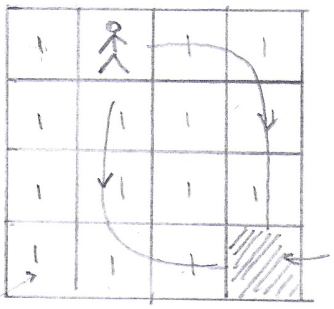
# Reinforcement learning (so far)

Example: shortest path problem to terminal state in grid world

state space $\mathcal{S}$

state sequence $S_t : 0 \leq t \leq T$

cost sequence $C_t : 0 \leq t < T$

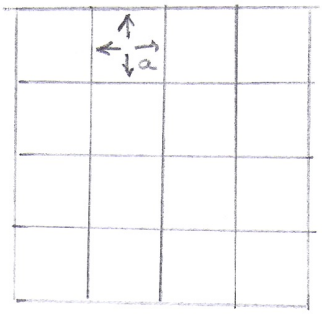


unit cost = shortest path

terminal state $S_T$ reached in $T$ for which $C_T = 0$

action space $\mathcal{A}$

action sequence $A_t : 0 \leq t < T$




* MDP dynamics is specified by a transition kernel: $P^a_{ss'} = \mathbb{P}[S_{t+1}=s' \mid S_t=s, A_t=a]$
* A policy is a decision strategy $\pi(a|s) = \mathbb{P}[A_t=a \mid S_t=s]$

## Cost formulation of the MDP (as in control theory)

The optimal value function (cost-to-go function) minimizes the expected cost (length): $v_*(s) = \min_\pi \mathbb{E}\left[\sum_{k=0}^{T} C_{t+k+1} \mid S_t=s\right]$

## Characterization via Bellman optimality

$$v_*(s) = \min_a \left\{ \mathbb{E}[C_{t+1} \mid S_t=s, A_t=a] + \mathbb{E}\left[\mathbb{E}\left[\sum_{k=0}^{T} C_{t+k+2} \mid S_{t+1}\right] \mid S_t=s, A_t=a\right] \right\}$$

$$= \min_a \left\{ c(s,a) + \mathbb{E}\left[v_*(S_{t+1}) \mid S_t=s, A_t=a\right] \right\}$$

$$\mathbb{E}\left[v_*(S_{t+1}) \mid S_t=s, A_t=a\right] = \sum_{s'} P^a_{s's}\, v_k(s')$$

## Reinforcement learning algorithms

↳ Dynamic programming: (requires full knowledge)

$$\cdots \pi_k \xrightarrow{E} v_k \xrightarrow{I} \pi_{k+1} \xrightarrow{E} v_{k+1} \cdots \longrightarrow v_*$$

E: Bellman expectation; I: greedy move

↳ Monte-Carlo methods: (no bootstrapping)

$$\cdots \pi_k \xrightarrow{E} v_k \xrightarrow{I} \pi_{k+1} \xrightarrow{E} v_{k+1} \cdots \longrightarrow v_*$$

E: sample mean; I: $\varepsilon$-greedy move

↳ Temporal difference:

(Q-learning on action-value function)

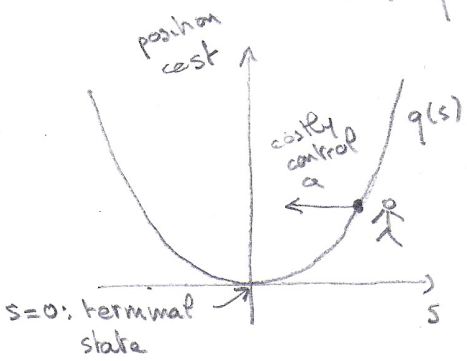
cheapest move from successor

$$q_{t+1}(S_t, A_t) \leftarrow (1-\alpha_t)\, q_t(S_t, A_t) + \alpha_t \left[C_{t+1} + \min_a q_t(S_{t+1}, a)\right]$$

⇒ Computational cost (and learning time) remains fundamentally limited by nonlinearities (presence of the "min")

# Analytical solution from control theory

**Example:** An agent is moving in a continuous state space incurring a cost $q(s)$ at each position. The agent want to minimize its long term cost by exerting control on its dynamics (reaching zero cost state as soon as possible). However exerting control also come at a cost (the faster, the more expensive).




Optimal control theory resolves the above trade off.

**Controlled dynamics:** $\dot{x} = a$ ← control is a time-dependent function

**Infinitesimal cost:** $c(s,a) = q(s) + \frac{a^2}{2} = \frac{s^2+a^2}{2}$ ← specifies tradeoff

**Cost-to-go function:** $v_*(s) = \min_a \left[ \int_0^T c(s_t, a_t)\, dt \right]$, $T$: time to reach zero.

**Bellman optimality:** $v_*(s) = \min_a \left[ c(s,a)\, dt + v_*(s + a\, dt) \right]$

$$\cancel{v_*(s)} = \min_a \left[ \frac{s^2+a^2}{2} dt + \cancel{v_*(s)} + v_*'(s)\, a\, dt \right]$$

$$-\frac{s^2}{2} = \min_a \left[ \frac{a^2}{2} + v_*'(s)\, a \right]$$

⚠ Minimization can be carried out explicitly (quadratic function of $a$)

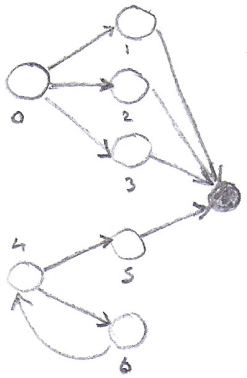
$$a_* = -v_*'(s) \Rightarrow -\frac{s^2}{2} = -\frac{v_*'(s)^2}{2} \Rightarrow \boxed{v_*(s) = \frac{s^2}{2},\ s = s_0 e^{-t}}$$

**Key ingredients allowing resolution:** * special functional form for cost
* minimization in a continuous setting

Expecting full analytical resolution is unreasonable in general. However one can hope to simplify the problem if the two above ingredients can be adapted to MDP settings.

# Continuous framework for MDPs

Example: shortest path on a graph with terminal state.

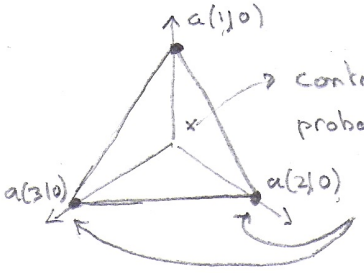



1] First idea: space of control = space of transition probabilities

$$a(s'|s) = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a] = P^a_{ss'}$$

↓ control can generate any transition probability as long as the transitions are allowed by the graph

For $s = 0$, 3 possible transitions to 1, 2, 3.




controls belongs to the simplex of probability distributions (continuous space)

vertices corresponds to deterministic policies

2] Second idea: passive dynamics = reference dynamics to quantify the cost of controls

Passive dynamics is naturally defined as a random walk on the graph via a transition kernel $p(s'|s) = P^{\emptyset}_{ss'}$ ← no control

3] Third idea: control decision cost = distance from the passive dynamics

$$c(s,a) = q(s) + D_{KL}\big(a(\cdot|s) \,\|\, p(\cdot|s)\big) = q(s) + \sum_{s'} a(s'|s) \log \frac{a(s'|s)}{p(s'|s)}$$

$q(s)$: site specific cost

$D_{KL}$: control/action cost = Kullback Leibler divergence

⇒ Bellman equation:

$$v_*(s) = \min_a \left\{ c(s,a) + \sum_{s'} P^a_{ss'} v_*(s') \right\}$$

$$v_*(s) = q(s) + \min_a \left\{ \sum_{s'} a(s'|s) \left[ \log \frac{a(s'|s)}{p(s'|s)} + v_*(s') \right] \right\}$$

# Reduction to linear Bellman equation

**Key insight**: performing a nonlinear change of variable by introducing the "desirability" function: $z_*(s) = e^{-v_*(s)}$.

**Goal**: using the well-known KL divergence property to perform the minimization explicitly in Bellman equation

**Bellman equation**: $-\log z_*(s) = \min_a \left\{ q(s) + \underbrace{\sum_{s'} a(s'|s) \log\left(\frac{a(s'|s)}{p(s'|s) z_*(s')}\right)}_{\text{almost a KL divergence: normalization!}} \right\}$

To normalize we introduce: $Z(s) = \sum_{s'} p(s'|s) z_*(s')$. Then:

$$-\log z_*(s) = \min_a \left\{ q(s) + \underbrace{\sum_{s'} a(s'|s) \log\left(\frac{a(s'|s) Z(s)}{p(s'|s) z_*(s')}\right)}_{D_{KL}\left(a(\cdot|s) \,\middle\|\, \frac{p(\cdot|s) z_*(\cdot)}{Z(s)}\right)} - \underbrace{\sum_{s'} a(s'|s) \log Z(s)}_{\to 1} \right\}$$

**Minimization**: $D_{KL} = 0 \iff a(\cdot|s) = \boxed{a^*(\cdot|s) = \frac{p(\cdot|s) z_*(\cdot)}{Z(s)}}$

($D_{KL}$: KL divergence)

optimal control

**Linear Bellman equation**:

$$-\log z_* = q - \log Z \iff z_* = e^{-q} Z \iff \boxed{\begin{aligned} z_*(s) &= e^{-q(s)} \sum_{s'} p(s'|s) z_*(s') \\ z_*(s) &= e^{-q(s)} \mathbb{E}\left[ z_*(S_{t+1}) \,\middle|\, S_t = s \right] \end{aligned}}$$

linear equation about $z_*$ !

**Concretely** under matrix form

$$\begin{bmatrix} z_T \\ z_N \end{bmatrix} = \left[\begin{array}{c|c} Id & 0 \\ \hline 0 & e^{-q_N} \end{array}\right] \left[\begin{array}{c|c} Id & 0 \\ \hline M_{NT} & M_{NN} \end{array}\right] \begin{bmatrix} z_T \\ z_N \end{bmatrix}$$

$z_T$ ← terminal nodes (zero cost) $q_T = 0$, $z_T = 1$

$z_N$ ← non terminal nodes

$$\left( \begin{bmatrix} e^{q_N} \end{bmatrix} - M_{NN} \right) z_N = M_{NT}$$

↑ diagonal matrix

← directly solvable equation as opposed to eigenvalue equation (solvable by iteration)

## Embedding of classical MDP

Example: shortest path on a graph with terminal state.

In MDP, this corresponds to deliver a unit cost at each site except for the terminal node. Actions are free and deterministic!

By contrast, actions have inherent costs in linear Bellman setting.

Idea: solving MDP via approximation. Specifically let assign $q(s) = p$ for all non terminal nodes. Taking $p \to +\infty$ makes the action cost small (negligible) compared to the residence cost $p$, while the linear problem remains solvable:

$$\text{shortest path} = \lim_{p \to +\infty} \frac{v_s^{(p)}(s)}{p} \leftarrow \text{value function obtain for cost } p.$$

How general is this approximation method? Very!

| Classical MDP specified by: | Linear Bellman framework specified by: |
|---|---|
| 1) $P^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$ | 1) $q(s) \leftarrow$ site specific cost |
| 2) $c(a,s) = \mathbb{E}[C_{t+1} \mid S_t = s, A_t = a]$ | 2) $p(s'|s) = P^{\emptyset}_{ss'} \leftarrow$ passive dynamics |

A MDP can be represented (embedded) in the linear framework if there $q(s)$, $p(s'|s)$ yielding the MDP costs $c(a,s)$ for the control $a(s'|s) = P^a_{ss'}$, i.e.:

$$\underbrace{q(s)}_{\text{unknown}} + \sum_{s'} P^a_{ss'} \log \frac{P^a_{ss'}}{\underbrace{p(s'|s)}_{\text{unknown}}} = c(a,s) \to |\mathcal{A}(s)| \text{ equations at fixed } s \text{ about } q(s) \text{ and } l_{s'} = \log p(s'|s)$$

$$q(s) - \sum_{s'} P^a_{ss'} \log p(s'|s) = c(a,s) - \sum_{s'} P^a_{ss'} \log P^a_{ss'} = b(a)$$

In matrix form: $q\underline{1} - P\underline{l} = \underline{b}$ where $P$ is $|\mathcal{A}(s)| \times |\mathcal{S}|$ matrix with $\sum_{s'} e^{l_{s'}} = \sum_{s'} p(s'|s) = 1 \leftarrow$ normalization

If $P$ is generally row-rank deficient ($|\mathcal{S}| \gg |\mathcal{A}(s)|$), thus there is $\underline{c}$ such that $P\underline{c} = \underline{b}$. Choosing $q = -\log \sum_{s'} \exp(-c_{s'})$ guarantees that $\underline{l} = q\underline{1} - \underline{c}$ solve the system of equations.

## Applications

### 1) Dynamic programming:

A) Given a MDP, build a linear embedding

B) Then optimize the policy by solving linear Bellman equation

C) Find an approximate optimal strategy for the original MDP by picking the most likely action (greedy move)

### 2) Monte Carlo method:

The desirability function $z_*$ can be learnt from sampled experiences. The estimator is build recursively from the linear Bellman equation

$$z_*(s) = e^{-q(s)} \mathbb{E}\left[z_*(s_1) \mid s_0 = s\right] = e^{-q(s)} \mathbb{E}\left[e^{-q(s_1)} \mathbb{E}\left[z_*(s_2)\right] \mid s_0 = s\right] \cdots$$

$$\hat{z}(s) = \mathbb{E}\left[e^{-\sum_{t=0}^{T} q(s_t)} \mid s_0 = s\right]$$ where $s_t$ is sampled according to the passive dynamics

### 3) Temporal difference method:

A naive $\alpha$-constant MC scheme ($\alpha$: learning rate) is

$$\hat{z}(s_t) \leftarrow \hat{z}(s_t) + \alpha\left[z^*(s_t) - \hat{z}(s_t)\right] \quad \leftarrow \text{no bootstraping}$$

↑ target ↑ averaging estimate

The linear Bellman equation suggests a target with bootstraping

$$z_*(s) = \mathbb{E}\left[e^{-q(s_t)} z_*(s_{t+1}) \mid s_t = s\right]$$

↑ expectation = MC averaging; ↑ evaluation at successor state = bootstraping

$$\hat{z}(s_t) \leftarrow \hat{z}(s_t) + \alpha_t\left(e^{-q(s_t)} \hat{z}(s_{t+1}) - \hat{z}(s_t)\right)$$ where the learning rates satisfy usual conditions: $\sum_{t \geq 0} \alpha_t = +\infty$

$$\sum_{t \geq 0} \alpha_t^2 < +\infty$$