RL 7: The Bellman Equation

Michael Herrmann

University of Edinburgh, School of Informatics

02/02/2016

Last time: Markovian Decision Problems (MDPs)

- Markovian property

$$P(X_{t+1}=j|X_t=i,X_{t-1}=k_{t-1},\ldots,X_0=k_0)=P(X_{t+1}=j|X_t=i)$$

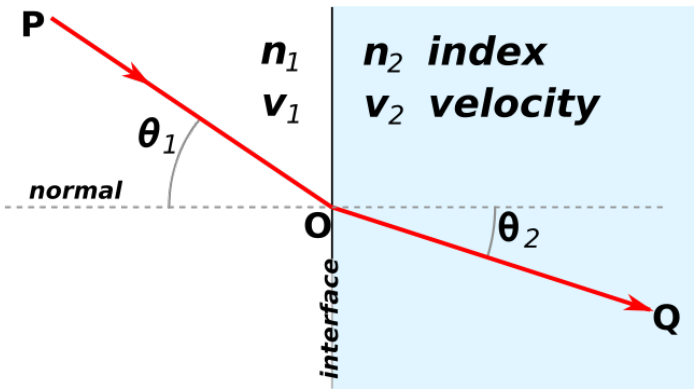
- finite-state Markov chain is characterised by transition matrix $\{p_{ij}\}$ and initial probabilities $\varpi = P\{X_0 = i\}$
- *stationary* $p_{ij} = P(X_{t+1} = j|X_t = i) = P(X_1 = j|X_0 = i)$
- j is *accessible* from i if $p_{ij}^{(n)} > 0$ (for some n)
- if j is accessible from i and v.v., then i and j *communicate*
- *irreducible* if every pair of states communicate
- if $\sum_{n=1}^{\infty} f_{ii}^{(n)} = 1$ and $\sum_{n=1}^{\infty} n\, f_{ij}^{(n)} < \infty$ then *positive recurrent*
- positive recurrent states that are aperiodic are *ergodic*
- if irreducible and ergodic $\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j^*$ (eVec zu eVal 1)
- $E(C) = \sum_{i=0}^{M} C_{ik}\pi_i^*$
- Change $\{p_{ij}\}$ such that $E(C)$ is minimal (or maximal)

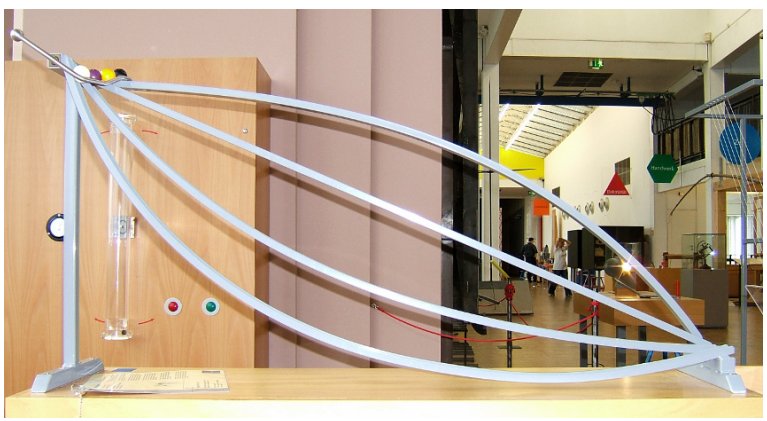
The reward hypothesis (Sutton)

- That all of what we mean by goals and purposes can be well thought of as the maximisation of the cumulative sum of a received scalar signal (reward)

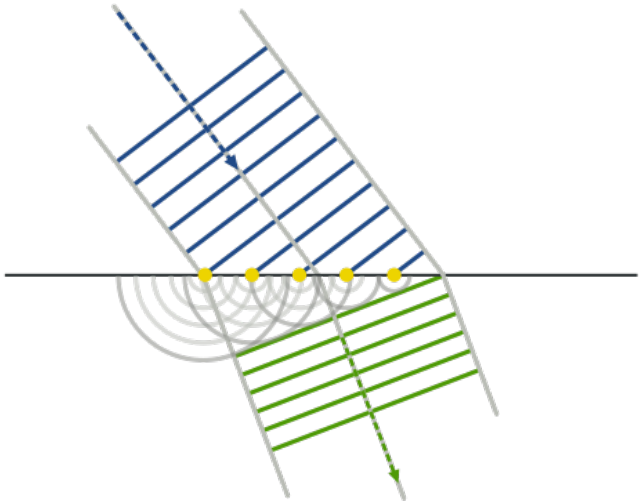
Optimality in Physics

Fermat's principle:
principle of least time





Huygens–Fresnel principle:
A wave front consists of elementary waves



Brachistochrone curve
(Johann Bernoulli, 1696)
$\Rightarrow$ Calculus of variations

Optimal Control

x: state	C: scalar cost	u: control
V: value	D: value of final state	T: total time

Value (or energy, time, action) as function of starting state

$$V(x(0), 0) = \min_u \left\{ \int_0^T C[x(t), u(t)]\, dt + D[x(T)] \right\}$$

Hamilton-Jacobi-Bellman equation

$$\dot{V}(x, t) + \min_u \{\nabla_x V(x, t) \cdot F(x, u) + C(x, u)\} = 0$$

with $\dot{x}(t) = F[x(t), u(t)]$ determining the evolution of the state

From wikipedia

Bellman Optimality

Bellman's Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. (1957)

Formulate learning problem such that the principle can be applied.

Markov reward processValue functions: Definition

Policy $\pi : \mathcal{S} \to \mathcal{A}$ or $a \sim \pi(\cdot|s)$ (means: $P(a_t = a|s_t) = \pi(a|s_t)$)

Value function $V^\pi : \mathcal{S} \to \mathbb{R}$

$$V^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s\right]$$

assuming that the initial probability $\varpi^0(s_0) > 0$

This is assuming an MDP with fixed π: $\left(\mathcal{S}, P^\pi, \varpi^0\right)$ which is extended to $\left(\mathcal{S}, P^\pi, \varpi^0, \mathcal{R}\right)$. The latter is a *Markov reward process* which arises naturally by assigning a reward distribution $R(\cdot|s)$ to each state s or to each state-action pair according to

$$R^\pi(r|s) = \sum_{a \in \mathcal{A}} \pi(a|s) R(r|s,a)$$

[ϖ is called "script-pi" and is used to avoid confusion with the policy π]

Value functions for state-action pairs

Policy $\pi : \mathcal{S} \to \mathcal{A}$ or $a \sim \pi(\cdot|s)$

Value function $\mathcal{Q}^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

$$\mathcal{Q}^\pi(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a\right]$$

assuming that the initial probability $\varpi^0(s_0) > 0$ and that $\pi(s_0) = a_0$ (deterministic) or $\pi(s_0, a_0) > 0$ (stochastic).

First action a_0 is applied now, later actions are chosen by π.

Optimal value functions

For MDPs an optimal policy can be defined as ($s \in \mathcal{S}$, policy π)

$$V^*(s) = \sup_\pi V^\pi(s)$$

Analogously, for state-action pairs we have ($s \in \mathcal{S}$ and $a \in \mathcal{A}$)

$$\begin{aligned} V^*(s) &= \sup_{a \in \mathcal{A}} Q^*(s,a) \\ \mathcal{Q}^*(s,a) &= r(s,a) + \gamma \sum_{u \in \mathcal{S}} P^{a=\pi(s)}(s,u)\, V^*(u) \end{aligned}$$

Optimality of $\mathcal{Q}$

Suppose π satisfies

$$\sum_{a \in \mathcal{A}} \pi(a|s)\, \mathcal{Q}^*(s,a) = V^*(s)$$

for all $s \in \mathcal{S}$. Then π is optimal.

Namely, $\pi(\cdot|s)$ selects the action(s) that maximise(s) $\mathcal{Q}^*(s,\cdot)$.

So, optimality implies greediness and knowing $\mathcal{Q}^*(s,a)$ allows us to act optimally.

Analogously, knowing V^*, r and P suffices to act optimally.

At each time step the algorithm causes a change of the estimate of the value function, we express this by the *operator* T:

$$V_{t+1} = T\, V_t$$

Can we expect global convergence (starting from arbitrary V_0)?

Similarly, for state-action values functions

$$\mathcal{Q}_{t+1} = T\, \mathcal{Q}_t$$

Once V_t (or $\mathcal{Q}_t$) is close to the optimal V^* (or $\mathcal{Q}^*$) then the greedy policy is close to optimal, more specifically, using the relation between V and $\mathcal{Q}_t$:

Suppose $\mathcal{Q}$ is fixed and π is the greedy policy w.r.t. $\mathcal{Q}$. Then

$$V^\pi(s) \geq V^*(s) - \frac{2}{1-\gamma}\|\mathcal{Q} - \mathcal{Q}^*\|_\infty$$

Singh and Yee, 1994

Bellman Equations for deterministic policies in an MDP

How to find the value of a policy? Requiring consistent V:

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{u \in \mathcal{S}} P(s, \pi(s), u) V^{\pi}(u)$$

This is the Bellman equation T for V^{π}.

Define the Bellman operator for π as $T^{\pi} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ (maps value functions to value functions)

$$(T^{\pi} V)(s) = r(s, \pi(s)) + \gamma \sum_{u \in \mathcal{S}} P(s, \pi(s), u) V(u)$$

Then naturally,

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

which is nothing but a compact formulation of the equation on top of this slide. This is a linear equation in V^{π} and T^{π}.

If $0 < \gamma < 1$ then T^{π} is a contraction w.r.t. the maximum norm.

Bellman optimality equations

How to characterise the optimal policy? Use the Bellman optimality principle.

$$V^*(s) = \sup_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{u \in \mathcal{S}} P(s,a,u) V^*(u) \right)$$

Bellman optimality operator $T^* : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$

$$(T^* V)(s) = \sup_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{u \in \mathcal{S}} P(s,a,u) V(u) \right)$$

Then naturally,

$$T^* V^* = V^*$$

which is nothing but a compact formulation of the equation on top of this slide.

If $0 < \gamma < 1$ then T^* is a contraction w.r.t. the maximum norm.

Bellman Operators for state-action value functions

$$T^{\pi} : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}},\ T^{*} : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$$

$$\begin{aligned} T^{\pi}\mathcal{Q}(s,a) &= r(s,a) + \gamma \sum_{u\in\mathcal{S}} P(s,a,u)\,\mathcal{Q}(u,\pi(s)) \\ T^{*}\mathcal{Q}(s,a) &= r(s,a) + \gamma \sum_{u\in\mathcal{S}} P(s,a,u) \sup_{b\in\mathcal{A}} \mathcal{Q}(u,b) \end{aligned}$$

T^{π} is a linear operator, but T^{*} is not. Both, T^{π} and T^{*} are contractions w.r.t. the maximum norm.

Defining $\mathcal{Q}(s,\pi(s)) = \mathcal{Q}^{\pi}$ we have $T^{\pi}\mathcal{Q}^{\pi} = \mathcal{Q}^{\pi}$ and $\mathcal{Q}^{\pi}$ is the unique solution of this equation. Similarly, we have $T^{*}\mathcal{Q}^{*} = \mathcal{Q}^{*}$ and $\mathcal{Q}^{*}$ is the unique solution of this equation.

Fix an arbitrary initial policy π_0.

Policy evaluation: At iteration $t > 0$ compute the action-value function underlying π_t

Policy improvement: Given $\mathcal{Q}^{\pi_t}$ define π_{t+1} as the policy that is greedy w.r.t. $\mathcal{Q}^{\pi_t}$.

π_{t+1} gives rise to $\mathcal{Q}^{\pi_{t+1}}$: continue

Works similar but not exactly as value iteration (later)

Summary on Bellman equations

- *The* Bellman equation is the Bellman optimality equation. It characterises the optimal strategy based on the Bellman optimality principle
- It uses the transition probabilities
- Outlook: Use the actual process to estimate the transition probabilities or to directly sample the value function (or the state-action value function)
- Next: value prediction

Value prediction

- The Bellman (optimality) equation characterises an optimal value function
- In general this equation is not solvable
- Solution is possible by iterative schemes
- Need to take into account the embeddedness of the agent

Temporal Difference (TD) Learning for Value Prediction

Ideal value function

$$\begin{aligned} V_t &= \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \\ &= r_t + \gamma \left(r_{t+1} + \gamma r_{t+2} + \cdots \right) \\ &= r_t + \gamma \sum_{\tau=t+1}^{\infty} \gamma^{\tau-(t+1)} r_\tau \\ &= r_t + \gamma V_{t+1} \end{aligned}$$

Real value function is based on estimates of V_t and V_{t+1}, which may not obey this relation. Even if the estimates $\hat{V}_t$ and $\hat{V}_{t+1}$ are far from their true values we can at least require consistency, i.e. minimise the absolute value of the δ error (δ for $\delta\iota\alpha\varphi o\rho\acute{\alpha}$)

$$\delta_{t+1} = r_t + \gamma \hat{V}_{t+1} - \hat{V}_t$$

The simplest TD algorithm

Let $\hat{V}_t$ be the t-th iterate of a learning rule for estimating the value function V.

Let s_t the state of the system at time step t.

$$\delta_{t+1} = r_t + \gamma \hat{V}_t(s_{t+1}) - \hat{V}_t(s_t)$$

$$\hat{V}_{t+1} = \begin{cases} \hat{V}_t(s) + \eta\delta_{t+1} & \text{if } s = s_t \\ \hat{V}_t(s) & \text{otherwise} \end{cases}$$

$$\begin{aligned} \hat{V}_{t+1}(s_t) = \hat{V}_t(s_t) + \eta\delta_{t+1} &= \hat{V}_t(s_t) + \eta\left(r_t + \gamma \hat{V}_t(s_{t+1}) - \hat{V}_t(s_t)\right) \\ &= (1-\eta)\hat{V}_t(s_t) + \eta\left(r_t + \gamma \hat{V}_t(s_{t+1})\right) \end{aligned}$$

The update of the estimate $\hat{V}$ is an exponential average over the cumulative expected reward.

TD(0) Algorithm

Initialise η and γ and execute after each state transition

function TD0(s,r,$s1$,V) {

$\delta := r + \gamma * V[s1] - V[s];$

$V[s] := V[s] + \eta * \delta;$

return V;

}

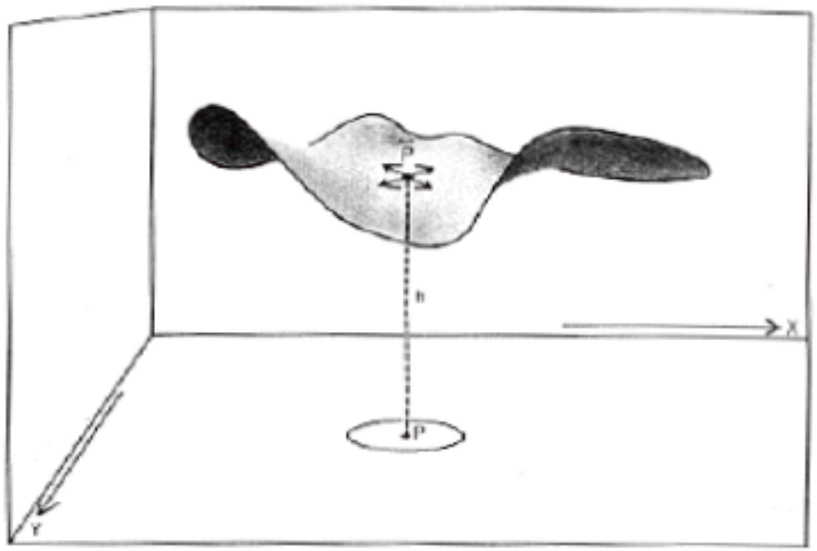
Remarks on the TD(0) Algorithm

- If the algorithm converges it must converge to a value function where the expected temporal differences are zero for all states. This state satisfies the Bellman optimality equation.
- The continuous version of the algorithm can be shown to be globally asymptotically stable
- TD(0) is a stochastic approximation algorithm. If the system is ergodic and the learning rate is appropriately decreased, it behaves like the continuous version.

Back to optimality in physics

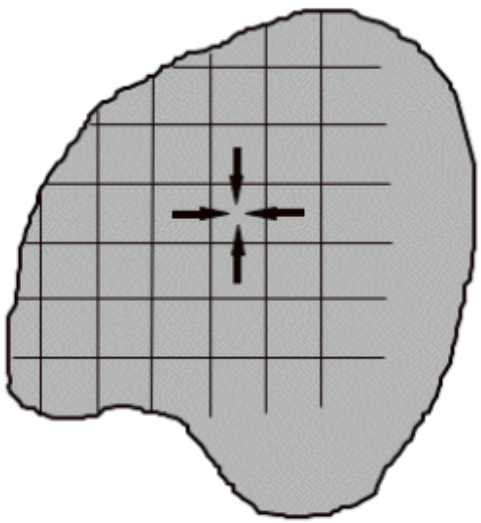
- Classical mathematics: Dirichlet problem (D. Hilbert, 1900)
  - Given function f that has values everywhere on the boundary of a region in $\mathbb{R}^n$
  - Is there a unique continuous function u twice continuously differentiable in the interior and continuous on the boundary, such that u is harmonic in the interior and $u = f$ on boundary?

- What is the shape of a soap bubble in a warped ring?



Two Approaches for Soap Bubble Example

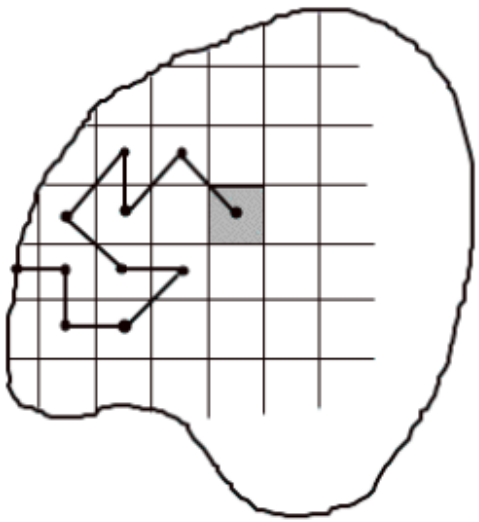
Relaxation (e.g., finite-difference)



$$\phi_{xx}(x_i, y_j) = [u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)]/(\Delta x)^2$$

$$\phi_{yy}(x_i, y_j) = [u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})]/(\Delta y)^2$$

Monte Carlo Method



Approximate from sample paths (from interior/boundary)

Any chance to find a good compromise between the two approaches?

Eligibility Traces for TD

- TD learning can often be accelerated by the addition of *eligibility traces*.
- TD≡TD(0) updates the value function only for the immediately preceding state
- But r_{t+1} provides useful information for learning earlier predictions as well
- Extend TD by eligibility traces:
  - Short-term memory of many previous state that are updated (though to a lesser extend the farther they are back)
  - Eligibility traces are usually implemented by an exponentially decaying memory trace, with decay parameter λ.
  - TD should thus be more specifically TD(λ) with $0 \leq \lambda \leq 1$
  - Eligibility extends less wide into the past than the time horizon towards the future that is controlled by the discount rate.
  - TD(1) updates all the preceding predictions equally (for $\gamma = 1$)

www.scholarpedia.org/article/Temporal_difference_learning

TD(λ) [i.e. with eligibility traces]

TD(0):

$$\hat{V}_{t+1}(s) = \begin{cases} \hat{V}_t(s) + \eta\delta_{t+1} & \text{if } s = s_t \\ \hat{V}_t(s) & \text{otherwise} \end{cases}$$

TD(λ):

$$\hat{V}_{t+1}(s) = \hat{V}_t(s) + \eta\delta_{t+1} e_{t+1}(s)$$

where e is a (replacing) eligibility trace with parameter[s] λ [and γ]

i.e. the value of all recently visited states is changed into the same direction.

λ is called *trace decay parameter*

1-D maze example is solved essentially when the goal was found once (it may take a bit more time to full convergence). Homework: Test this numerically!

Eligibility Traces

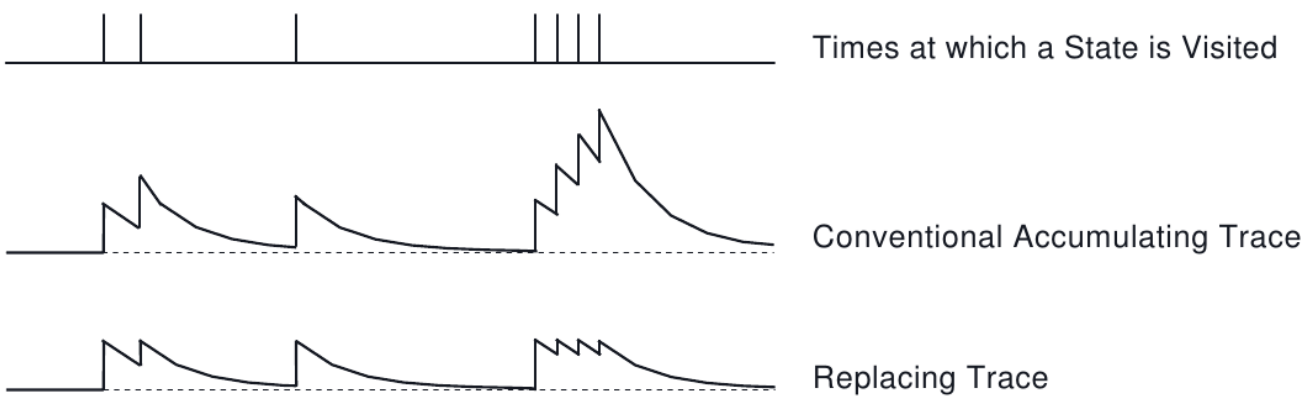
TD(λ): Unification of TD (update only value of previous state) and MC (update all states since start) using eligibility traces

Accumulating eligibility trace (e.g. in certain dynamical problems)

$$e_{t+1}(s) = \begin{cases} \gamma\lambda e_t(s) + 1 & \text{if } s = s_t \\ \gamma\lambda e_t(s) & \text{if } s \neq s_t \end{cases}$$

Replacing eligibility trace (e.g. in a maze)

$$e_{t+1}(s) = \begin{cases} 1 & \text{if } s = s_t \\ \gamma\lambda e_t(s) & \text{if } s \neq s_t \end{cases}$$

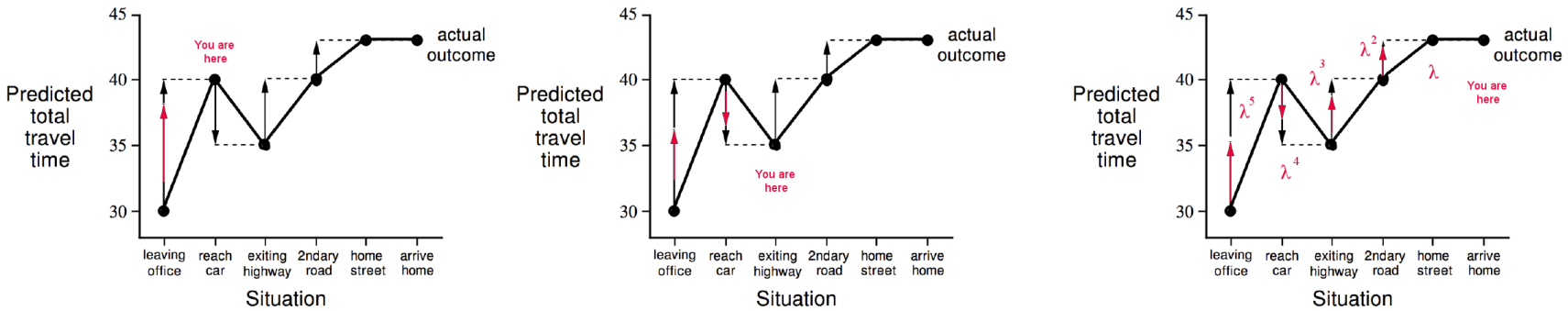


S.P. Singh & R.S. Sutton. Reinforcement learning with replacing eligibility traces. Rec. Adv. in RL 1996

Example: Driving home

State	Elapsed Time (minutes)		Predicted Time to Go	Predicted Total Time
leaving office	0		30	30
reach car, raining	5	(5)	35	40
exit highway	20	(15)	15	35
behind truck	30	(10)	10	40
home street	40	(10)	3	43
arrive home	43	(3)	0	43

Eligibility Traces



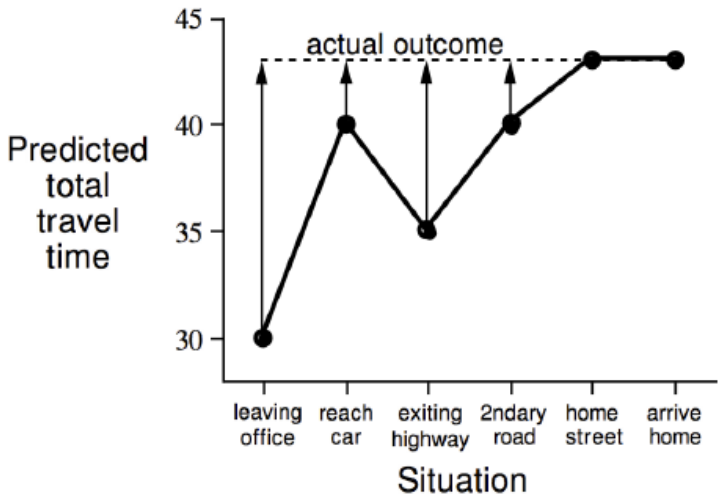
...

Changes when using TD with eligibility traces.

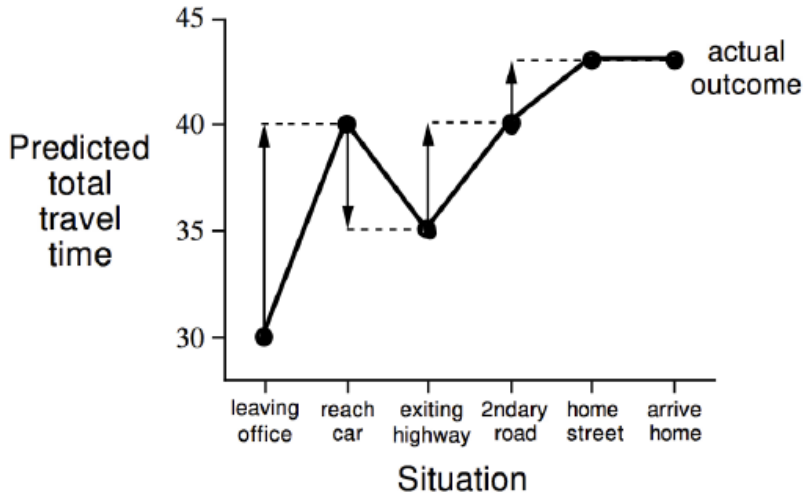
(This illustration is not to scale and does not reflect the accumulation of the changes)

Comparison: MC vs. TD

Expected time (= cost = neg. reward) to arrive home:



Changes recommended by MC (once at the end of episode)



Changes recommended by TD (updating one value per step)

Summary

- TD(λ) provides efficient estimates of the value function in Markovian reward processes
- It generalises Monte Carlo methods, but can be used as well in non-episodic tasks
- Appropriate tuning of λ (and γ) can lead to a significantly faster convergence than both Monte Carlo and TD(0).
- Next time: MC and DP and TD(λ)

Literature

Szepesvári, C. (2010). *Algorithms for reinforcement learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning **4**:1, 1-103. (Chapter 1)

Giegerich, R., Meyer, C., & Steffen, P. (2002). Towards A Discipline of Dynamic Programming. *GI Jahrestagung* **19**, 3-44. (make sure to read sections 1 and 2)