Unrolling Dynamic Programming via Graph Filters

S. Rozada, S. Rey, M. Alcocer, G. Mateos, and A. G. Marques

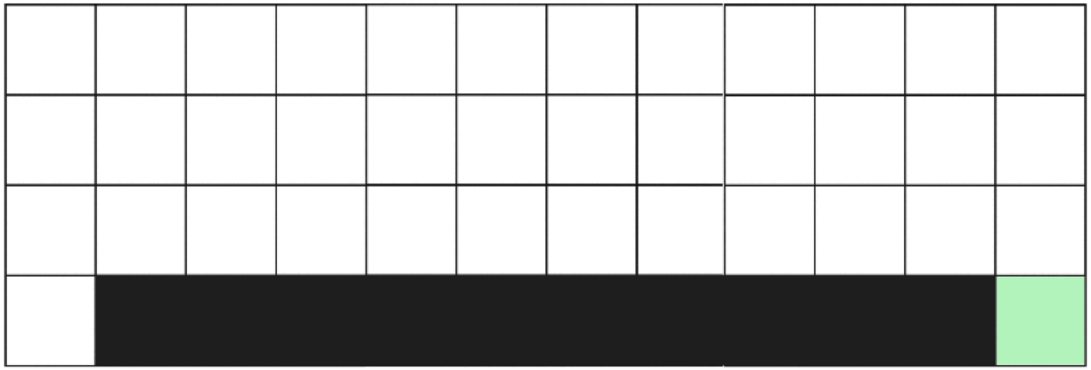
King Juan Carlos University, University of Rochester

GSP 2026 - Madrid - June 8 - 10, 2026

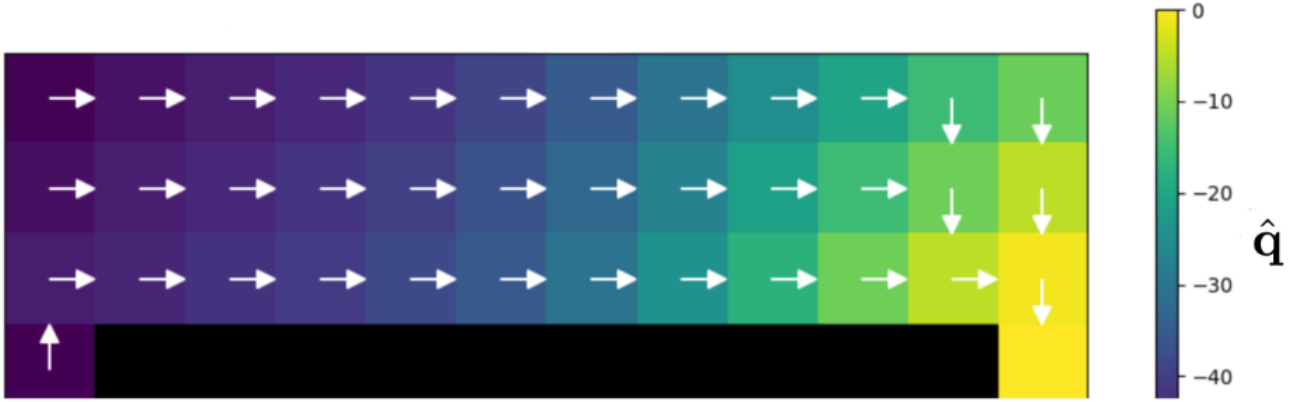
Dynamic programming (DP)

Universidad Rey Juan Carlos

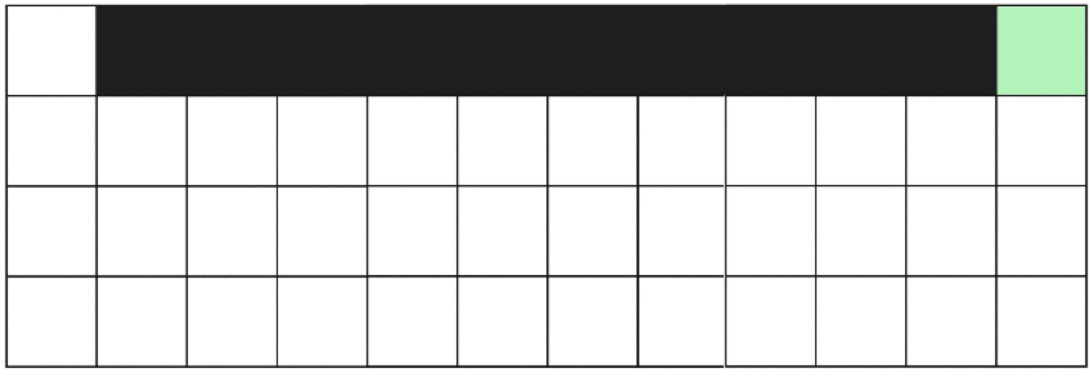
- We want to solve the cliff problem via dynamic programming (**DP**)



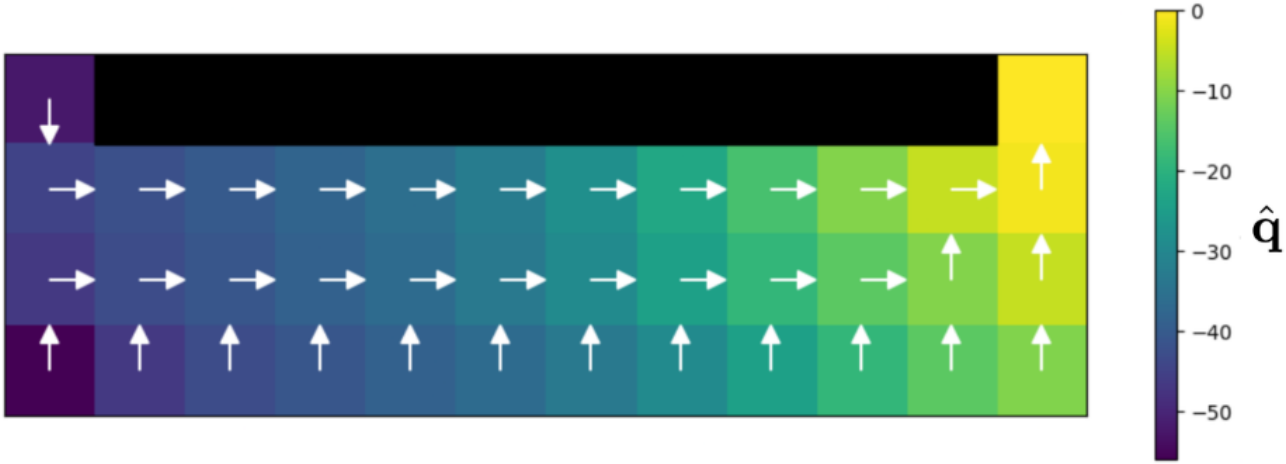
- Find value function (**VF**) via Bellman's equation and obtain policy



- If the problem changes in structured way



- We have to solve again ⇒ Solution does not transfer



Can we transfer DP solutions?

Outline

1) Preliminaries: Policy iteration

2) DP meets GSP: Unrolling policy iteration

3) BellNet: A GNN to solve DP

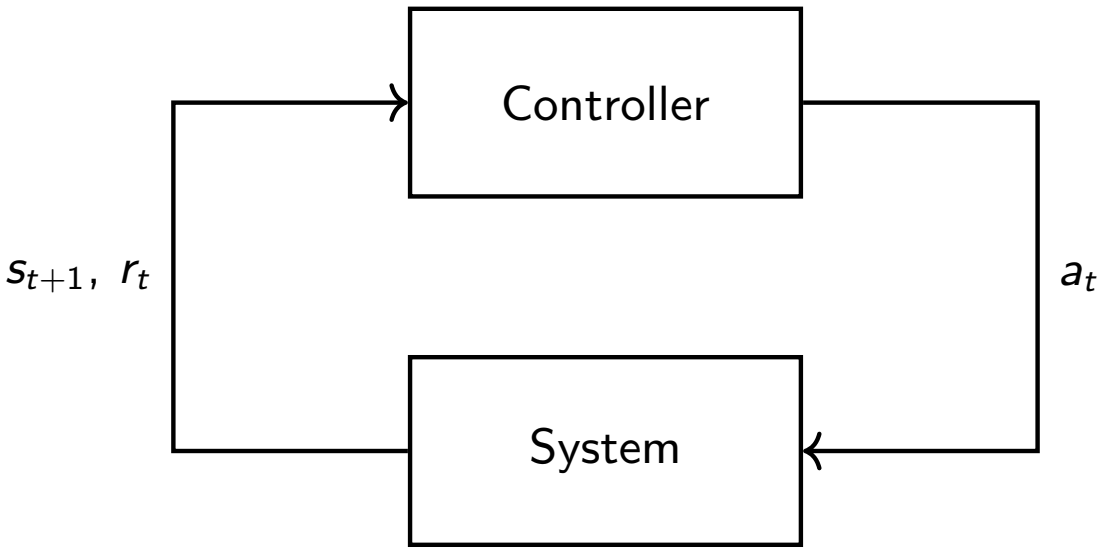
4) Numerical results

Outline

The Markov Decision Process (MDP)

Universidad Rey Juan Carlos

- DP formalized via the Markov decision process (MDP)
  - $\Rightarrow$ Discrete state space $\mathcal{S}$
  - $\Rightarrow$ Discrete action space $\mathcal{A}$
  - $\Rightarrow$ Transition probability function $p(\cdot|s,a)$
  - $\Rightarrow$ Reward function $r(s,a)$

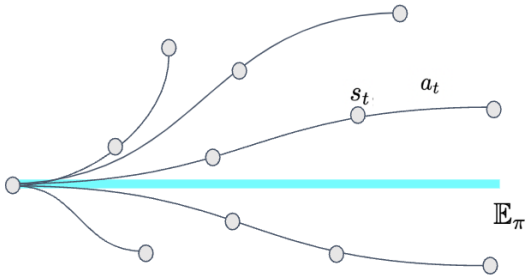


[Bertsekas, Athena Scientific]

The goal of DP

- Learn optimal policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ solving

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

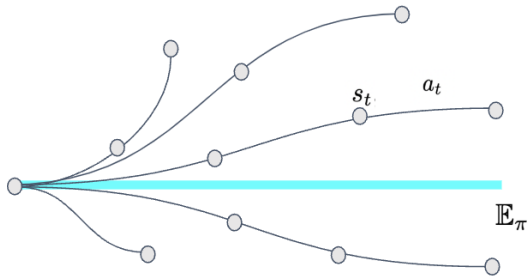


$\Rightarrow$ Challenging functional optimization problem

The goal of DP

- Learn optimal policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ solving

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$



$\Rightarrow$ Challenging functional optimization problem

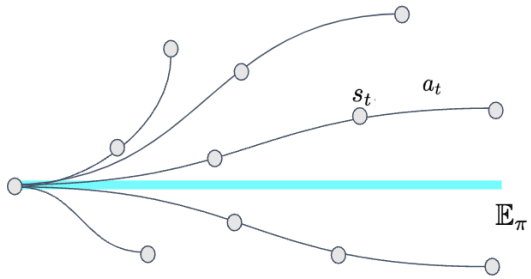
- VFs key to DP formalization $\Rightarrow$ Characterize optimality conditions

$$Q^{\pi}(s, a) := \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right]$$

The goal of DP

- Learn optimal policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ solving

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$



$\Rightarrow$ Challenging functional optimization problem

- VFs key to DP formalization $\Rightarrow$ Characterize optimality conditions

$$Q^{\pi}(s, a) := \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right]$$

$$Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{\pi} \left[Q^{\pi}(s', a') \right]$$

Value functions (VFs)

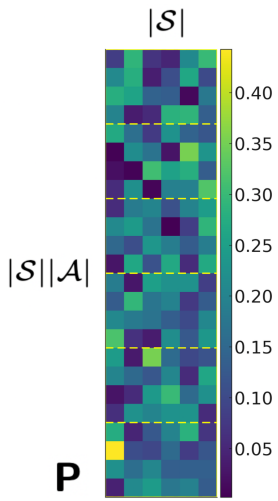
Universidad Rey Juan Carlos

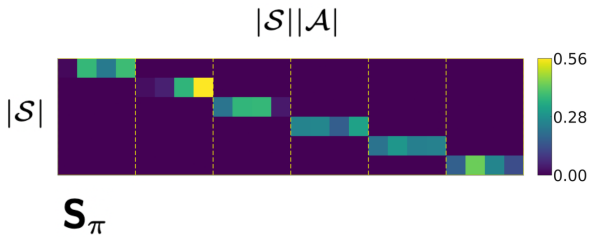
- Bellman's equations $\Rightarrow$ Linear system for fixed policy π

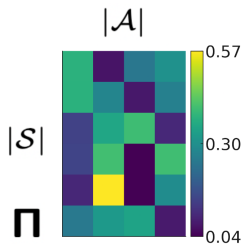
$$\mathbf{q}_\pi = \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}_\pi \quad \text{with} \quad \mathbf{P}_\pi = \mathbf{P}\mathbf{S}_\pi$$

$\Rightarrow \mathbf{P} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ is transition operator

$\Rightarrow \mathbf{S}_\pi = (\mathbf{I} \odot \mathbf{\Pi}^\top)^\top \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|}$ is VF mapping $\Rightarrow \mathbf{v}_\pi = \mathbf{S}_\pi \mathbf{q}_\pi$





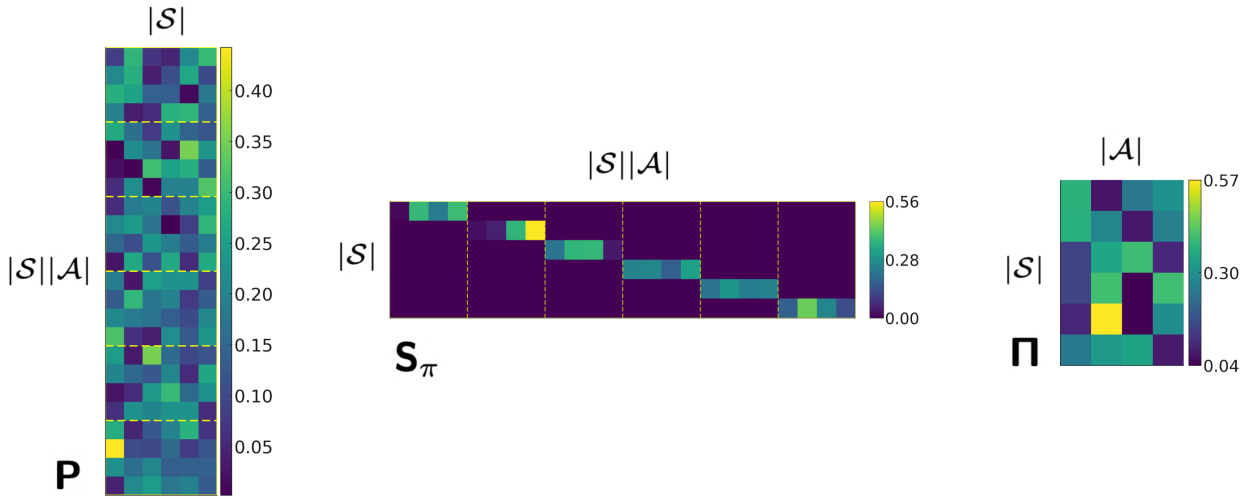


- Bellman's equations $\Rightarrow$ Linear system for fixed policy π

$$\mathbf{q}_\pi = \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}_\pi \quad \text{with} \quad \mathbf{P}_\pi = \mathbf{P}\mathbf{S}_\pi$$

$\Rightarrow \mathbf{P} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ is transition operator

$\Rightarrow \mathbf{S}_\pi = (\mathbf{I} \odot \mathbf{\Pi}^\top)^\top \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|}$ is VF mapping $\Rightarrow \mathbf{v}_\pi = \mathbf{S}_\pi \mathbf{q}_\pi$



- Leads to better policy $\Rightarrow \pi'(s) = \max_{a \in \mathcal{A}} Q^\pi(s, a)$

Policy iteration

- **Policy iteration** $\Rightarrow$ Classic algorithm to solve DP
  - $\Rightarrow$ Guaranteed to converge to unique solution
  - $\Rightarrow$ Policy evaluation (**PE**) $\Rightarrow$ Solve Bellman's equation
  - $\Rightarrow$ Policy improvement (**PI**) $\Rightarrow$ Improve policy greedily
  - $\Rightarrow$ Nested (double) loop:

$$\textbf{(PE)} \qquad \mathbf{q}^{(n+1)} = \mathbf{r} + \gamma \mathbf{P}_{\pi^{(m)}} \mathbf{q}^{(n)} \quad \text{for } n = 1, \ldots, N$$

$$\textbf{(PI)} \qquad \pi^{(m+1)}(s) = \max_{a \in \mathcal{A}} Q^{(N)}(s, a) \quad \text{for } m = 1, \ldots, M$$

Policy iteration

- **Policy iteration** $\Rightarrow$ Classic algorithm to solve DP
  - $\Rightarrow$ Guaranteed to converge to unique solution
  - $\Rightarrow$ Policy evaluation (**PE**) $\Rightarrow$ Solve Bellman's equation
  - $\Rightarrow$ Policy improvement (**PI**) $\Rightarrow$ Improve policy greedily
  - $\Rightarrow$ Nested (double) loop:

$$\textbf{(PE)} \qquad \mathbf{q}^{(n+1)} = \mathbf{r} + \gamma \mathbf{P}_{\pi^{(m)}} \mathbf{q}^{(n)} \quad \text{for } n = 1, \ldots, N$$

$$\textbf{(PI)} \qquad \pi^{(m+1)}(s) = \max_{a \in \mathcal{A}} Q^{(N)}(s, a) \quad \text{for } m = 1, \ldots, M$$

- New DP problem requires running policy iteration again

Outline

Unrolling policy evaluation

- We unroll PE iteration

$$
\begin{aligned}
\mathbf{q}^{(k)} &= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}^{(k-1)} \\
&= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 \left(\mathbf{P}_\pi\right)^2 \mathbf{q}^{(k-2)} \\
&= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 \left(\mathbf{P}_\pi\right)^2 \mathbf{r}^2 + \gamma^3 \left(\mathbf{P}_\pi\right)^3 \mathbf{q}^{(k-3)} \\
&= \dots \\
&= \textstyle\sum_{j=0}^{k-1} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r} + \gamma^k \left(\mathbf{P}_\pi\right)^k \mathbf{q}^{(0)}
\end{aligned}
$$

Unrolling policy evaluation

- We unroll PE iteration

$$\begin{aligned}
\mathbf{q}^{(k)} &= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}^{(k-1)} \\
&= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 \left(\mathbf{P}_\pi\right)^2 \mathbf{q}^{(k-2)} \\
&= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 \left(\mathbf{P}_\pi\right)^2 \mathbf{r}^2 + \gamma^3 \left(\mathbf{P}_\pi\right)^3 \mathbf{q}^{(k-3)} \\
&= \dots \\
&= \textstyle\sum_{j=0}^{k-1} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r} + \gamma^k \left(\mathbf{P}_\pi\right)^k \mathbf{q}^{(0)}
\end{aligned}$$

- In the limit $\Rightarrow$ PE converges to true $\mathbf{q}_\pi$ and $\gamma^k \to 0$

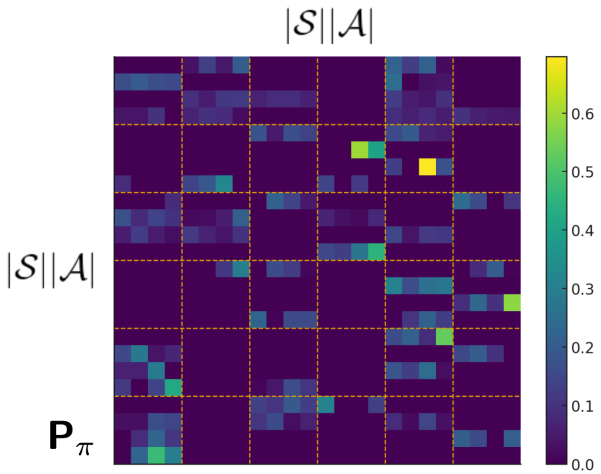
$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r}$$

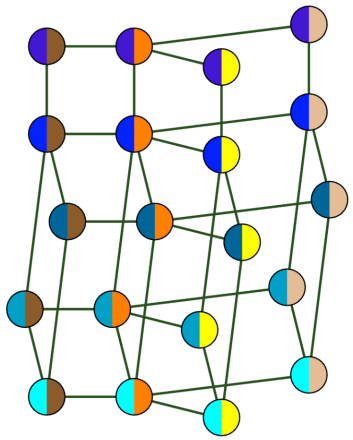
Policy evaluation is a graph filter

Universidad Rey Juan Carlos

- VFs are graph-filtered rewards with known coefficients

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r}$$

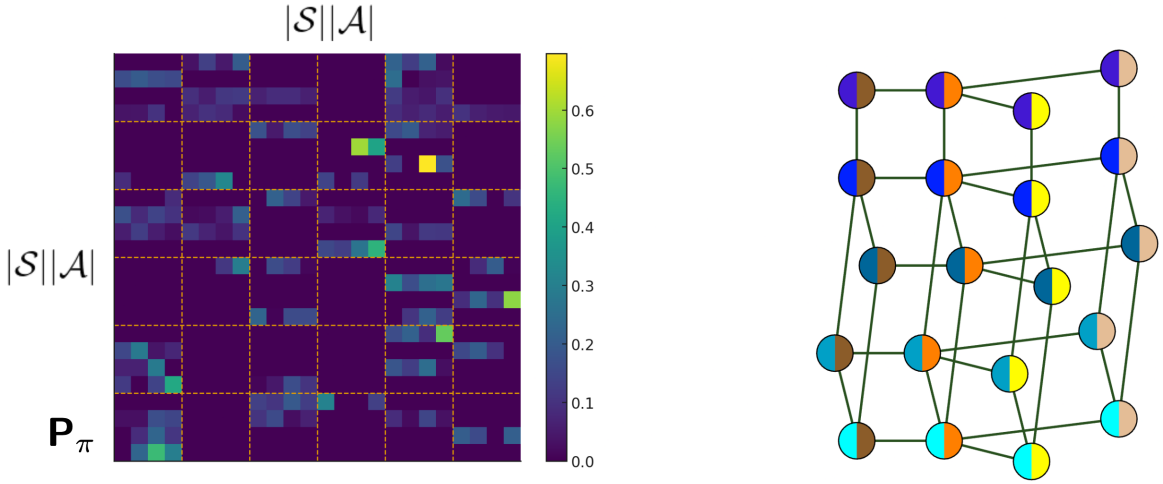




Policy evaluation is a graph filter

- VFs are graph-filtered rewards with known coefficients

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r}$$



- PI improves over actions $\Rightarrow \pi'(s) = \max_{a \in \mathcal{A}} Q^\pi(s,a)$

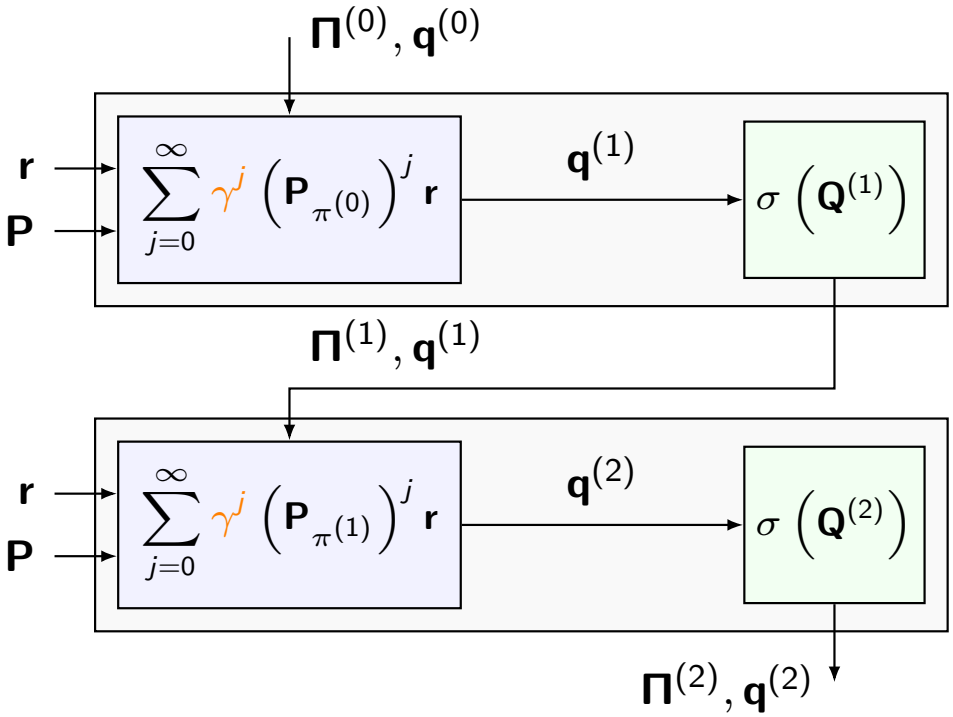
$$\mathbf{\Pi}' = \sigma(\mathbf{Q}^\pi) \quad \Leftrightarrow \quad \mathbf{\Pi}'_{ij} = \begin{cases} 1 & \text{if} \quad j = \operatorname{argmax}_k Q^\pi_{ik} \\ 0 & \text{otherwise} \end{cases}$$

Policy iteration is cascade of graph filters

- PE + PI yields linear–nonlinear operator

$$\textbf{(PE)} \qquad \mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j \left(\mathbf{P}_\pi\right)^j \mathbf{r}$$

$$\textbf{(PI)} \qquad \mathbf{\Pi}' = \sigma(\mathbf{Q}^\pi)$$



Outline

Compact representation

- **Obs.1:** By Cayley-Hamilton there exists finite-order filter

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} = \sum_{j=0}^{K} h_j (\mathbf{P}_\pi)^j \mathbf{r}$$

Compact representation

- **Obs.1:** By Cayley-Hamilton there exists finite-order filter

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} = \sum_{j=0}^{K} h_j (\mathbf{P}_\pi)^j \mathbf{r}$$

- **Obs.2:** In practice, policy iteration can be truncated

$$\hat{\mathbf{q}}_\pi = \sum_{j=0}^{K} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} + \gamma^{K+1} (\mathbf{P}_\pi)^{K+1} \mathbf{q}^{(0)}$$

- **Obs.1:** By Cayley-Hamilton there exists finite-order filter

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} = \sum_{j=0}^{K} h_j (\mathbf{P}_\pi)^j \mathbf{r}$$

- **Obs.2:** In practice, policy iteration can be truncated

$$\hat{\mathbf{q}}_\pi = \sum_{j=0}^{K} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} + \gamma^{K+1} (\mathbf{P}_\pi)^{K+1} \mathbf{q}^{(0)}$$

- **Obs.3:** Max. non-linearity can be relaxed for tractability

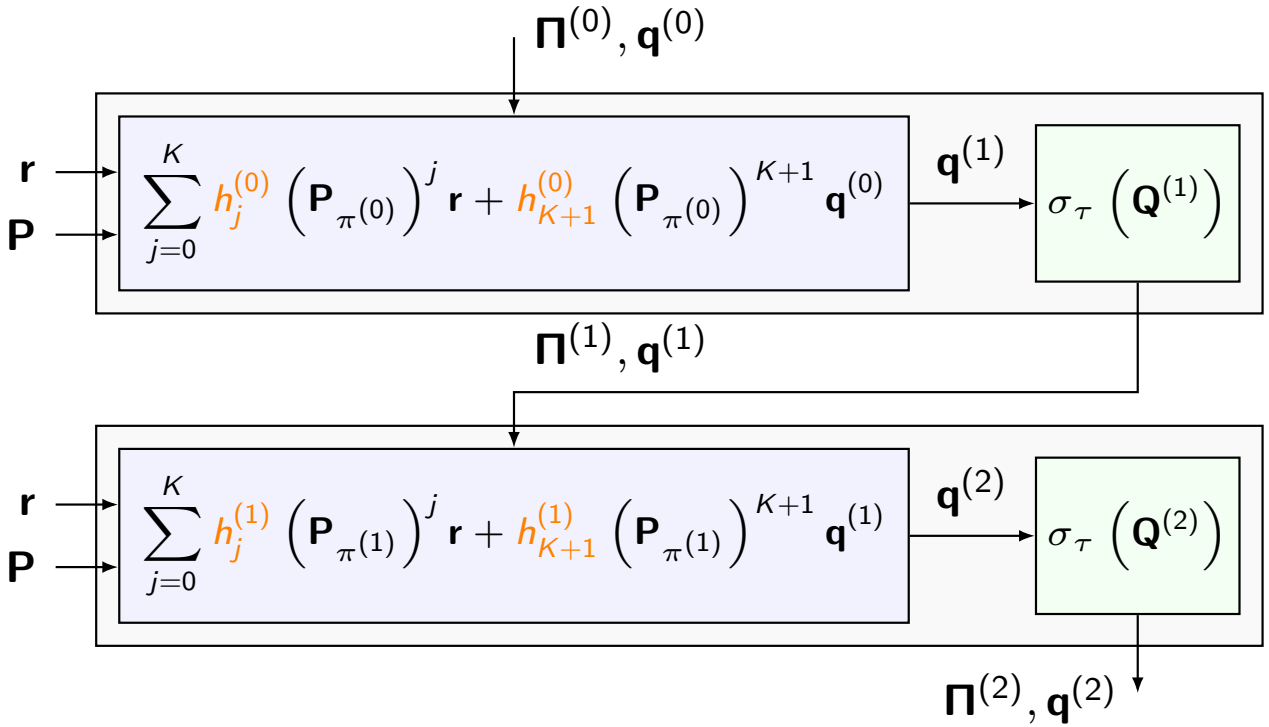
$$\mathbf{\Pi}' = \sigma_\tau(\mathbf{Q}^\pi) \quad \Leftrightarrow \quad \Pi'_{ij} = \frac{e^{Q^\pi_{ij}/\tau}}{\sum_{k=1}^{|\mathcal{A}|} e^{Q^\pi_{ik}/\tau}}$$

- BellNet $\Rightarrow \boldsymbol{\Phi}(\cdot; \mathcal{H})$ with parameters $\mathcal{H} = \{(h_1^{(l)}, \ldots, h_{K+1}^{(l)})\}_{l=0}^{L}$

  $\Rightarrow$ Implements PE and PI L times $\Rightarrow \{\hat{\mathbf{q}}, \hat{\boldsymbol{\Pi}}\} := \boldsymbol{\Phi}(\bar{\mathbf{q}}; \mathcal{H})$

$$\textbf{(PE)} \qquad \mathbf{q}^{(l+1)} = \sum_{j=0}^{K} h_j^{(l)} \left(\mathbf{P}_{\pi^{(l)}}\right)^j \mathbf{r} + h_{K+1}^{(l)} \left(\mathbf{P}_{\pi^{(l)}}\right)^{K+1} \mathbf{q}^{(l)}$$

$$\textbf{(PI)} \qquad \boldsymbol{\Pi}^{(l+1)} = \sigma_\tau(\mathbf{Q}^{(l+1)})$$



Training via gradient descent

- How to train our BellNet arquitecture?
- Optimality condition of VFs in DP

$$\mathbf{q}^\star = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^\star \quad \text{with} \quad \mathbf{v}^\star(s) = \max_{a \in \mathcal{A}} \mathbf{q}^\star(s,a)$$

Training via gradient descent

- How to train our BellNet arquitecture?
- Optimality condition of VFs in DP

$$\mathbf{q}^\star = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^\star \quad \text{with} \quad \mathbf{v}^\star(s) = \max_{a \in \mathcal{A}} \mathbf{q}^\star(s,a)$$

- Consider the quadratic optimization problem

$$\begin{aligned} \min_{\mathcal{H}} \quad & \ell(\mathcal{H}) = \|\mathbf{r} + \gamma \mathbf{P}\hat{\mathbf{v}} - \hat{\mathbf{q}})\|_2^2 \\ \text{s.t.} \quad & \{\hat{\mathbf{q}}, \hat{\mathbf{\Pi}}\} = \mathbf{\Phi}(\bar{\mathbf{q}}; \mathcal{H}) \quad \hat{\mathbf{v}} = \mathbf{S}_{\hat{\pi}} \hat{\mathbf{q}} \end{aligned}$$

Training via gradient descent

- How to train our BellNet arquitecture?
- Optimality condition of VFs in DP

$$\mathbf{q}^\star = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^\star \quad \text{with} \quad \mathbf{v}^\star(s) = \max_{a \in \mathcal{A}} \mathbf{q}^\star(s,a)$$

- Consider the quadratic optimization problem

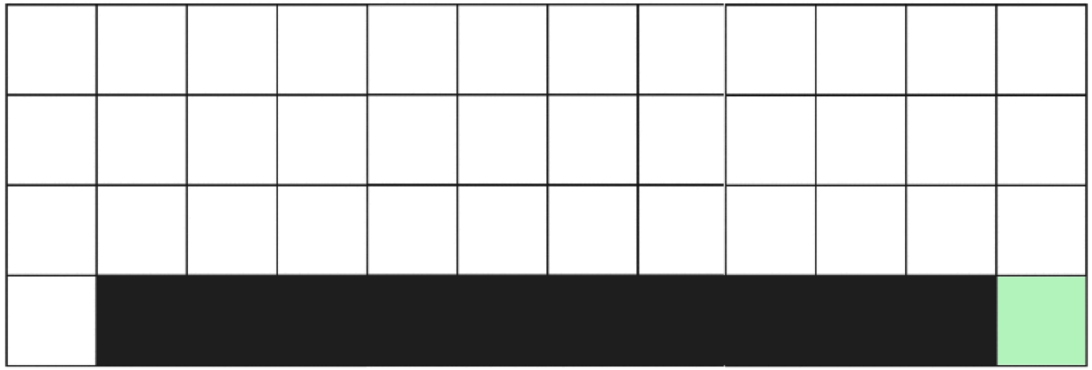
$$\begin{aligned} \min_{\mathcal{H}} \quad & \ell(\mathcal{H}) = \|\mathbf{r} + \gamma \mathbf{P}\hat{\mathbf{v}} - \hat{\mathbf{q}})\|_2^2 \\ \text{s.t.} \quad & \{\hat{\mathbf{q}}, \hat{\mathbf{\Pi}}\} = \mathbf{\Phi}(\bar{\mathbf{q}}; \mathcal{H}) \quad \hat{\mathbf{v}} = \mathbf{S}_{\hat{\pi}} \hat{\mathbf{q}} \end{aligned}$$

- Solve via gradient descent $\Rightarrow \nabla_{\mathcal{H}} \ell(\mathcal{H})$

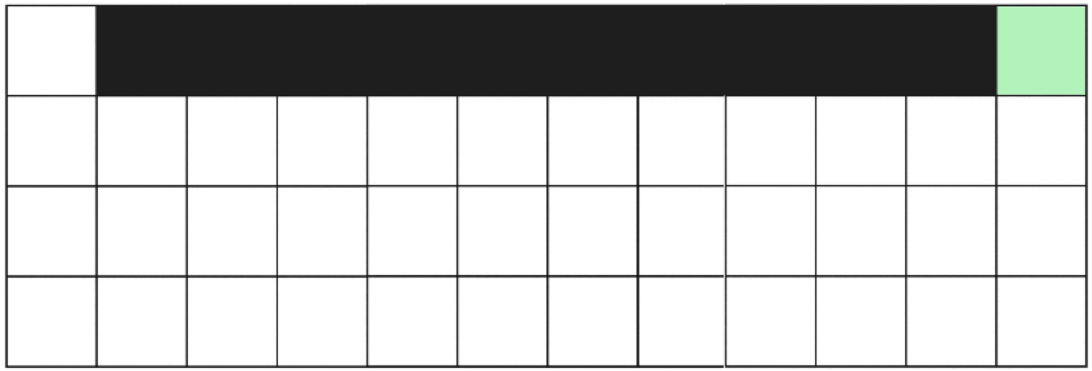
Outline

Cliff-walking problem

- We train BellNet to solve the cliff-walking problem



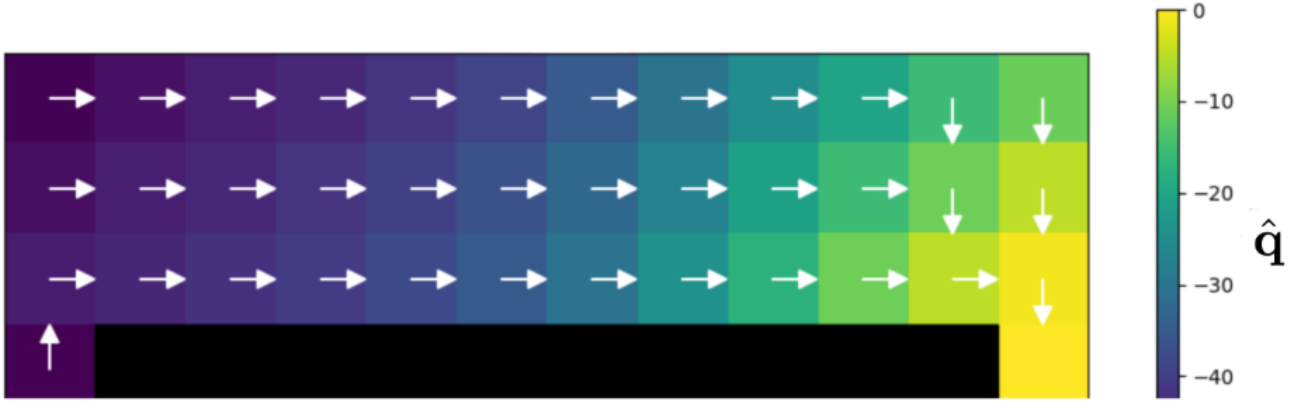
- We test BellNet by solving the mirrored cliff-walking problem



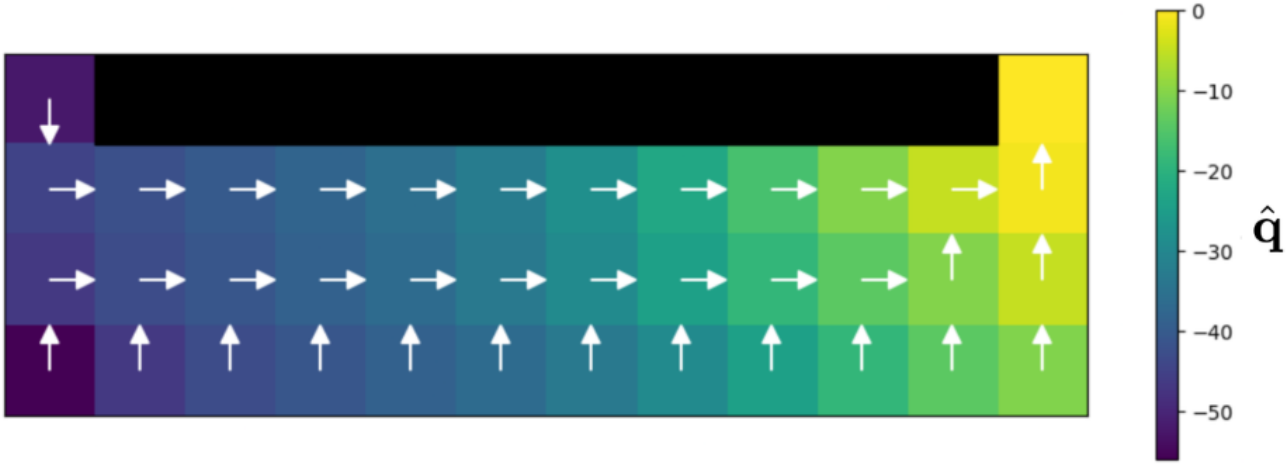
Cliff-walking problem

Universidad Rey Juan Carlos

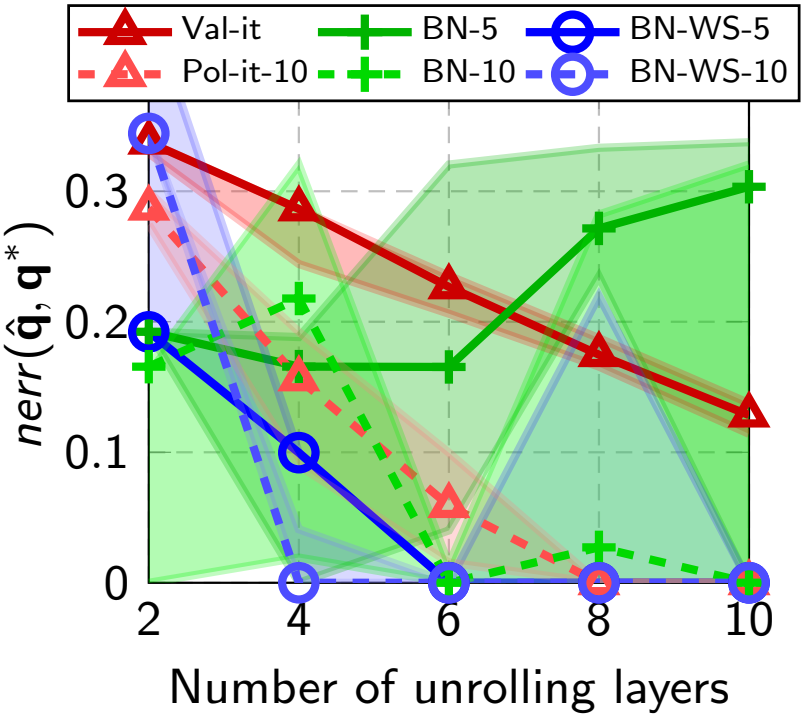
- We train BellNet to solve the cliff-walking problem

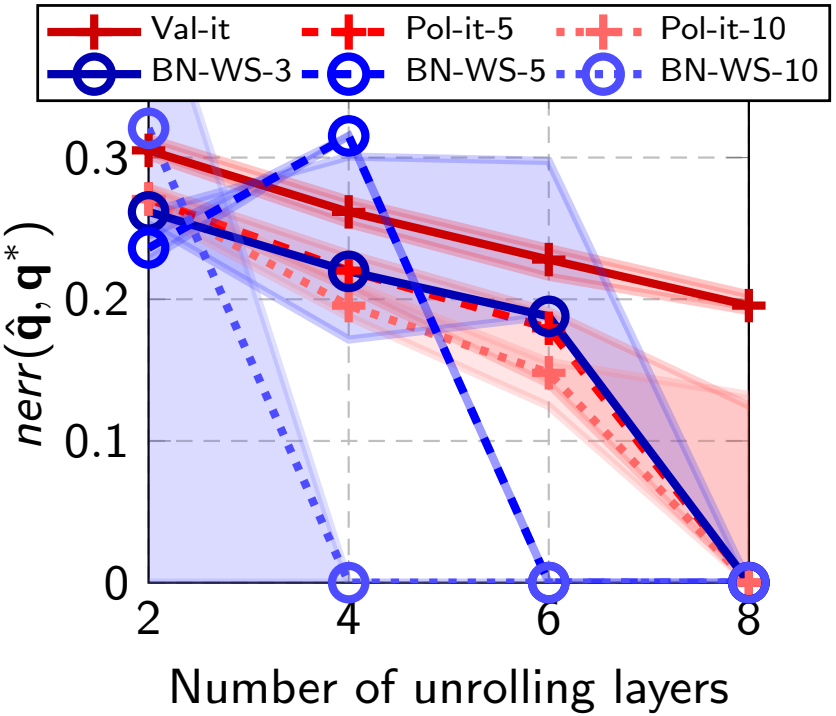


- We test BellNet by solving the mirrored cliff-walking problem



Numerical evaluation





Many thanks!