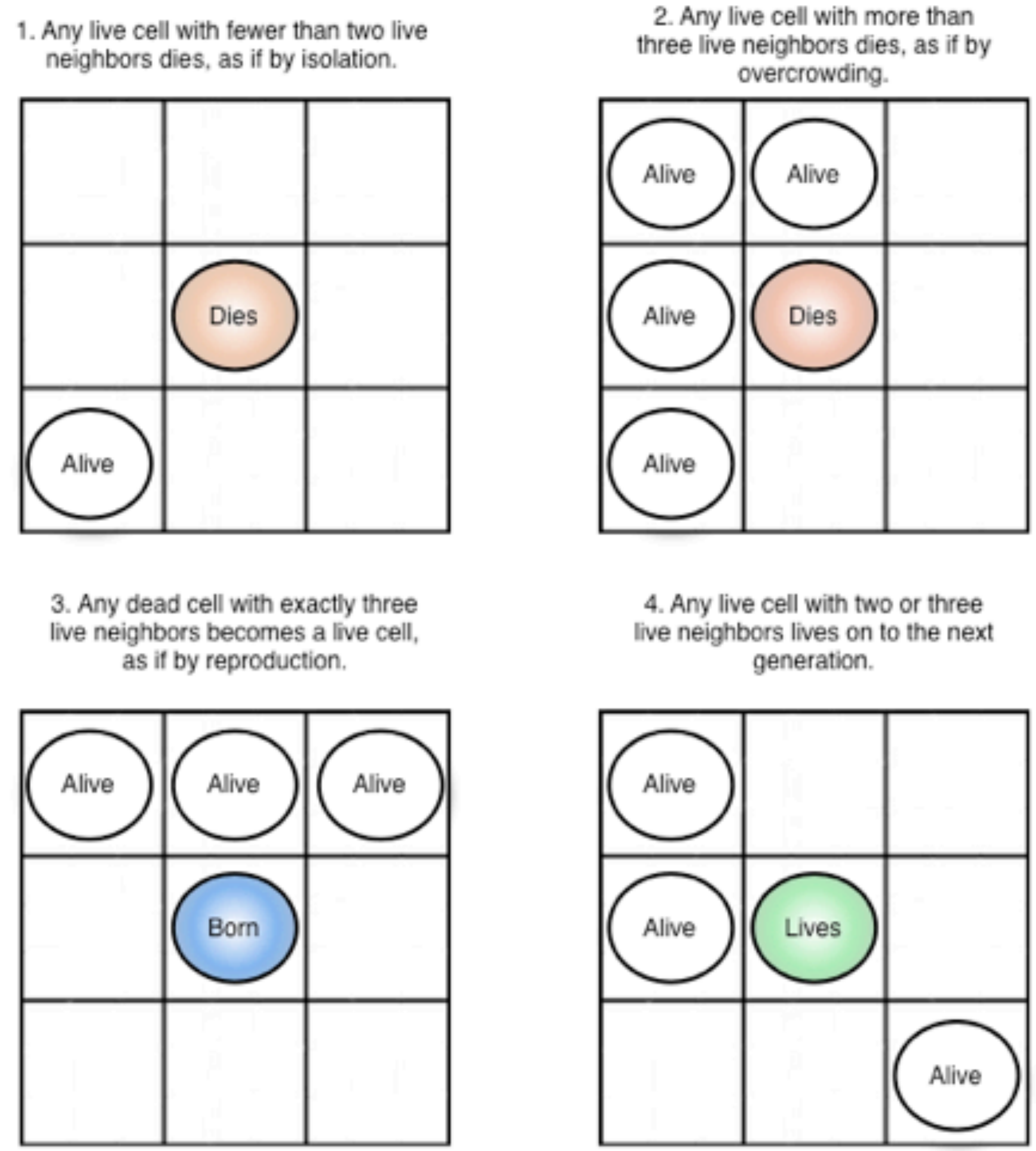




Are MDPs trivial?

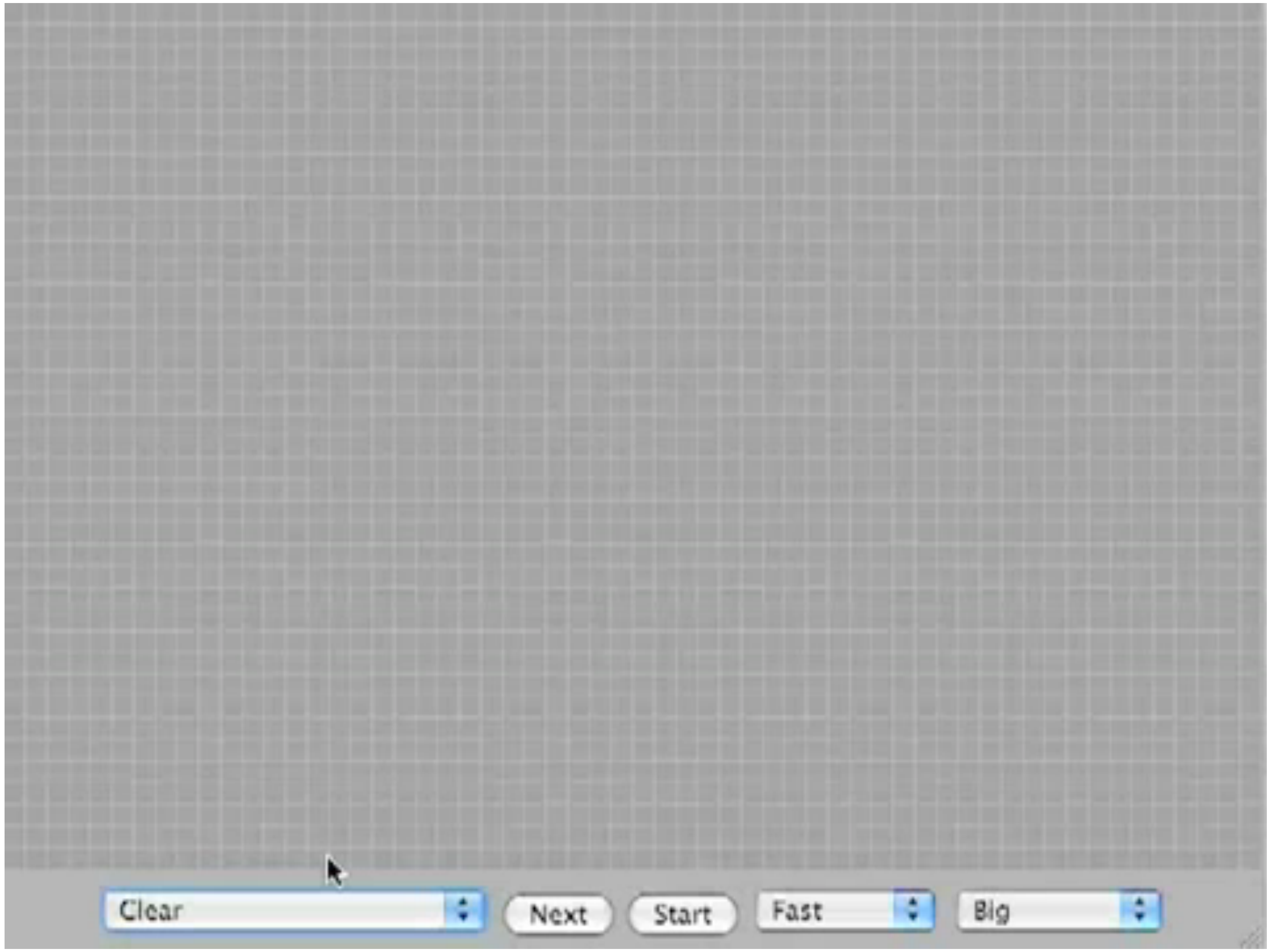
What can discrete systems express?

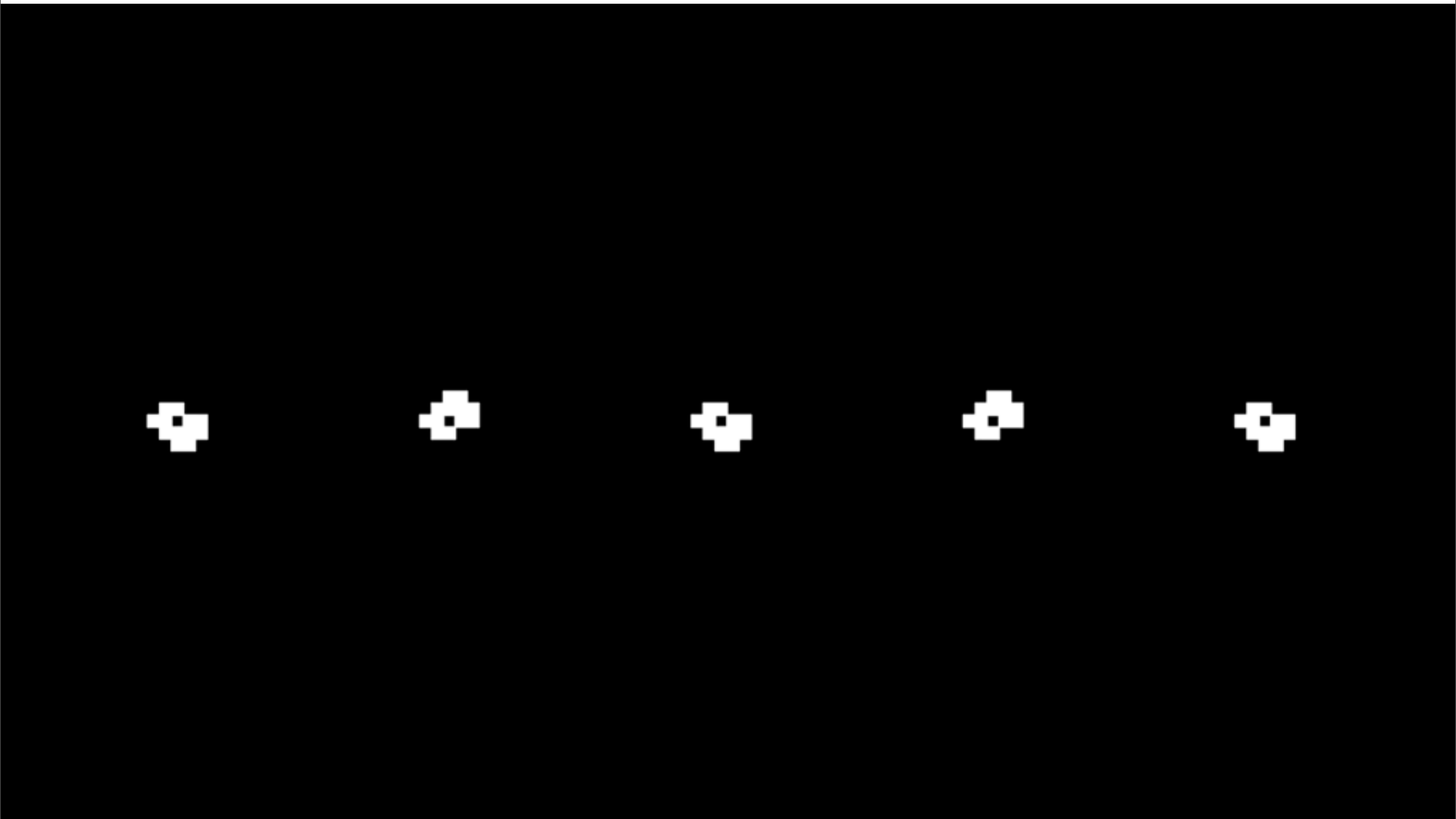
Conway's Game of Life



not an MDP, not a Markov Chain, even simpler: deterministic finite state automaton!

4 simple rules!

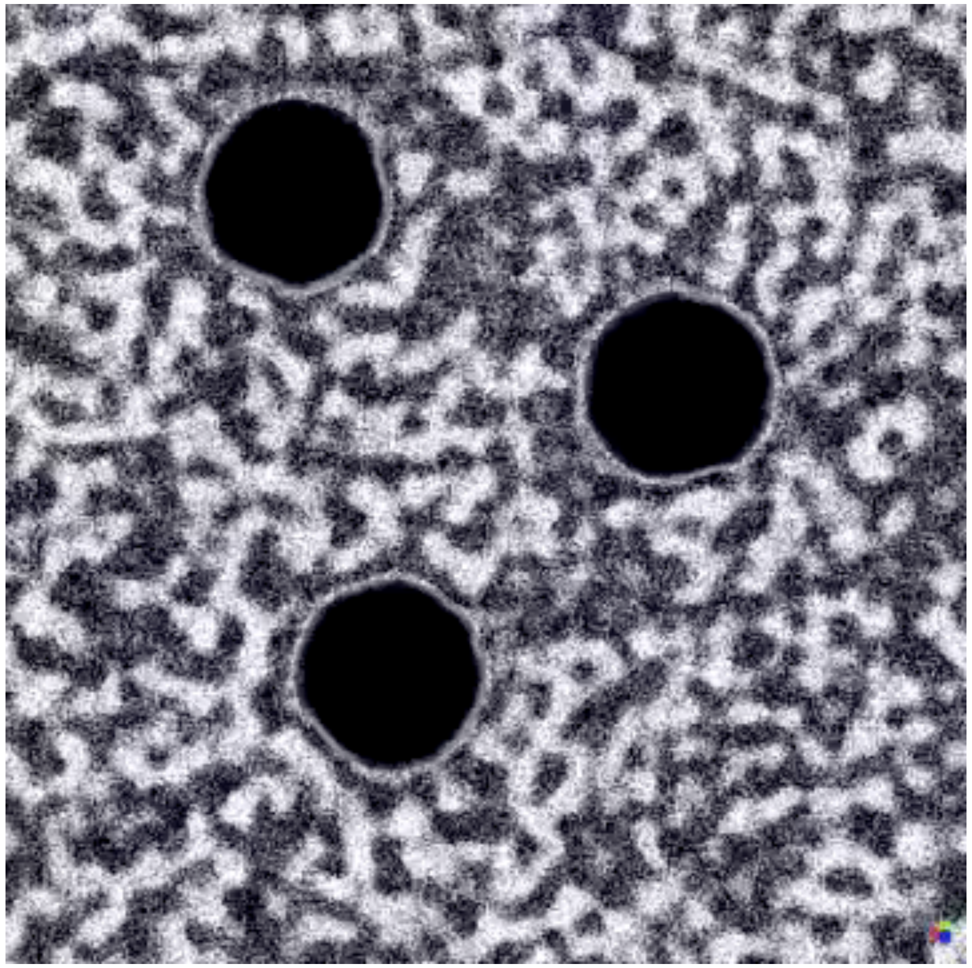

Clear
Next
Start
Fast
Big



Turing machine

… still discrete, deterministic

continuous, deterministic



... but implementation is discrete!

The basic question...

How do we find V / optimal policy???

Dynamic Programming (again...)

Goal: Find V^{π} V^{*} Q^{π} Q^{*}

$V(s)$ $Q(s,a)$

Policy evaluation

in principle: n linear equations with n unknowns

Idea: start with (arbitrary) initial V, iteratively find true Value function

$$V_{k+1}(x) = \sum_u \pi(x,u) \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \alpha V_k(x')]$$

$V_k = V^\pi$ is fixed point

Algorithm 1 Iterative Policy Evaluation Algorithm

Input: π, the policy to be evaluated

Initialize $V(x) = 0$, for all $x \in \mathcal{X}^+$

repeat

$\Delta \leftarrow 0$

for each $x \in \mathcal{X}$

$v \leftarrow V(x)$

$V(x) \leftarrow \sum_u \pi(x,u) \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \gamma V(x')]$

$\Delta \leftarrow \max(\Delta, |v - V(x)|)$

until $\Delta < \theta$ (a small positive number)

Return: $V \approx V^\pi$

two nested loops

visits all states

stop if increment is 'small enough'

Theoretically: loop infinitely

A D R L

ETH Zürich

Policy evaluation f. Q

- Policy evaluation also works for state-action value function
- use the corresponding Bellman equation as update rule

$$Q_{k+1}(x,u) = \sum_{x'} \mathcal{P}^u_{xx'} \left[\mathcal{R}^u_{xx'} + \alpha \sum_{u'} \pi \ (x',u') Q_k(x',u') \right]$$

A D R L

ETH Zürich

EOF L7

Policy improvement

A policy π' is better than policy π if

$$\forall\ x \in \mathbf{X}: \qquad V^{\pi'}(x) \geq V^{\pi}(x)$$

$$\exists\ x \in \mathbf{X}: \qquad V^{\pi'}(x) > V^{\pi}(x)$$

A D R L

ETH Zürich

Policy improvement theorem

Assume two policies π π'

Identical for all states, but x $\pi'(x) \neq \pi(x)$

If in 1(!) chosen state x $Q^{\pi}(x, \pi'(x)) \geq V^{\pi}(x)$

then π' is a better policy than π

Policy improvement theorem:

what if improvement in all states $\overset{?}{\Rightarrow}$ $V^{\pi'} \geq V^{\pi}$

A D R L

ETH Zürich

Policy improvement theorem - proof

Preliminaries:

consider deterministic policy

$$\mu(x) = a_s, \qquad \text{where: } \pi(a_s \mid x) = 1.$$

To show:

$$Q^\pi(x, \mu'(x)) \geq V^\pi(x) \quad \Rightarrow \quad V^{\pi'} \geq V^\pi$$

Policy improvement theorem - proof

$$V^\pi(x) \leq Q^\pi(x, \mu'(x)) = E_\pi \left\{ r_n + \alpha V^\pi(x_{n+1}) | u_n = \mu'(x), x_n = x \right\}$$

$$V^\pi(x) \leq E_\pi \left\{ r_n + \alpha\, Q^\pi(x_{n+1}, \mu'(x_{n+1})) \mid u_n = \mu'(x), x_n = x \right\}$$

$$V^\pi(x) \leq E_\pi \left\{ r_n + \alpha\, E_\pi \left\{ r_{n+1} + \alpha V^\pi(x_{n+2}) \mid u_{n+1} = \mu'(x'), x_{n+1} = x' \right\} | u_n = \mu'(x), x_n = x \right\}$$

$$V^\pi(x) \leq E_\pi \left\{ r_n + \alpha r_{n+1} + \alpha^2 V^\pi(x_{n+2}) | u_{n+1} = \mu'(x'), x_{n+1} = x', u_n = \mu'(x), x_n = x \right\}$$

simplify notation

$$V^\pi(x) \leq E_{\substack{u_{[n,n+1]} \sim \pi' \\ u_{[n+2,\dots]} \sim \pi}} \left\{ r_n + \alpha r_{n+1} + \alpha^2 V^\pi(x_{n+2}) | x_n = x \right\}$$

repeat argument to infinity

$$V^\pi(x) \leq E_\pi \left\{ r_n + \alpha r_{n+1} + \alpha^2 r_{n+2} + \alpha^3 r_{n+3} + \dots \mid u_{[n,n+1,n+2,\dots]} \sim \pi', x_n = x \right\}$$

Policy improvement theorem - proof - conclusion

Policy π' is followed for all time steps

thus omit condition on $u_{[n,n+1,n+2,\ldots]} \sim \pi'$ yields $V^{\pi'}$

$$V^{\pi}(x) \leq E_{\pi'}\left\{r_n + \alpha r_{n+1} + \alpha^2 r_{n+2} + \alpha^3 r_{n+3} + \ldots \mid x_n = x\right\} = V^{\pi'}(x)$$

$$V^{\pi}(x) \leq V^{\pi'}(x)$$

QED

Greedy policy update

Consider the following greedy policy update rule

$$\pi'(x) = \underset{u}{\operatorname{argmax}}\, Q^{\pi}(x,u)$$
$$= \underset{u}{\operatorname{argmax}}\, E\left\{r_n + \alpha V^{\pi}(x_{n+1}) | x_n = x, u_n = u\right\}$$

PIT: greedy policy update will always yield better policy

Convergence of update

What if new policy is strictly equal to the old one:

$$V^{\pi'} = V^{\pi}$$

using policy update rule

$$V^{\pi'}(x) = \max_u E\left\{r_n + \alpha V^{\pi'}(x_{n+1}) | x_n = x, u_n = u\right\}$$
$$= \max_u \sum_{x'} \mathcal{P}^u_{xx'}[\mathcal{R}^u_{xx'} + \alpha V^{\pi'}(x')]$$

→ yields optimal Bellman Equation

⇒ PI will give better policy, unless Policy is already optimal

Policy iteration

Using Policy Evaluation and Policy improvement, find optimal policy iteratively

Policy iteration

1) Policy evaluation (complete!)
2) Policy improvement
3) repeat

need to sweep ('visit all') x in an iteration ('full backup')

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

discount factor < 1 or
termination state under policy
(convergence of cost-to-go!)

[SB] Ch 4.1

Algorithm 2 Policy Iteration

1. Initialization

select $V(x) \in \Re$ and $\pi(x) \in \mathbf{U}$ arbitrarily for all $x \in \mathbf{X}$

2. Policy evaluation

repeat

$\Delta \leftarrow 0$

for each: $x \in \mathcal{X}$

$v \leftarrow V(x)$

$V(x) \leftarrow \sum_u \pi(x,u) \sum_{x'} \mathcal{P}_{xx'}^{\pi(x)} [\mathcal{R}_{xx'}^u + \alpha V(x')]$

$\Delta \leftarrow \max(\Delta, |v - V(x)|)$

until $\Delta < \theta$ (a small positive number)

3. Policy Improvement

$policyIsStable \leftarrow true$

for $x \in \mathcal{X}$ **do**

$b \leftarrow \pi(x)$

$\pi(x) \leftarrow \text{argmax}_u \sum_{x'} \mathcal{P}_{xx'}^u [\mathcal{R}_{xx'}^u + \alpha V(x')]$

if $b \neq \pi(x)$ **then**

$policyIsStable \leftarrow false$

end if

end for

if $policyIsStable$ **then**

stop

else

go to 2

end if

Return: a policy, π, such that: $\pi(x) = \arg\max_u \sum_{x'} \mathcal{P}_{xx'}^u [\mathcal{R}_{xx'}^u + \alpha V(x')]$

Visits all states! repeatedly

Policy evaluation

Policy improvement

Visits all states!

Value iteration

Can we get away without several ('infinite') # of sweeps in the Policy Evaluation step?

In reality: truncate PE after a finite number of steps

Idea: truncate after ONE iteration

→ PE and PI can be merged into a single update rule:

$$V_{k+1}(x) = \max_u \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \alpha V_k(x')]$$

Value iteration

Algorithm 3 Value Iteration

Initialization: $V(x) \in \Re$ and $\pi(x) \in \mathbf{U}$ arbitrarily for all $x \in \mathbf{X}$

repeat

$\Delta \leftarrow 0$

for $x \in \mathbf{X}$ **do**

$v \leftarrow V(x)$

$V(x) \leftarrow \max_u \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \alpha V(x')]$

$\Delta \leftarrow \max(\Delta, |v - V(x)|)$

end for

until $\Delta < \theta$ (a small positive number)

Return: a policy, π, such that: $\pi(x) = \arg\max_u \sum_{x'} \mathcal{P}^u_{xx'} [\mathcal{R}^u_{xx'} + \alpha V(x')]$

Value iteration vs. Policy evaluation

VI

$$V_{k+1}(s) = \max_a E\{r_{t+1} + \gamma V_k(s_{t+1}) | s_t = s, a_t = a\}$$
$$= \max_a \sum_{s'} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma \ V_k(s')]$$

evaluate most rewarding successor state

PE

$$V_{k+1}(s) = E_\pi\{r_{t+1} + \gamma V_k(s_{t+1}) | s_t = s\}$$
$$= \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma V_k(s')]$$

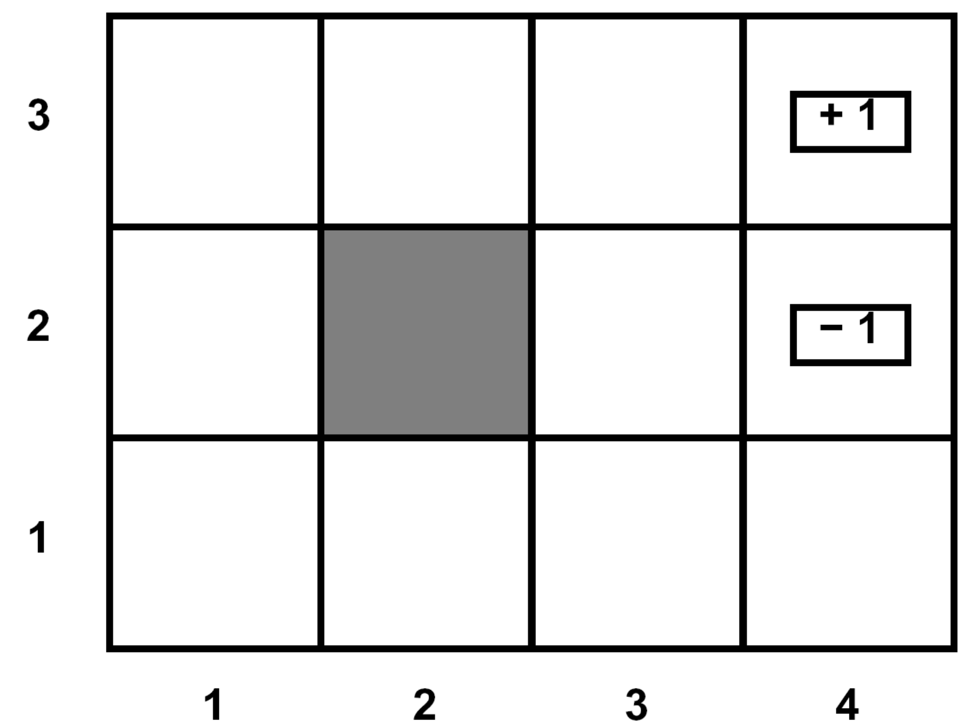
evaluate all successor states

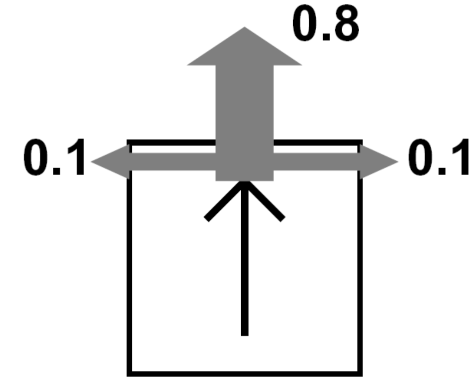
ADRL

ETH Zürich

Canonical Example: Grid World

- The agent lives in a grid
- Walls block the agent's path
- The agent's actions do not always go as planned:
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- Big rewards come at the end





Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1

0.8*0.9*1
+0.1*0.9*0.0
+0.1*0.9*0

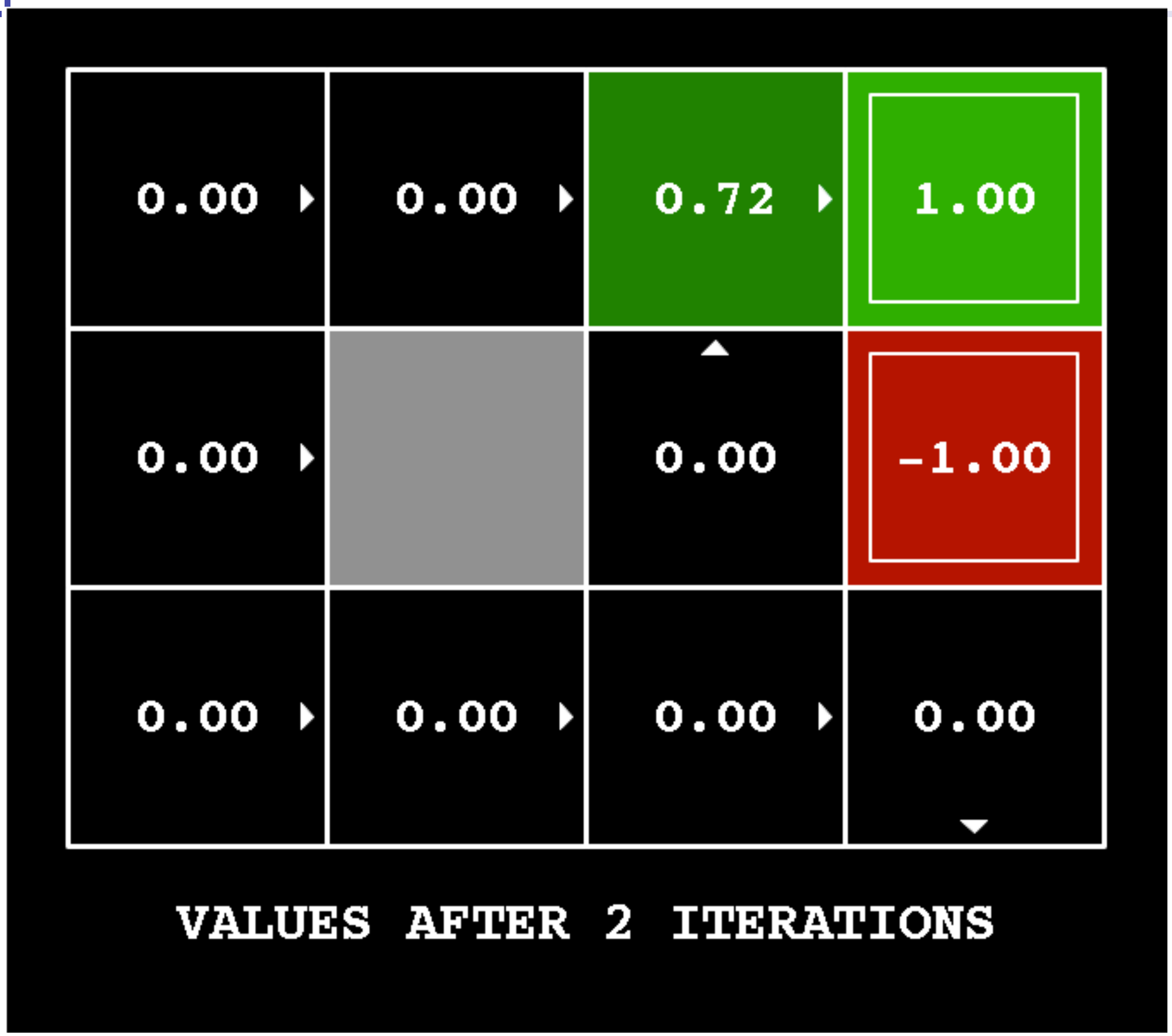
= 0.72

$$V_{k+1}(s) = \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \; V_k(s')\right]$$

A D R L

Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1



0.8*0.9*1
+0.1*0.9*0.72
+0.1*0.9*0

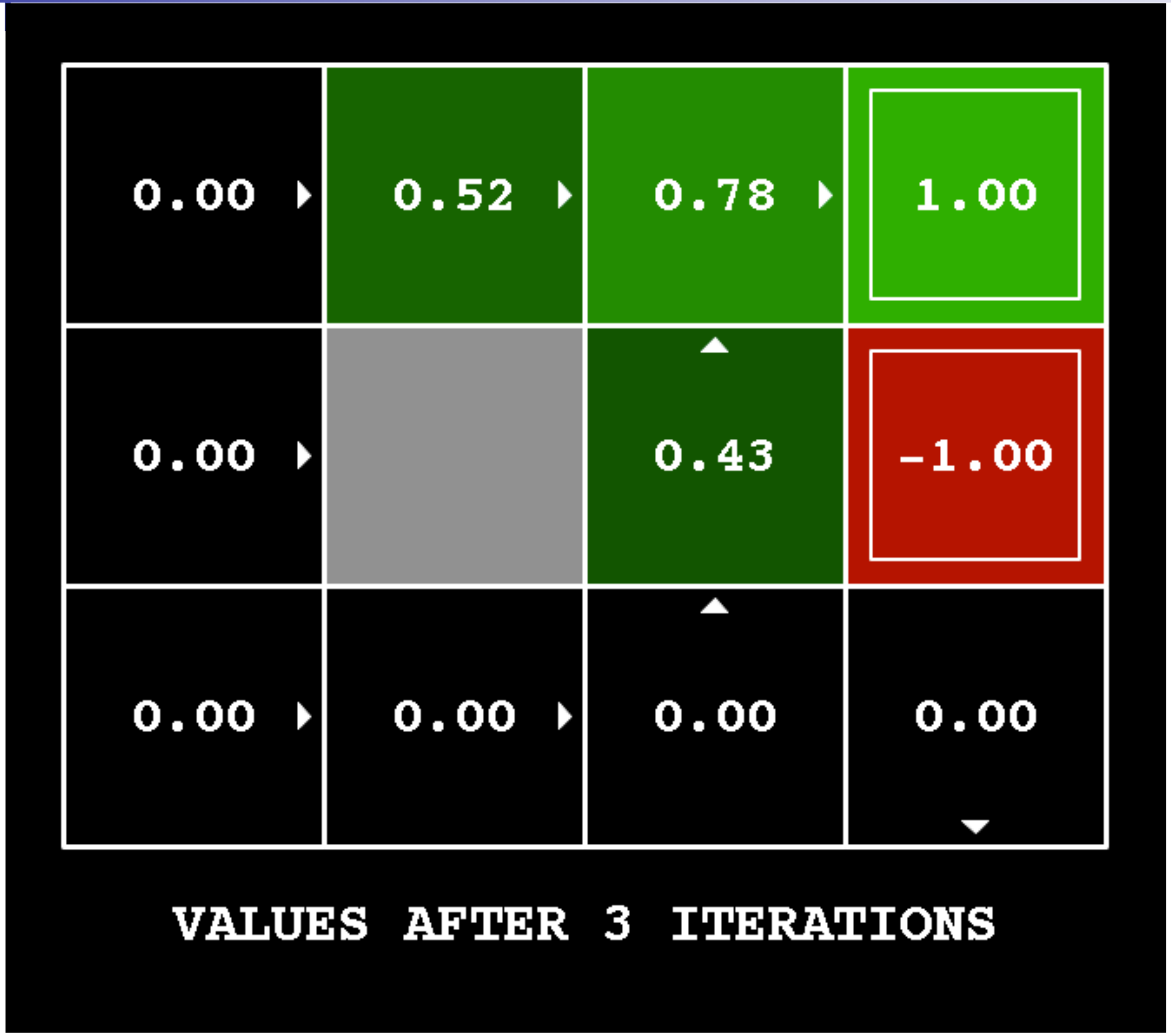
= 0.78

A D R L

$$V_{k+1}(s) = \max_a \sum_{s'} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma \; V_k(s')]$$

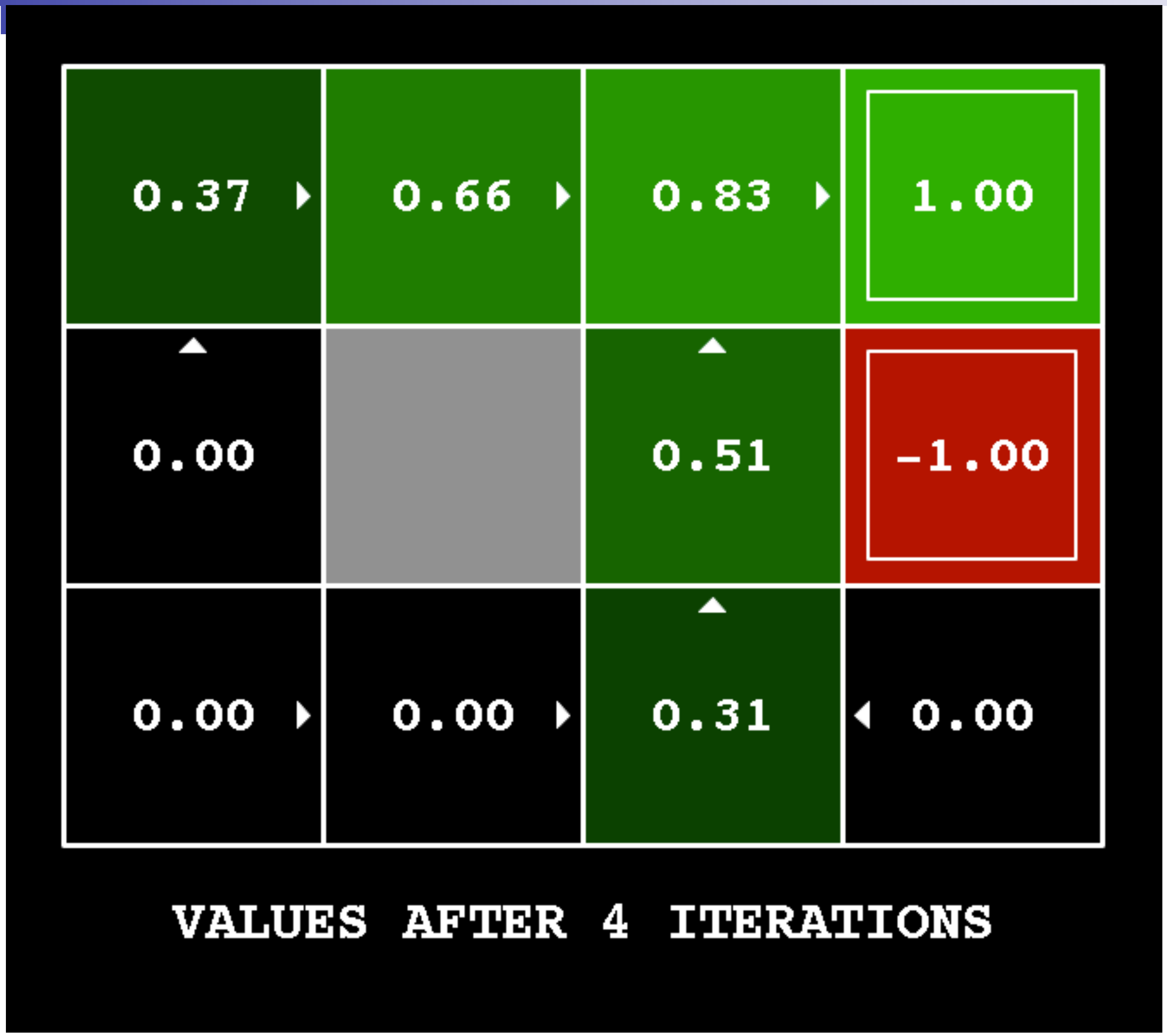
Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1



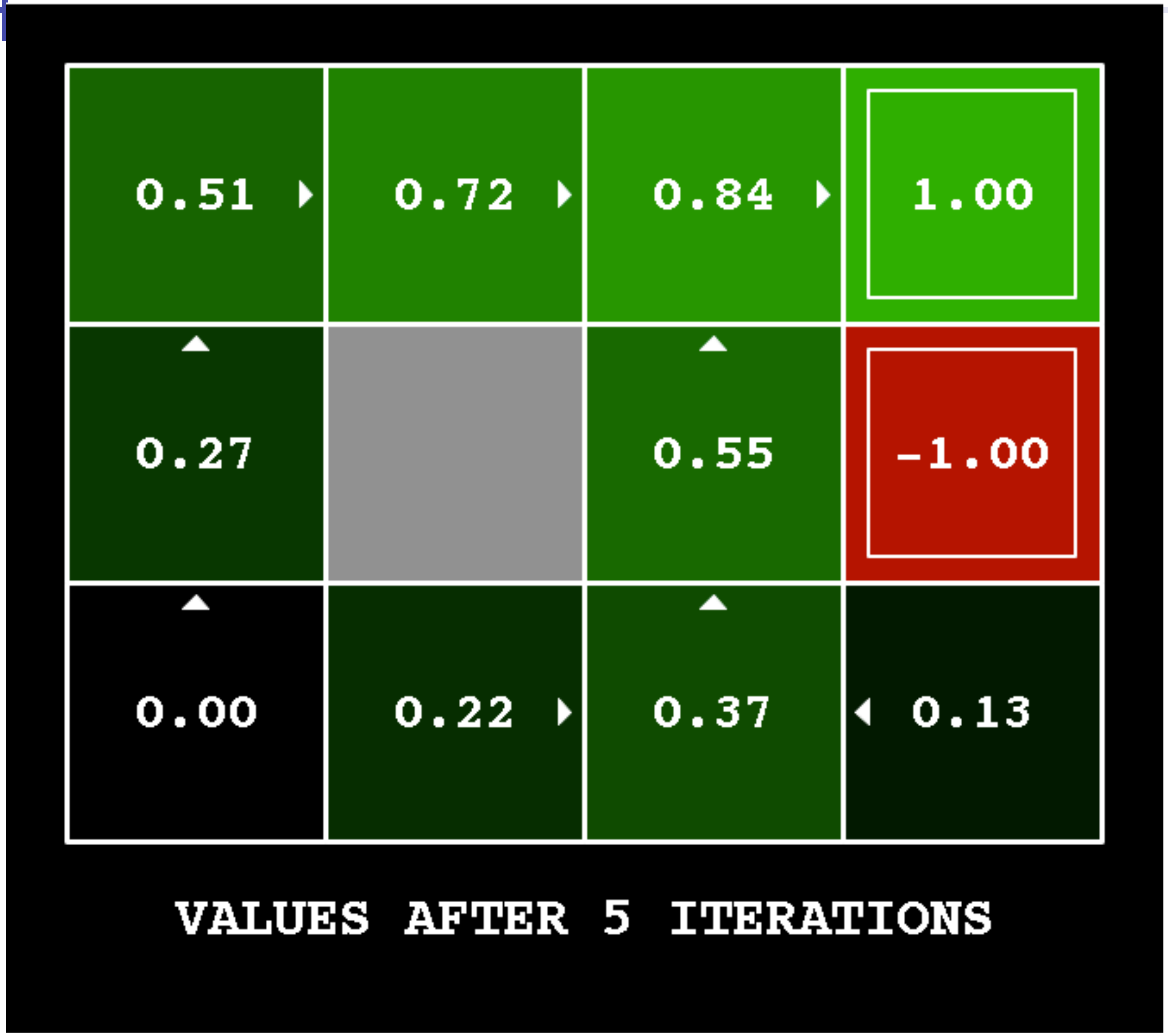
Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1



Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1

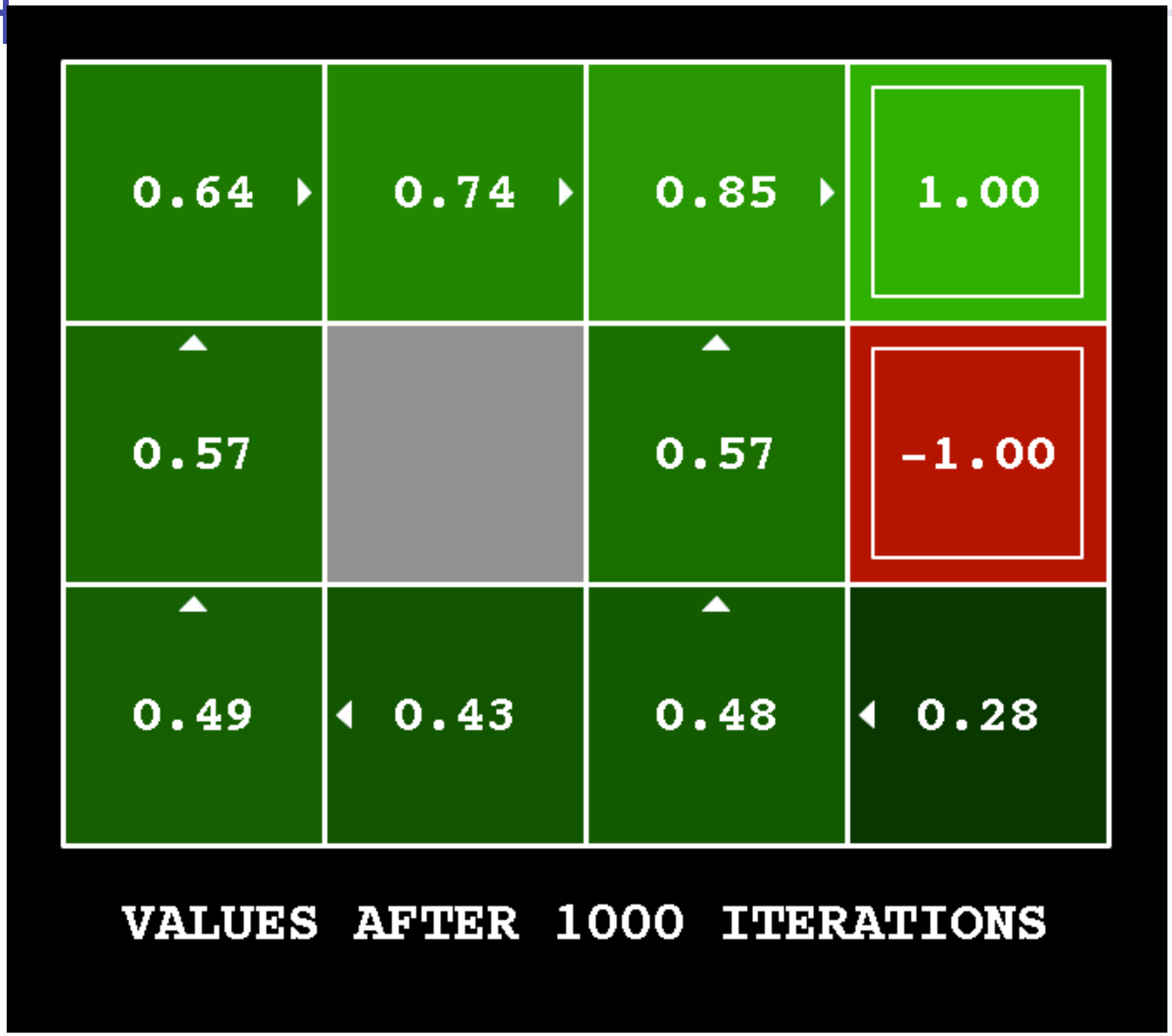


Value Iteration in Gridworld

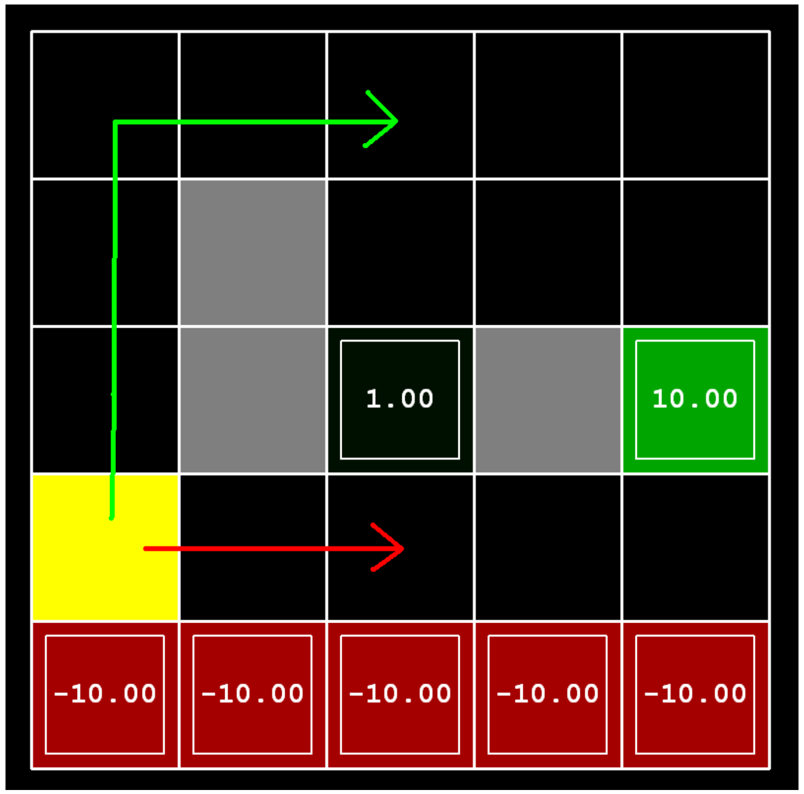
noise = 0.2, γ =0.9, two terminal states with R = +1 and -1

Value Iteration in Gridworld

noise = 0.2, γ =0.9, two terminal states with R = +1 and -1



Exercise 1: Effect of discount, noise



	1	2	3	4
a				
b				
c				
d				

(a) Prefer the close exit (+1), risking the cliff (-10)

(b) Prefer the close exit (+1), but avoiding the cliff (-10)

(c) Prefer the distant exit (+10), risking the cliff (-10)

(d) Prefer the distant exit (+10), avoiding the cliff (-10)

(1) γ = 0.1, noise = 0.5

(2) γ = 0.99, noise = 0

(3) γ = 0.99, noise = 0.5

(4) γ = 0.1, noise = 0

Policy iteration

1) Policy evaluation (complete!)
2) Policy improvement
3) repeat

need to visit all s in an iteration ('full backup')

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

discount factor < 1 or
termination under policy
(convergence of cost-to-go!)

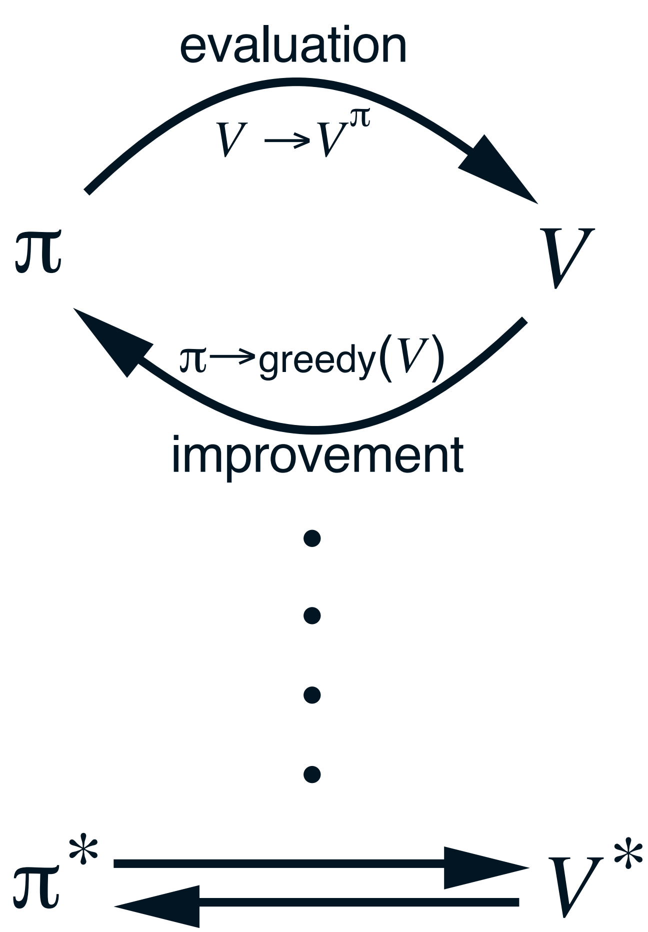
[SB] Ch 4.1

A D R L

ETH Zürich

Generalized Policy Iteration

policy evaluation - improvement - iteration



value iteration

many variants to the previously seen 'basic' algorithms

Full backup???

Both VI and PI have to visit all states!

```
traceroute to duerer.usc.edu (128.125.125.41), 64 hops max, 52 byte packets
 1  192.168.1.1 (192.168.1.1)  1.014 ms  0.992 ms  0.780 ms
 2  * * *
 3  217-168-57-101.static.cablecom.ch (217.168.57.101)  14.084 ms  18.090 ms  7.835 ms
 4  84.116.211.25 (84.116.211.25)  189.901 ms  190.472 ms  190.575 ms
 5  84.116.202.241 (84.116.202.241)  183.342 ms  188.042 ms  200.363 ms
 6  84.116.210.217 (84.116.210.217)  183.742 ms  182.100 ms  183.804 ms
 7  fr-par02a-rd1-gi-15-0-0.aorta.net (84.116.130.213)  177.586 ms
    at-vie15a-rd1-xe-4-1-0.aorta.net (84.116.130.193)  184.864 ms  185.053 ms
 8
 9
10  xe-0.equinix.snjsca04.us.bb.gin.ntt.net (206.223.116.12)  183.620 ms  186.715 ms  191.808 ms
11  ae-7.r20.snjsca04.us.bb.gin.ntt.net (129.250.5.52)  186.407 ms  186.333 ms  204.364 ms
12  ae-4.r21.lsanca03.us.bb.gin.ntt.net (129.250.6.10)  198.648 ms  402.438 ms  198.831 ms
13  ae-2.r05.lsanca03.us.bb.gin.ntt.net (129.250.5.86)  192.220 ms  200.324 ms  392.242 ms
14  165.254.21.242 (165.254.21.242)  208.798 ms  193.002 ms  194.576 ms
15  130.152.181.131 (130.152.181.131)  182.530 ms  188.588 ms  181.767 ms
16  rtr30-v255.usc.edu (128.125.251.148)  183.561 ms  189.412 ms  181.406 ms
17  duerer.usc.edu (128.125.125.41)  182.365 ms  183.795 ms  186.961 ms
```

Shortest(?) path: ZH-LA 17 hops

Credits

some material from:

Pieter Abbeel's Fall 2012: CS 287 Advanced Robotics @ UC Berkeley

Sutton & Barto's book: http://webdocs.cs.ualberta.ca/~sutton/book/the-book.html