Machine Intelligence 1

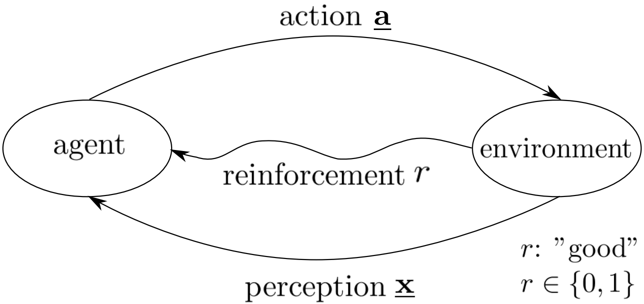
4.1 Reinforcement Learning – Evaluation

Prof. Dr. Klaus Obermayer

Fachgebiet Neuronale Informationsverarbeitung (NI)

WS 2016/2017

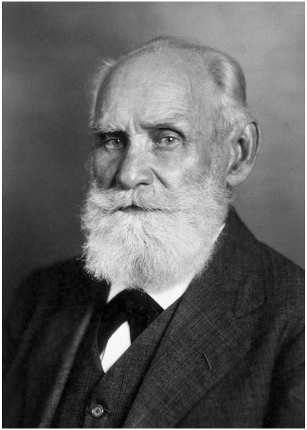
Reinforcement learning



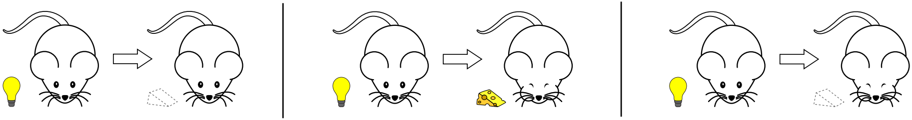
r: "good" or "bad"
$r \in \{0, 1\}$
$r \in [0, 1]$
$r \in \mathbb{R}$

4.1.1 Conditioning

Classical conditioning

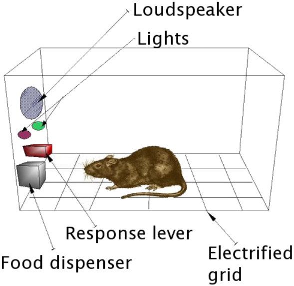


- Ivan Pavlov (1849–1936)
- : conditioned stimulus (neutral)
- : unconditioned stimulus (rewarding)
- experience reinforces *involuntary response*
- animal learns to *expect* reward



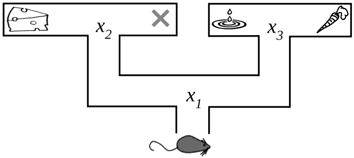
Operant conditioning

- B.F. Skinner (1904–1990)
- animal has to act voluntarily
- actions are rewarded or punished
- experience reinforces *voluntary behavior*
- animal learns how to *achieve* reward



Future rewards

- not all decisions are immediately rewarded
  - decision in **state** x_1 is crucial, but not rewarded
- some decisions require foresight
  - future reward of decision in x_1 depends on decisions in x_2 and x_3
- animal must *delay* the reinforcement of behavior



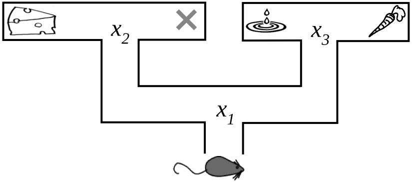
(see Dayan and Niv, 2008; Dayan, 2008)

4.1.2 Markov Decision Processes

Markov decision processes

A Markov decision process (MDP) consist of

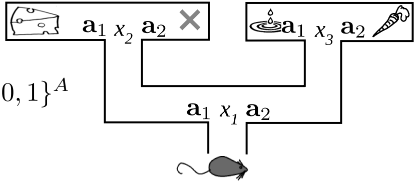
- a set of **states** $\underline{\mathbf{x}} \in \mathcal{X}$,
  - e.g. $\mathcal{X} := \{\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_S\} \subset \{0,1\}^S$ with 1-out-of-S encoding



Markov decision processes

A Markov decision process (MDP) consist of

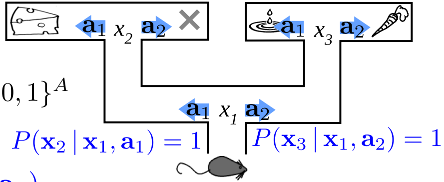
- a set of **states** $\underline{\mathbf{x}} \in \mathcal{X}$,
  - e.g. $\mathcal{X} := \{\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_S\} \subset \{0,1\}^S$ with 1-out-of-S encoding
- a set of **actions** $\underline{\mathbf{a}} \in \mathcal{A}$,
  - e.g. $\mathcal{A} := \{\underline{\mathbf{a}}_1, \ldots, \underline{\mathbf{a}}_A\} \subset \{0,1\}^A$ with 1-out-of-A encoding



Markov decision processes

A Markov decision process (MDP) consist of

- a set of **states** $\underline{\mathbf{x}} \in \mathcal{X}$,
  - e.g. $\mathcal{X} := \{\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_S\} \subset \{0,1\}^S$ with 1-out-of-S encoding
- a set of **actions** $\underline{\mathbf{a}} \in \mathcal{A}$,
  - e.g. $\mathcal{A} := \{\underline{\mathbf{a}}_1, \ldots, \underline{\mathbf{a}}_A\} \subset \{0,1\}^A$ with 1-out-of-A encoding

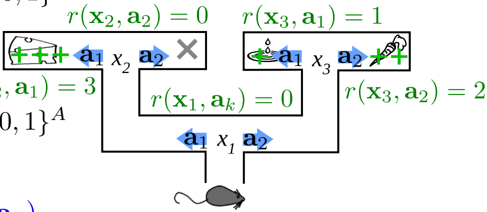


- a **transition model** $P(\underline{\mathbf{x}}_j | \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)$
  - probability to end up in $\underline{\mathbf{x}}_j$ after choosing $\underline{\mathbf{a}}_k$ in $\underline{\mathbf{x}}_i$
  - stationary distribution (Markov property)

Markov decision processes

A Markov decision process (MDP) consist of

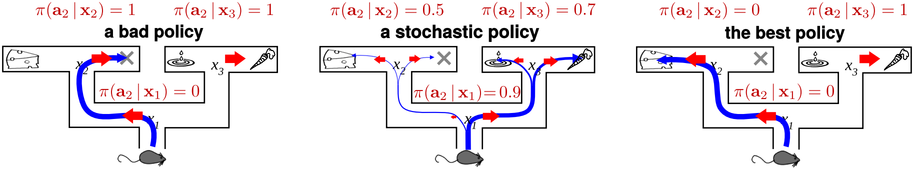
- a set of **states** $\underline{\mathbf{x}} \in \mathcal{X}$,
  - e.g. $\mathcal{X} := \{\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_S\} \subset \{0,1\}^S$ with 1-out-of-S encoding
- a set of **actions** $\underline{\mathbf{a}} \in \mathcal{A}$,
  - e.g. $\mathcal{A} := \{\underline{\mathbf{a}}_1, \ldots, \underline{\mathbf{a}}_A\} \subset \{0,1\}^A$ with 1-out-of-A encoding



- a **transition model** $P(\underline{\mathbf{x}}_j | \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)$
  - probability to end up in $\underline{\mathbf{x}}_j$ after choosing $\underline{\mathbf{a}}_k$ in $\underline{\mathbf{x}}_i$
  - stationary distribution (Markov property)
- a bounded **reward function** $r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)$
  - denotes the *average immediate reward* for choosing $\underline{\mathbf{a}}_k$ in $\underline{\mathbf{x}}_i$
  - extension with randomized rewards possible

Policy

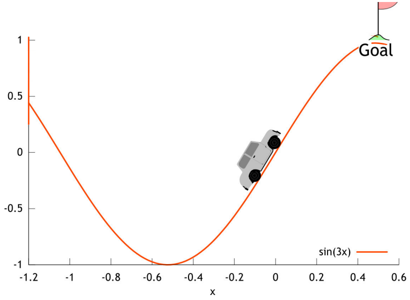
- the agent's behavior is expressed by a **policy** $\pi(\underline{\mathbf{a}}_k|\underline{\mathbf{x}}_i)$
  - the probability that the agent chooses $\underline{\mathbf{a}}_k$ in $\underline{\mathbf{x}}_i$



- the goal of RL is to find the "**optimal policy**" π^*

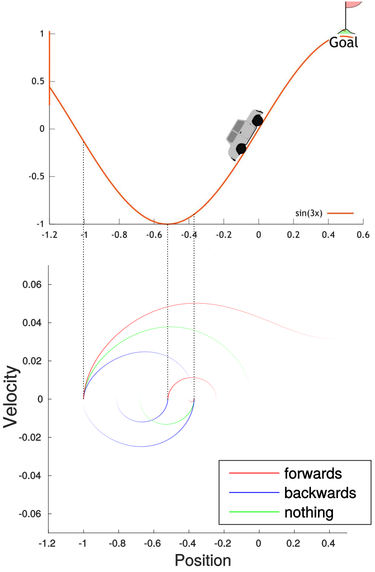
Example: mountain car

- a car in a valley between mountains
  - $\mathcal{X}$: position and velocity
- agent drives the car
  - $\mathcal{A}$: forward, backward, nothing
    (i.e., accelerate the car by $+a$, $-a$ and 0)
- dynamics are given by physics
  - transition model P simulated
  - gravitation but no friction
- goal: reach right hilltop
  - reward $r\!=\!0$, except $r\!=\!1$ at goal
- but car is underpowered
  - policy π must first pick up speed



Markov chains

- a **Markov chain** of length p
  - is a sequence of states and actions
    $$\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}\}_{t=0}^{p} \quad \subset \quad \mathcal{X} \times \mathcal{A}$$
  - actions $\underline{\mathbf{a}}^{(t)}$ are drawn from policy:
    $$\underline{\mathbf{a}}^{(t)} \quad \sim \quad \pi(\cdot \,|\, \underline{\mathbf{x}}^{(t)})$$
  - successive states $\underline{\mathbf{x}}^{(t+1)}$ are drawn from transition model:
    $$\underline{\mathbf{x}}^{(t+1)} \quad \sim \quad P(\,\cdot\,|\,\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)})$$
- given an MDP, a Markov chain depends on initial $\underline{\mathbf{x}}^{(0)}$ and policy π

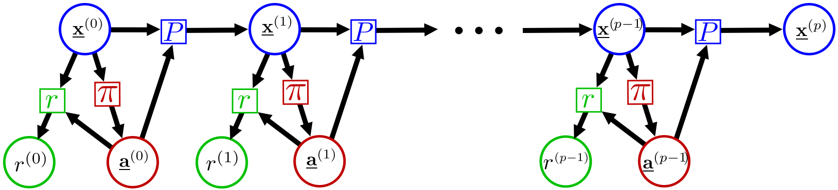


Markov chain distribution

- Markov chains are sets of random variables
  - depend on initial state $\underline{\mathbf{x}}^{(0)}$ and policy π

- joint distribution of states in a Markov chain factorizes

$$P(\underline{\mathbf{x}}^{(0)}, \ldots, \underline{\mathbf{x}}^{(p)}) \;=\; P(\underline{\mathbf{x}}^{(0)}) \prod_{t=0}^{p-1} \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}^{(t)}) \, P\big(\underline{\mathbf{x}}^{(t+1)} \big| \, \underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}_k\big)$$



(see blackboard)

4.1.3 Policy Evaluation

Value function

- a **value function** measures the quality of a policy π in state $\underline{\mathbf{x}}^{(0)}$
  - $V^\pi(\underline{\mathbf{x}}^{(0)})$ is the *expected* *reward*

$$V^\pi(\underline{\mathbf{x}}^{(0)}) \;=\; \mathbb{E}\Big[\; r(\underline{\mathbf{x}}^{(0)}, \underline{\mathbf{a}}^{(0)}) \;\Big|\; \underline{\mathbf{a}}^{(0)} \sim \pi(\cdot\,|\,\underline{\mathbf{x}}^{(0)}) \;\Big]$$

- average over selected actions

Value function

- a **value function** measures the quality of a policy π in state $\underline{\mathbf{x}}^{(0)}$
  - $V^\pi(\underline{\mathbf{x}}^{(0)})$ is the *expected* *finite sum of* *rewards*

$$V^\pi(\underline{\mathbf{x}}^{(0)}) = \mathbb{E}\left[\sum_{t=0}^{H} r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \,\middle|\, \begin{array}{c} \underline{\mathbf{a}}^{(t)} \sim \pi(\cdot\,|\,\underline{\mathbf{x}}^{(t)}) \\ \underline{\mathbf{x}}^{(t+1)} \sim P(\cdot\,|\,\underline{\mathbf{x}}^{(t)},\underline{\mathbf{a}}^{(t)}) \end{array}\right]$$

- average over Markov chains

- Markov chains start at $\underline{\mathbf{x}}^{(0)}$ and follow the transition model P and the policy π

Value function

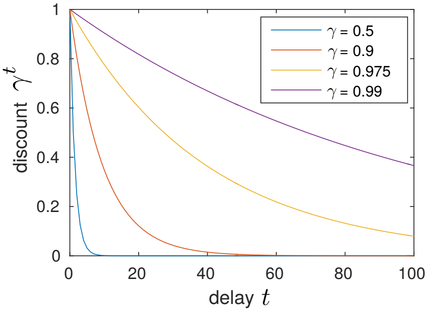
- a **value function** measures the quality of a policy π in state $\underline{\mathbf{x}}^{(0)}$
  - $V^\pi(\underline{\mathbf{x}}^{(0)})$ is the *expected infinite sum of discounted future rewards*

$$V^\pi(\underline{\mathbf{x}}^{(0)}) \;=\; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \, r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \;\middle|\; \begin{array}{c} \underline{\mathbf{a}}^{(t)} \sim \pi(\cdot \,|\, \underline{\mathbf{x}}^{(t)}) \\ \underline{\mathbf{x}}^{(t+1)} \sim P(\cdot \,|\, \underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \end{array} \right], \quad \gamma \in [0,1) \,.$$

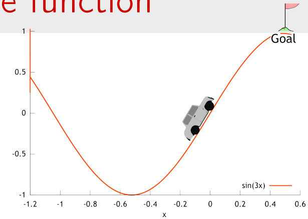
- average over Markov chains

- Markov chains start at $\underline{\mathbf{x}}^{(0)}$ and follow the transition model P and the policy π

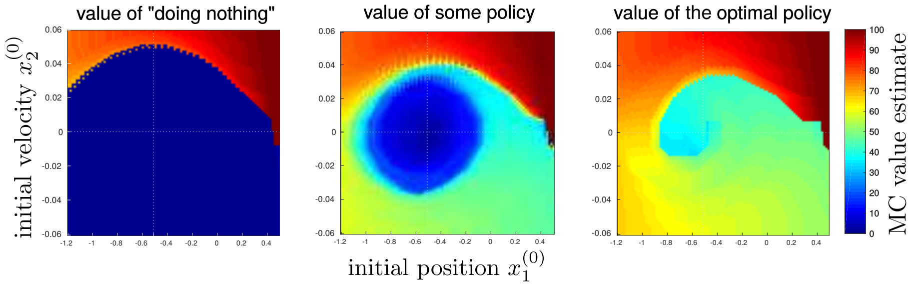
- discount factor γ: preference for short- vs. long-term goals



Monte Carlo (MC) estimation of the value function



- finite approximation of infinite Markov chains
  - rewards weighted by $\gamma^H < \epsilon$ are neglected
  - value is the discounted reward averaged over n Markov chains of length H
  - n must be sufficiently large
- requires simulator to draw n chains from the same initial state $\underline{\mathbf{x}}^{(0)}$
- every state must be evaluated often $\rightsquigarrow$ not sample efficient



The Bellman equation (1)

Richard E. Bellman
(1920–1984)

$$
\begin{aligned}
V^\pi(\underline{\mathbf{x}}_i) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \;\middle|\; \begin{array}{rcl} \underline{\mathbf{x}}^{(0)} &:=& \underline{\mathbf{x}}_i \\ \underline{\mathbf{a}}^{(t)} &\sim& \pi(\cdot\,|\,\underline{\mathbf{x}}^{(t)}) \\ \underline{\mathbf{x}}^{(t+1)} &\sim& P(\cdot\,|\,\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \end{array} \right] \\
&= \mathbb{E}\left[r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}^{(0)}) \;\middle|\; \underline{\mathbf{a}}^{(0)} \sim \pi(\cdot|\underline{\mathbf{x}}_i) \right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t\, r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \;\middle|\; \begin{array}{rcl} \underline{\mathbf{x}}^{(0)} &:=& \underline{\mathbf{x}}_i \\ \underline{\mathbf{a}}^{(t)} &\sim& \pi(\cdot\,|\,\underline{\mathbf{x}}^{(t)}) \\ \underline{\mathbf{x}}^{(t+1)} &\sim& P(\cdot\,|\,\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) \end{array} \right] \\
&= \mathbb{E}\left[r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}^{(0)}) \;\middle|\; \underline{\mathbf{a}}^{(0)} \sim \pi(\cdot|\underline{\mathbf{x}}_i) \right] + \gamma\, \mathbb{E}\left[V^\pi(\underline{\mathbf{x}}^{(1)}) \;\middle|\; \begin{array}{rcl} \underline{\mathbf{a}}^{(0)} &\sim& \pi(\cdot\,|\,\underline{\mathbf{x}}_i) \\ \underline{\mathbf{x}}^{(1)} &\sim& P(\cdot|\underline{\mathbf{x}}_i, \underline{\mathbf{a}}^{(0)}) \end{array} \right] \\
&= \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k\,|\,\underline{\mathbf{x}}_i) \Big(r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) + \gamma \sum_{j=1}^{S} P(\underline{\mathbf{x}}_j\,|\,\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)\, V^\pi(\underline{\mathbf{x}}_j) \Big)
\end{aligned}
$$

$\underline{\mathbf{x}}_i \in \{0,1\}^S$: 1-out-of-$S$ coded state i

The Bellman equation (2)

$$\begin{aligned} V^{\pi}(\underline{\mathbf{x}}_i) &= \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) \Big(r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) + \gamma \sum_{j=1}^{S} P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)\, V^{\pi}(\underline{\mathbf{x}}_j) \Big) \\ &= \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)}_{\text{``controlled'' reward function } r_i^{\pi}} + \gamma \sum_{j=1}^{S} \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)}_{\text{``controlled'' transition model } P_{ij}^{\pi}} V^{\pi}(\underline{\mathbf{x}}_j) \end{aligned}$$

$$\begin{aligned} \underline{\mathbf{v}}^{\pi} &= \underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\mathbf{v}}^{\pi}, \quad \text{with} \underbrace{\left\{ \begin{aligned} r_i^{\pi} &:= \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i)\, r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) \\ P_{ij}^{\pi} &:= \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i)\, P(\underline{\mathbf{x}}_j | \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) \end{aligned} \right.}_{\text{``controlled'' models } \underline{\mathbf{r}}^{\pi} \in \mathbb{R}^S \text{ and } \underline{\mathbf{P}}^{\pi} \in \mathbb{R}^{S \times S}} \\ &=: \hat{B}^{\pi}[\underline{\mathbf{v}}^{\pi}] \end{aligned}$$

$\underline{\mathbf{x}}_i \in \{0,1\}^S$: 1-out-of-$S$ coded state i, $\underline{\mathbf{v}}^{\pi} \in \mathbb{R}^S$: vector containing all values V^{π}

4.1.4 Model-based Approaches

The analytic solution of the Bellman equation

Bellman operator $\hat{B}^\pi$ for discrete state values

$$\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}] \quad = \quad \underline{\mathbf{r}}^\pi + \gamma \underline{\mathbf{P}}^\pi \underline{\tilde{\mathbf{v}}}\,, \qquad\qquad \forall \underline{\tilde{\mathbf{v}}} \in \mathbb{R}^S$$

- Bellman operator $\hat{B}^\pi$ of policy π uses "controlled" models
  - of the reward function $\underline{\mathbf{r}}^\pi \in \mathbb{R}^S$
  - and transition model $\underline{\mathbf{P}}^\pi \in \mathbb{R}^{S \times S}$

- $\hat{B}^\pi$ has an analytic solution of the value function $\underline{\mathbf{v}}^\pi \in \mathbb{R}^S$

$$\underline{\mathbf{v}}^\pi = \underline{\mathbf{r}}^\pi + \gamma \underline{\mathbf{P}}^\pi \underline{\mathbf{v}}^\pi \;\;\rightsquigarrow\;\; \big(\underline{\mathbf{I}} - \gamma \underline{\mathbf{P}}^\pi\big)\underline{\mathbf{v}}^\pi = \underline{\mathbf{r}}^\pi \;\;\rightsquigarrow\;\; \underline{\mathbf{v}}^\pi = \big(\underline{\mathbf{I}} - \gamma \underline{\mathbf{P}}^\pi\big)^{-1} \underline{\mathbf{r}}^\pi$$

- matrix $(\underline{\mathbf{I}} - \gamma \underline{\mathbf{P}}^\pi) \in \mathbb{R}^{S \times S}$ is always invertible
  - $|\lambda_k| \leq 1$ for all eigenvalues λ_k of transition matrices $\underline{\mathbf{P}}^\pi$
  - discount factor $\gamma < 1$

(see e.g. Bertsekas, 2007, for details)

Model-based value iteration

- the value function $\underline{\mathbf{v}}^{\pi}$ is the **fixed-point** of the Bellman operator $\hat{B}^{\pi}$

$$\underline{\mathbf{v}}^{\pi} \quad = \quad \hat{B}^{\pi}[\underline{\mathbf{v}}^{\pi}] \quad = \quad \underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\mathbf{v}}^{\pi}$$

- **value iteration**: repeated application of the Bellman operator

$$\underline{\tilde{\mathbf{v}}}^{\pi(t+1)} \quad = \quad \underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\tilde{\mathbf{v}}}^{\pi(t)}$$

- is value iteration convergent, i.e. $\lim\limits_{t \to \infty} \underline{\tilde{\mathbf{v}}}^{\pi(t)} = \underline{\mathbf{v}}^{\pi}$?

Convergence of value iteration

Contraction mapping (in supremum norm)

A function $\hat{B} : \mathbb{R}^S \to \mathbb{R}^S$ is called a *contraction mapping* with Lipschitz constant $\lambda < 1$ if $\max_j \left| (\hat{B}[\underline{\tilde{\mathbf{v}}}] - \hat{B}[\underline{\tilde{\mathbf{w}}}])_j \right| \leq \lambda \max_j \left| \tilde{v}_j - \tilde{w}_j \right|, \forall \underline{\tilde{\mathbf{v}}}, \underline{\tilde{\mathbf{w}}} \in \mathbb{R}^S$.

- application to the Bellman operator $\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}] = \underline{\mathbf{r}}^\pi + \gamma \underline{\mathbf{P}}^\pi \underline{\tilde{\mathbf{v}}}$

$$\begin{aligned} \max_j \left| \hat{B}^\pi[\underline{\tilde{\mathbf{v}}}]_j - \hat{B}^\pi[\underline{\tilde{\mathbf{w}}}]_j \right| &= \max_j \left| r_j^\pi + \gamma(\underline{\mathbf{P}}^\pi \underline{\tilde{\mathbf{v}}})_j - r_j^\pi - \gamma(\underline{\mathbf{P}}^\pi \underline{\tilde{\mathbf{w}}})_j \right| \\ &\overset{\text{(i)}}{\leq} \max_j \gamma \left(\underline{\mathbf{P}}^\pi |\underline{\tilde{\mathbf{v}}} - \underline{\tilde{\mathbf{w}}}| \right)_j \overset{\text{(ii)}}{\leq} \gamma \max_j |\tilde{v}_j - \tilde{w}_j| \end{aligned}$$

(i) $\left| \sum_{i=1}^S P_{ji}^\pi x_i \right| \leq \sum_{i=1}^S P_{ji}^\pi |x_i|, \quad \forall \underline{\mathbf{x}} \in \mathbb{R}^S$ (Jensen's inequality)

(ii) $\sum_{i=1}^S P_{ji}^\pi |x_i| \leq \sum_{i=1}^S P_{ji}^\pi \max_{1 \leq k \leq S} |x_k| = \max_{1 \leq k \leq S} |x_k|$ $\quad (\sum_{i=1}^S P_{ji}^\pi = 1)$

Convergence of value iteration

Contraction mapping (in supremum norm)

A function $\hat{B} : \mathbb{R}^S \to \mathbb{R}^S$ is called a *contraction mapping* with Lipschitz constant $\lambda < 1$ if $\max_j \left| (\hat{B}[\underline{\tilde{\mathbf{v}}}] - \hat{B}[\underline{\tilde{\mathbf{w}}}])_j \right| \leq \lambda \max_j \left| \tilde{v}_j - \tilde{w}_j \right|, \forall \underline{\tilde{\mathbf{v}}}, \underline{\tilde{\mathbf{w}}} \in \mathbb{R}^S$.

- $\hat{B}^\pi$ is a contraction mapping with Lipschitz constant γ

$$\underline{\tilde{\mathbf{v}}}^{\pi(t+1)} = \hat{B}^\pi[\underline{\tilde{\mathbf{v}}}^{\pi(t)}] = \underline{\mathbf{r}}^\pi + \gamma \underline{\mathbf{P}}^\pi \underline{\tilde{\mathbf{v}}}^{\pi(t)} \qquad \text{(value iteration)}$$

- $\max_j \left| (\hat{B}^\pi[\underline{\mathbf{v}}^{\pi(t)}])_j - v_j^\pi \right| \leq \gamma \max_j \left| v_j^{\pi(t)} - v_j^\pi \right|$ for *any* $\underline{\mathbf{v}}^{\pi(t)} \in \mathbb{R}^S$

$\Rightarrow$ value iteration converges in the limit to unique fix-point $\underline{\mathbf{v}}^\pi$

- number of iterations until convergence $\sim -\frac{1}{\log(\gamma)}$
- analytic solution is faster for large γ

4.1.5 Model-free Approaches: Online Value Estimation

Inductive value estimation

- agent must learn through interaction with the environment
  - "controlled" models $\underline{\mathbf{r}}^\pi$ and $\underline{\mathbf{P}}^\pi$ are not available

$$V^\pi(\underline{\mathbf{x}}_i) \quad = \quad \sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i)\Big(r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) + \gamma \sum_{j=1}^{S} P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)\, V^\pi(\underline{\mathbf{x}}_j)\Big)$$

- estimate value function *inductively* from one long Markov chain
  - actions are drawn according to the policy $\underline{\mathbf{a}}^{(t)} \sim \pi(\cdot|\,\underline{\mathbf{x}}^{(t)})$
  - which lead to transitions $\underline{\mathbf{x}}^{(t+1)} \sim P(\cdot|\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)})$
  - and yield rewards $r_t := r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)})$

Temporal difference (TD) learning

- online estimation named after the difference in values (TD-error ΔV_t)

$$\tilde{V}^{\pi}_{t+1}(\underline{\mathbf{x}}^{(t)}) \quad = \quad \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)}) \; + \; \eta\Big(\underbrace{r_t + \gamma \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t+1)}) - \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)})}_{\text{TD-error } \Delta V_t}\Big)$$

- TD learning performs value iteration *on average*
  - for the average over all Markov chains that pass $\underline{\mathbf{x}}_i$ at time t holds:

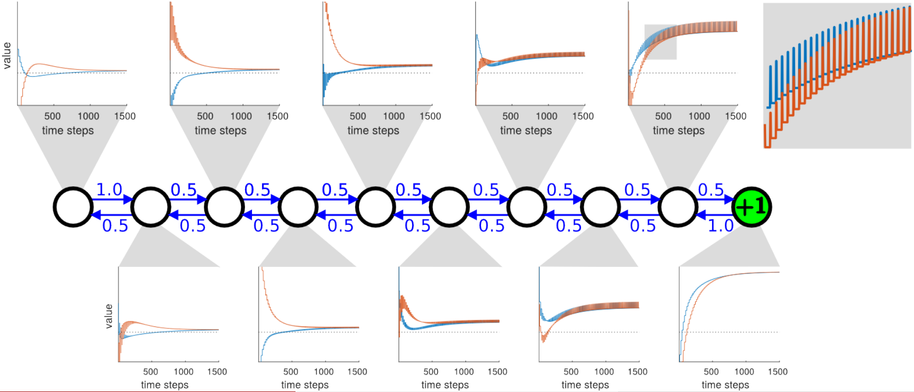
$$\underbrace{\mathbb{E}\Big[\tilde{V}^{\pi}_{t+1}(\underline{\mathbf{x}}^{(t)})\Big]}_{\tilde{v}_i^{\pi(t+1)}} \quad = \quad (1-\eta)\,\underbrace{\mathbb{E}\Big[\tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)})\Big]}_{\tilde{v}_i^{\pi(t)}} + \eta\Big(\underbrace{\mathbb{E}[r_t] + \gamma \mathbb{E}\Big[\tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t+1)})\Big]}_{(\underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\tilde{\mathbf{v}}}^{\pi(t)})_i}\Big)$$

- *asynchronous online estimate* of $\hat{B}^{\pi}[\underline{\tilde{\mathbf{v}}}^{\pi(t)}] = \underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\tilde{\mathbf{v}}}^{\pi(t)}$
  - asynchronous update of one state at a time
  - estimates Bellman operator $\hat{B}^{\pi}$ by online average

(see Sutton and Barto, 1998)

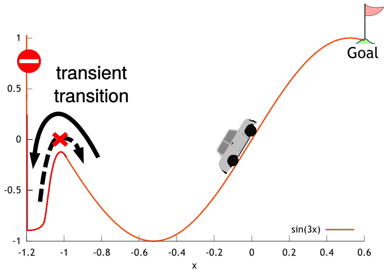
Convergence of TD learning

- **example:** Markov chain running back and forth on 10 states
  - two randomly initialized value functions (red/blue)
  - deterministic transitions with stochastic policy
  - rightmost state is rewarded, $\gamma = 0.95$, $\eta = 0.5$
- TD learning **contracts** different initializations, but does **not converge**



Requirements for contraction

- TD learning contracts
  - for an infinite Markov chain,
  - which visits *all* states infinitely often



- no **transient** transitions allowed
  - transitions must be reversible
  - "you cannot learn from death"

- **positive recurrence**: a non-zero probability to return in finite time

Ergodic Markov chains

Ergodicity

A Markov chain is **ergodic** if it is **positivly recurrent** (non-zero probability to leave any state and eventually return to it) and **aperiodic** (returns to the same state can occur at irregular times).

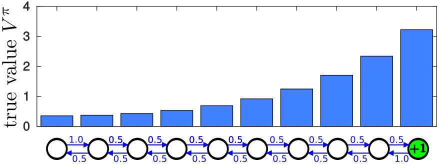
- **steady state distribution** $P_{ss}(\underline{\mathbf{x}}) > 0$ exists and visits all states $\underline{\mathbf{x}}$

$\Rightarrow$ TD learning is a contraction mapping for *ergodic* Markov chains

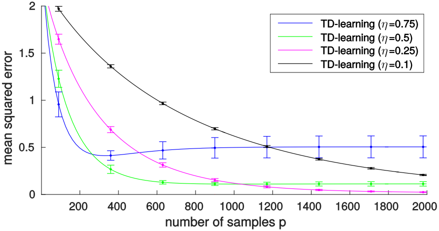
Influence of learning rate η

$$\begin{aligned}\tilde{V}^{\pi}_{t+1}(\underline{\mathbf{x}}^{(t)}) &= \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)}) \;+\; \eta\,\Delta V_t \\ \Delta V_t &= r_t + \gamma \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t+1)}) - \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)})\end{aligned}$$

- stochastic transitions/rewards $\leadsto \tilde{V}^{\pi}_{t}$ may not converge
- TD learning let $\tilde{V}^{\pi}_{t}$ fluctuate around the true value function V^{π}
- influence of the learning rate η
  - large η: fast learning, large variance
  - small η: slow learning, small variance
  - decaying η_t are not practical as ΔV_t are (initially) not stationary



- 10 states Markov chain
- regular movement back and forth
- rightmost state rewarded, $\gamma = 0.95$



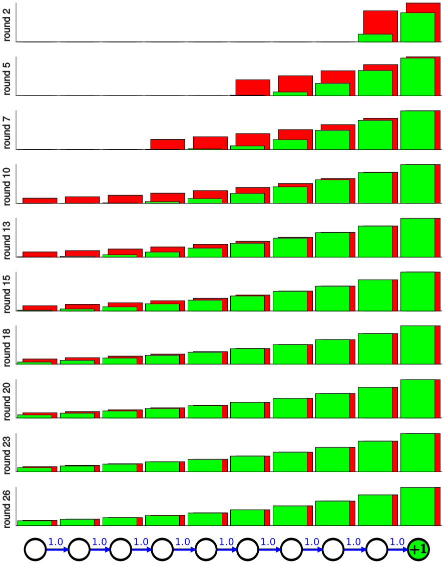
4.1.6 Model-free Approaches: Eligibility Traces & TD(λ)

Value propagation in TD learning

$$\tilde{V}^{\pi}_{t+1}(\underline{\mathbf{x}}^{(t)}) = \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)}) + \eta \, \Delta V_t$$

$$\Delta V_t = r_t + \gamma \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t+1)}) - \tilde{V}^{\pi}_{t}(\underline{\mathbf{x}}^{(t)})$$

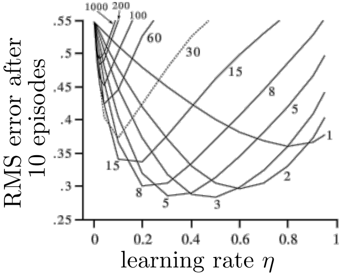
- TD learning propagates values one step into the past
  - many steps to convergence
- deterministic example:
  - 10 states, 1 action
  - only forward transitions
  - reward in last state
  - $\gamma = 0.9$; $\eta = 1$ or $\eta = 0.5$
- value propagation requires
  - exactly 10 rounds ($\eta = 1$)
  - roughly 26 rounds ($\eta = 0.5$)



n-step temporal difference learning

- accumulation of observed rewards

$$
\begin{aligned}
R_t^{(1)} &= r_t + \gamma \tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t+1)}) \\
R_t^{(2)} &= r_t + \gamma r_{t+1} + \gamma^2 \tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t+2)}) \\
&\vdots \\
R_t^{(n)} &= \sum_{\tau=0}^{n-1} \gamma^\tau r_{t+\tau} + \gamma^n \tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t+n)})
\end{aligned}
$$



RMS averaged over 100 random-walks on a 19-state chain, rewarded at one end

(Sutton and Barto, 1998)

- online estimation similar to TD learning

$$
\tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t)}) \quad \leftarrow \quad \tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t)}) \; + \; \eta \Big(R_t^{(n)} - \tilde{V}_t^\pi(\underline{\mathbf{x}}^{(t)}) \Big)
$$

Discounted average

$$\tilde{V}_t^{\pi}(\underline{\mathbf{x}}^{(t)}) \quad \leftarrow \quad \tilde{V}_t^{\pi}(\underline{\mathbf{x}}^{(t)}) \;+\; \eta\Big(R_t^{(n)} - \tilde{V}_t^{\pi}(\underline{\mathbf{x}}^{(t)})\Big)$$

- there is an optimal combination of η and n, however,
  - agent must memorize the last n steps
  - values are updated with a delay of n steps

- trick: consider a discounted average of $R_t^{(n)}$

$$R_t^{\lambda} \quad = \quad (1-\lambda)\sum_{k=0}^{\infty} \lambda^k \, R_t^{(k+1)}$$

Eligibility traces & TD(λ)

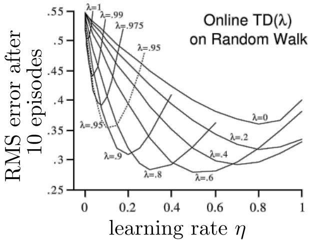
- the **eligibility trace** $\underline{\mathbf{e}}^{(t)} \in \mathbb{R}^S$ stores the past of state $\underline{\mathbf{x}}_i$

$$e_i^{(t)} = \sum_{k=0}^{t} (\gamma\lambda)^{t-k}\, \delta_{ik}\,, \qquad \delta_{ik} = \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(k)} \qquad \forall \underline{\mathbf{x}}_i \in \mathcal{X}\,,$$

- The **TD(λ) method**:

$$\begin{aligned} \tilde{V}^\pi_{t+1}(\underline{\mathbf{x}}_i) &= \tilde{V}^\pi_t(\underline{\mathbf{x}}_i) + \eta\, e_i \Big(\overbrace{r_t + \gamma \tilde{V}^\pi_t(\underline{\mathbf{x}}^{(t+1)}) - \tilde{V}^\pi_t(\underline{\mathbf{x}}^{(t)})}^{\text{TD-error } \Delta V_t} \Big) \\ \underline{\mathbf{e}}^{(t+1)} &= \gamma\, \lambda\, \underline{\mathbf{e}}^{(t)} + \underline{\mathbf{x}}^{(t+1)} \end{aligned}$$

- TD(0): TD learning as defined before

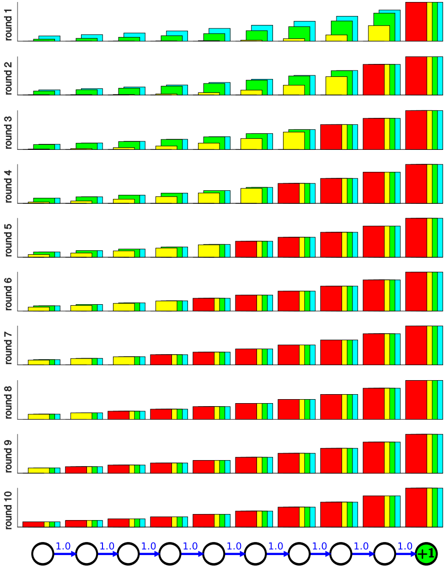


RMS averaged over 100 random-walks on a 19-state chain, rewarded at one end

(Sutton and Barto, 1998)

Value propagation in TD(λ)

- deterministic example:
  - 10 states, 1 action
  - only forward transitions
  - reward in last state
  - $\gamma = 1$, $\eta = 1$
- value propagation finishes
  - after 1 round with $\lambda = 1$
  - after 4 rounds with $\lambda = 0.9$
  - after 7 rounds with $\lambda = 0.5$
  - after 10 rounds with $\lambda = 0$



4.1.7 Model-free approaches: Batch Value Estimation

Reminder: the Bellman equation

Richard E. Bellman
(1920–1984)

$$V^{\pi}(\underline{\mathbf{x}}_i) = \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)}_{\text{“controlled” reward function } r_i^{\pi}} + \gamma \sum_{j=1}^{S} \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)}_{\text{“controlled” transition model } P_{ij}^{\pi}} V^{\pi}(\underline{\mathbf{x}}_j)$$

$$\underline{\mathbf{v}}^{\pi} = \hat{B}^{\pi}[\underline{\mathbf{v}}^{\pi}] = \underline{\mathbf{r}}^{\pi} + \gamma \underline{\mathbf{P}}^{\pi} \underline{\mathbf{v}}^{\pi}$$

$\underline{\mathbf{x}}_i \in \{0,1\}^S$: 1-out-of-$S$ coded state i, $\underline{\mathbf{r}}^{\pi} \in \mathbb{R}^S$ “controlled” reward function,

$\underline{\mathbf{v}}^{\pi} \in \mathbb{R}^S$: vector containing all values V^{π} $\underline{\mathbf{P}}^{\pi} \in \mathbb{R}^{S \times S}$ “controlled” transition model

Batch approximation of the Bellman operator (1)

- approximate $\tilde{V}^\pi_{t+1} \approx \hat{B}^\pi[\tilde{V}^\pi_t]$ using samples from an ergodic Markov chain $\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}\}_{t=0}^p$, executing policy π

$$\hat{B}^\pi[\tilde{V}^\pi_t](\underline{\mathbf{x}}_i) \;=\; \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i) \Big(r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) + \gamma \overbrace{\sum_{j=1}^{S} P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)\, \tilde{V}^\pi_t(\underline{\mathbf{x}}_j)}^{\text{average } \{\tilde{V}^\pi_t(\underline{\mathbf{x}}^{(t+1)}) | \underline{\mathbf{x}}^{(t)} = \underline{\mathbf{x}}_i\}} \Big)}_{\text{approximate by averaging over } \{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)} \,|\, \underline{\mathbf{x}}^{(t)} = \underline{\mathbf{x}}_i, \underline{\mathbf{a}}^{(t)} \sim \pi\}}$$

Batch approximation of the Bellman operator (1)

- approximate $\tilde{V}^\pi_{t+1} \approx \hat{B}^\pi[\tilde{V}^\pi_t]$ using samples from an ergodic Markov chain $\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}\}_{t=0}^{p}$, executing policy π

$$\hat{B}^\pi[\tilde{V}^\pi_t](\underline{\mathbf{x}}_i) \;=\; \underbrace{\sum_{k=1}^{A} \pi(\underline{\mathbf{a}}_k \,|\, \underline{\mathbf{x}}_i)\Big(r(\underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k) + \gamma \overbrace{\sum_{j=1}^{S} P(\underline{\mathbf{x}}_j \,|\, \underline{\mathbf{x}}_i, \underline{\mathbf{a}}_k)\, \tilde{V}^\pi_t(\underline{\mathbf{x}}_j)}^{\text{average } \{\tilde{V}^\pi_t(\underline{\mathbf{x}}^{(t+1)}) | \underline{\mathbf{x}}^{(t)} = \underline{\mathbf{x}}_i\}}\Big)}_{\text{approximate by averaging over } \{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)} \,|\, \underline{\mathbf{x}}^{(t)} = \underline{\mathbf{x}}_i, \underline{\mathbf{a}}^{(t)} \sim \pi\}}$$

$$\approx \; \underbrace{\frac{1}{\sum_{\tau=0}^{p-1} \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(\tau)}}}_{\text{normalization}} \sum_{t=0}^{p-1} \underbrace{\underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(t)}}_{\text{selection}} \Big(r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) + \gamma \tilde{V}^\pi_t(\underline{\mathbf{x}}^{(t+1)}) \Big)$$

- selection $\underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(t)}$ applies update only for states $\underline{\mathbf{x}}_i = \underline{\mathbf{x}}^{(t)}$
- normalization $\sum_{\tau=0}^{p-1} \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(\tau)}$ counts how often $\underline{\mathbf{x}}_i$ appears in batch

Batch approximation of the Bellman operator (2)

- approximate $\tilde{V}^\pi_{t+1} \approx \hat{B}^\pi[\tilde{V}^\pi_t]$ using samples from an ergodic Markov chain $\{\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}\}_{t=0}^p$, executing policy π

$$\hat{B}^\pi[\tilde{V}^\pi](\underline{\mathbf{x}}_i) \approx \frac{1}{\sum_{\tau=0}^{p-1} \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(\tau)}} \sum_{t=0}^{p-1} \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(t)} \Big(r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)}) + \gamma \tilde{V}^\pi(\underline{\mathbf{x}}^{(t+1)}) \Big)$$

$$= \underline{\mathbf{x}}_i^\top \Big(\underbrace{\sum_{\tau=0}^{p-1} \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(\tau)}}_{C_{ii}} \Big)^{-1} \Big(\underbrace{\sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} r(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)})}_{\underline{\mathbf{b}}} + \gamma \underbrace{\sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} \tilde{V}^\pi(\underline{\mathbf{x}}^{(t+1)})}_{\underline{\mathbf{D}}^\pi \underline{\tilde{\mathbf{v}}}^\pi} \Big)$$

$$\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}^\pi] \approx \underline{\mathbf{C}}^{-1}\big(\underline{\mathbf{b}} + \gamma \underline{\mathbf{D}}^\pi \underline{\tilde{\mathbf{v}}}^\pi\big)$$

$\underline{\mathbf{C}} = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} (\underline{\mathbf{x}}^{(t)})^\top \in \mathbb{R}^{S\times S}$
diagonal normalization matrix

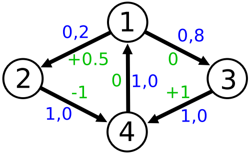
$\underline{\mathbf{D}}^\pi = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} (\underline{\mathbf{x}}^{(t+1)})^\top \in \mathbb{R}^{S\times S}$
matrix to count transitions

$\underline{\mathbf{b}} = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} r_{(\underline{\mathbf{x}}^{(t)}, \underline{\mathbf{a}}^{(t)})} \in \mathbb{R}^S$
vector with the sum of rewards

Example batch approximation

- the approximated Bellman operator:

$$\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}^\pi] \quad \approx \quad \underline{\mathbf{C}}^{-1}\left(\underline{\mathbf{b}} + \gamma \underline{\mathbf{D}}^\pi \underline{\tilde{\mathbf{v}}}^\pi\right)$$



- example MDP with 4 states:
  - transition probabilities $\underline{\mathbf{P}}^\pi \approx \underline{\mathbf{C}}^{-1}\underline{\mathbf{D}}^\pi$
  - reward for transitions $\underline{\mathbf{r}}^\pi \approx \underline{\mathbf{C}}^{-1}\underline{\mathbf{b}}$

$$\underline{\mathbf{C}} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \qquad \underline{\mathbf{D}}^\pi = \begin{bmatrix} 0 & 3 & 7 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 7 \\ 10 & 0 & 0 & 0 \end{bmatrix}, \qquad \underline{\mathbf{b}} = \begin{bmatrix} 1.5 \\ -3 \\ +7 \\ 0 \end{bmatrix}$$

state visit count — transition count — collected rewards

$$\underline{\mathbf{C}} = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)}(\underline{\mathbf{x}}^{(t)})^\top \in \mathbb{R}^{S\times S} \quad \underline{\mathbf{D}}^\pi = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)}(\underline{\mathbf{x}}^{(t+1)})^\top \in \mathbb{R}^{S\times S} \quad \underline{\mathbf{b}} = \sum_{t=0}^{p-1} \underline{\mathbf{x}}^{(t)} r_{(\underline{\mathbf{x}}^{(t)},\underline{\mathbf{a}}^{(t)})} \in \mathbb{R}^S$$

Solution to the approximated Bellman operator

- the approximated Bellman operator:

$$\hat{B}^{\pi}[\underline{\tilde{\mathbf{v}}}^{\pi}] \quad \approx \quad \underline{\mathbf{C}}^{-1}\big(\underline{\mathbf{b}} + \gamma \underline{\mathbf{D}}^{\pi} \underline{\tilde{\mathbf{v}}}^{\pi}\big)$$

- fixed-point $\underline{\mathbf{v}}^* \approx \hat{B}^{\pi}[\underline{\mathbf{v}}^*]$ can be computed analytically

$$\underline{\mathbf{v}}^* \;=\; \big(\underline{\mathbf{C}} - \gamma \underline{\mathbf{D}}^{\pi}\big)^{-1} \underline{\mathbf{b}}$$

Solution to the approximated Bellman operator

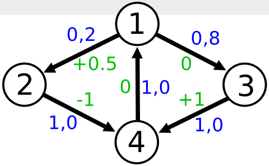
- the approximated Bellman operator:

$$\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}^\pi] \quad \approx \quad \underline{\mathbf{C}}^{-1}\big(\underline{\mathbf{b}} + \gamma \underline{\mathbf{D}}^\pi \underline{\tilde{\mathbf{v}}}^\pi\big)$$

- fixed-point $\underline{\mathbf{v}}^* \approx \hat{B}^\pi[\underline{\mathbf{v}}^*]$ can be computed analytically

$$\underline{\mathbf{v}}^* \;=\; \big(\underline{\mathbf{C}} - \gamma \underline{\mathbf{D}}^\pi\big)^{-1}\underline{\mathbf{b}} \;=\; \big(\underline{\mathbf{I}} - \gamma\,\underbrace{\underline{\mathbf{C}}^{-1}\underline{\mathbf{D}}^\pi}_{\underline{\tilde{\mathbf{P}}}^\pi}\big)^{-1}\underbrace{\underline{\mathbf{C}}^{-1}\underline{\mathbf{b}}}_{\underline{\tilde{\mathbf{r}}}^\pi} \;=\; \big(\underline{\mathbf{I}} - \gamma \underline{\tilde{\mathbf{P}}}^\pi\big)^{-1}\underline{\tilde{\mathbf{r}}}^\pi$$

Solution to the approximated Bellman operator



- the approximated Bellman operator:

$$\hat{B}^\pi[\underline{\tilde{\mathbf{v}}}^\pi] \quad \approx \quad \underline{\mathbf{C}}^{-1}\big(\underline{\mathbf{b}} + \gamma \underline{\mathbf{D}}^\pi \underline{\tilde{\mathbf{v}}}^\pi\big)$$

- fixed-point $\underline{\mathbf{v}}^* \approx \hat{B}^\pi[\underline{\mathbf{v}}^*]$ can be computed analytically

$$\underline{\mathbf{v}}^* \;=\; \big(\underline{\mathbf{C}} - \gamma \underline{\mathbf{D}}^\pi\big)^{-1} \underline{\mathbf{b}} \;=\; \big(\underline{\mathbf{I}} - \gamma \underbrace{\underline{\mathbf{C}}^{-1}\underline{\mathbf{D}}^\pi}_{\underline{\tilde{\mathbf{P}}}^\pi}\big)^{-1} \underbrace{\underline{\mathbf{C}}^{-1}\underline{\mathbf{b}}}_{\underline{\tilde{\mathbf{r}}}^\pi} \;=\; \big(\underline{\mathbf{I}} - \gamma \underline{\tilde{\mathbf{P}}}^\pi\big)^{-1} \underline{\tilde{\mathbf{r}}}^\pi$$

- equivalent to empirically estimated model-based solution

$$\underline{\tilde{\mathbf{P}}}^\pi \;=\; \underline{\mathbf{C}}^{-1}\underline{\mathbf{D}}^\pi = \begin{bmatrix} 0 & .3 & .7 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \qquad \underline{\tilde{\mathbf{r}}} \;=\; \underline{\mathbf{C}}^{-1}\underline{\mathbf{b}} = \begin{bmatrix} .15 \\ -1 \\ +1 \\ 0 \end{bmatrix}$$

- in the limit convergence to V^π for ergodic Markov chains
  - $\underline{\tilde{\mathbf{P}}}^\pi \to \underline{\mathbf{P}}^\pi$ and $\underline{\tilde{\mathbf{r}}}^\pi \to \underline{\mathbf{r}}^\pi$ if all states are visited infinitely often

Comparison of batch and online value estimation

- different reward propagation

TD(0): one time step into the past
TD(λ): all λ-discounted steps
batch: instantaneous everywhere

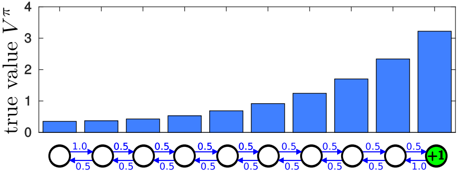
- different convergence behavior

TD(0): fluctuates around V^π
TD(λ): fluctuates around V^π
batch: converges exactly to V^π

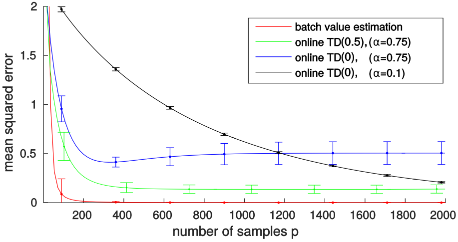
- different complexities (computational and memory)

TD(0): $\mathcal{O}(p)$ and $\mathcal{O}(S)$
TD(λ): $\mathcal{O}(pS)$ and $\mathcal{O}(S)$
batch: $\mathcal{O}(p + S^3)$ and $\mathcal{O}(S^2)$

S: number of states, p: number of samples

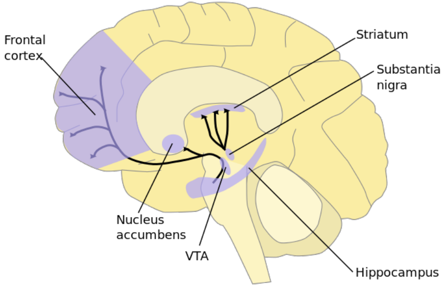


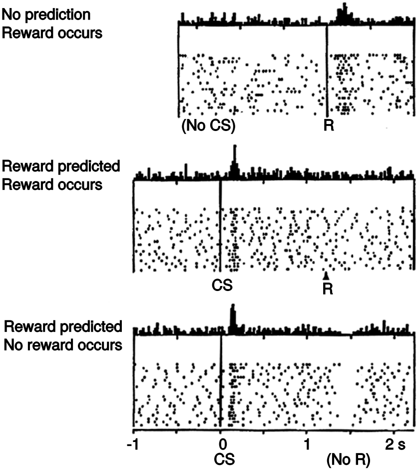
- Markov chain back and forth
- only last state rewarded
- value estimated for $\gamma = 0.95$



Neurological relevance of reinforcement learning

- dopamine neurons encode online TD-errors in most mammals

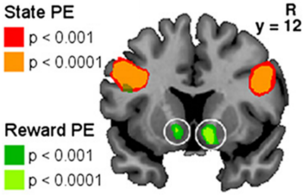




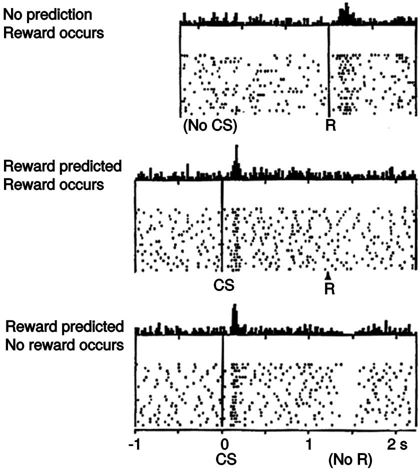
(Schultz et al., 1997)

Neurological relevance of reinforcement learning

- dopamine neurons encode online TD-errors in most mammals
- model-based prediction errors were found in human *pre-frontal cortex*



(Gläscher et al., 2010)



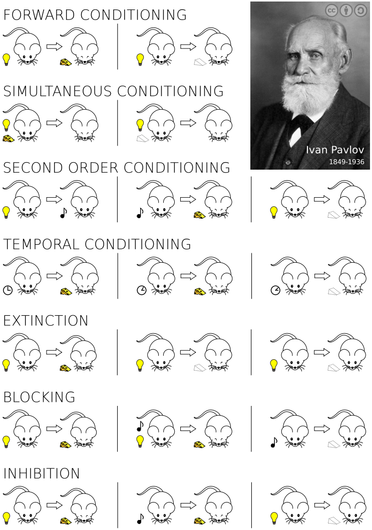
(Schultz et al., 1997)

End of Section 4.1

the following slides contain

OPTIONAL MATERIAL

The many faces of classical conditioning



Contraction properties of TD learning

- asynchronous TD update at time t for all states $\underline{\mathbf{x}}_i$:
  - let $v_t := \underline{\mathbf{v}}^\top \underline{\mathbf{x}}^{(t)}$ and $\mu_{it} = \underline{\mathbf{x}}_i^\top \underline{\mathbf{x}}^{(t)}$

$$\hat{B}_t^\pi[\underline{\mathbf{v}}]_i \;:=\; v_i + \eta \underbrace{\mu_{it}\left(r_t + \gamma v_{t+1} - v_t\right)}_{\text{TD-error } \Delta v_t \text{ if } \underline{\mathbf{x}}_i = \underline{\mathbf{x}}^{(t)}}$$

- $\hat{B}_t^\pi$ is *in general* a **non-expansion**

$$\max_{1 \le i \le S} \left| \hat{B}_t^\pi[\underline{\mathbf{v}}]_i - \hat{B}_t^\pi[\underline{\mathbf{w}}]_i \right| \quad \le \quad \max_{1 \le i \le S} |v_i - w_i|$$

- $\hat{B}_t^\pi$ is *sometimes* a **contraction mapping**
  - in states $\underline{\mathbf{x}}^{(t)}$ with $|v_t - w_t| \ge \max_{i \ne t} |v_i - w_i|$

$$\left| \hat{B}_t^\pi[\underline{\mathbf{v}}]_t - \hat{B}_t^\pi[\underline{\mathbf{w}}]_t \right| \quad \le \quad (1 - \eta(1-\gamma)) \left| v_t - w_t \right|$$

Temporal difference learning with eligibility traces: TD(λ)

The TD(λ) algorithm

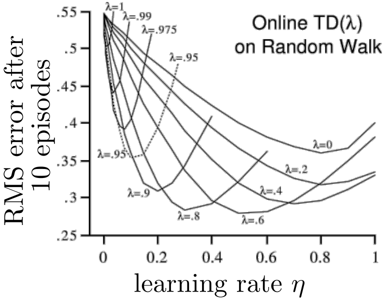
for $t \in \{0, \dots, p-1\}$ **do**

$$\Delta v \leftarrow r_t + \gamma \underline{\mathbf{v}}^\top \underline{\mathbf{x}}^{(t+1)} - \underline{\mathbf{v}}^\top \underline{\mathbf{x}}^{(t)}$$ // TD-error Δv at time t

$$\underline{\mathbf{v}} \leftarrow \underline{\mathbf{v}} + \eta \, \Delta v \, \underline{\mathbf{e}}$$ // update all visited states

$$\underline{\mathbf{e}} \leftarrow \gamma \, \lambda \, \underline{\mathbf{e}} + \underline{\mathbf{x}}^{(t+1)}$$ // update eligibility trace $\underline{\mathbf{e}}$

end



- TD(0): TD learning as defined before

RMS averaged over 100 random-walks on a 19-state chain, rewarded at one end

(Sutton and Barto, 1998)

References I

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition, 2007.

Peter Dayan. The role of value systems in decision making. In *Better Than Conscious?: Decision Making, the Human Mind, and Implications for Institutions*. MIT Press, 2008. ISBN 0-262-19580-1.

Peter Dayan and Yael Niv. Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology*, 18:185–196, 2008.

Jan Gläscher, Nathaniel Daw, Peter Dayan, and John P. O'Doherty. States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, 66:585–595, 2010.

Wolfram Schultz, Peter Dayan, and P. Read Montague. A Neural Substrate of Prediction and Reward. *Science*, 275(5306):1593–1599, March 1997. ISSN 1095-9203.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.