Exploiting Structure and Uncertainty of Bellman Updates in Markov Decision Processes

D. Tateo, C. D'Eramo, A. Nuara, M. Restelli, A. Bonarini

{davide.tateo, carlo.deramo, alessandro.nuara, marcello.restelli, andrea.bonarini}@polimi.it

Politecnico Milano 1863

Problem

- Learning is **difficult** in **highly stochastic** environments
- **Uncertainty** in action-value function estimates propagates
- Some algorithms face this problem focusing on the **bias** of the estimate
- Despite empirical evidence, there is **no proof** that focusing on the bias is the solution

Contributions

1. Split the estimate in **two components**:
   - The expected reward $\widetilde{R}(x,u)$
   - The expected next state value function $\widetilde{Q}(x,u)$
2. Use **different learning rates** for the two components
3. We provide **empirical results** showing the effectiveness of our approach

RQ-Learning Algorithm

Idea

To improve estimates, exploit:

Structure of the Bellman update — Uncertainty of the estimation

Approach

Split the action-value function in two components:

- $\widetilde{R}(x,u) = \mathbb{E}_{x'\sim\mathcal{P}(x'|x,u)}\left[r(x,u,x')\right]$
- $\widetilde{Q}(x,u) = \mathbb{E}_{x'\sim\mathcal{P}(x'|x,u)}\left[\max_{u'} Q^*(x',u')\right]$
- $Q^*(x,u) = \widetilde{R}(x,u) + \gamma\widetilde{Q}(x,u)$

Compute the update as follows:

- $\widetilde{R}_{t+1}(x,u) \leftarrow \widetilde{R}_t(x,u) + \alpha_t(R(x,u,x') - \widetilde{R}_t(x,u))$
- $\widetilde{Q}_{t+1}(x,u) \leftarrow \widetilde{Q}_t(x,u) + \beta_t(\max_{u'} Q_t(x',u') - \widetilde{Q}_t(x,u))$

Exploit the **variance of estimation** to set the learning rate:

1. Estimate the variance of the estimator $\widetilde{Q}$, using the sample variance of the target:

$$\boxed{\mathrm{Var}\left[\widetilde{Q}\right] \approx S_t^2\omega_t} \qquad \boxed{\omega_{t+1} = (1-\beta_t)^2\omega_t + \beta_t^2}$$

2. Compute the learning rate
   - Select a β that **decreases** when the estimate precision increases:

$$\boxed{\beta_t = \frac{\sigma_e^2(t)}{\sigma_e^2(t)+\eta}}$$

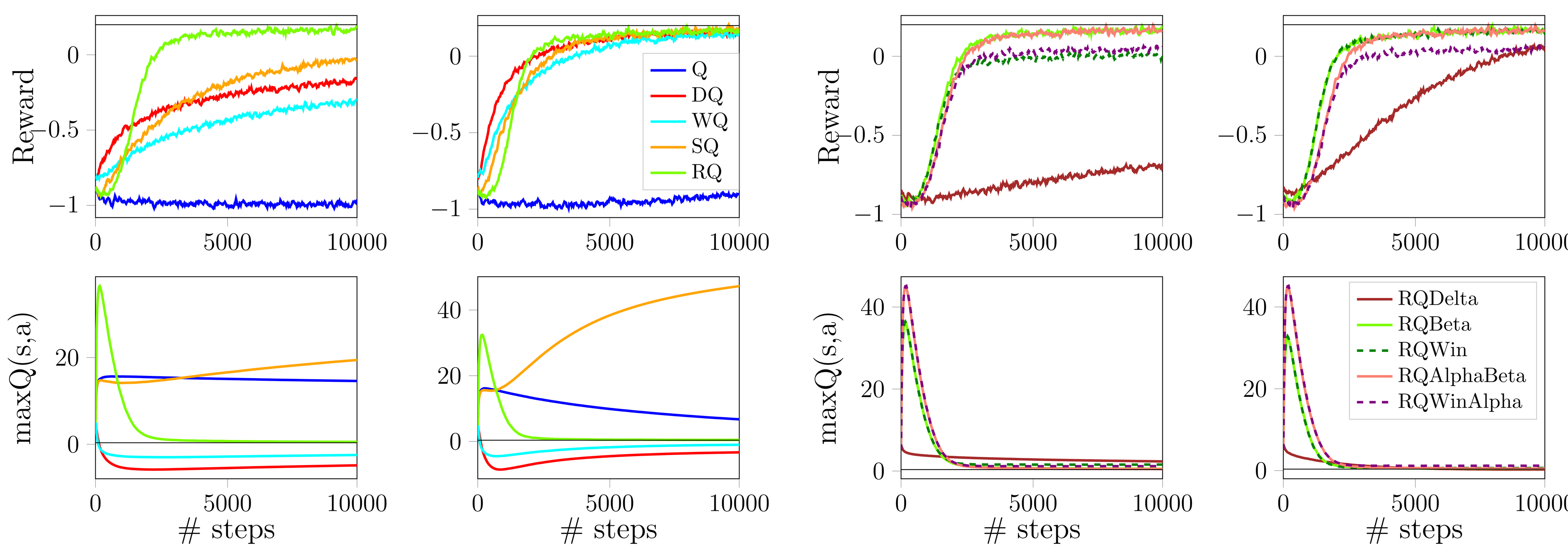
   - Or, select a δ that **increases** when the estimate precision increases:

$$\boxed{\delta_t = e^{\frac{\sigma_e^2}{\eta}\log\frac{1}{2}}}$$

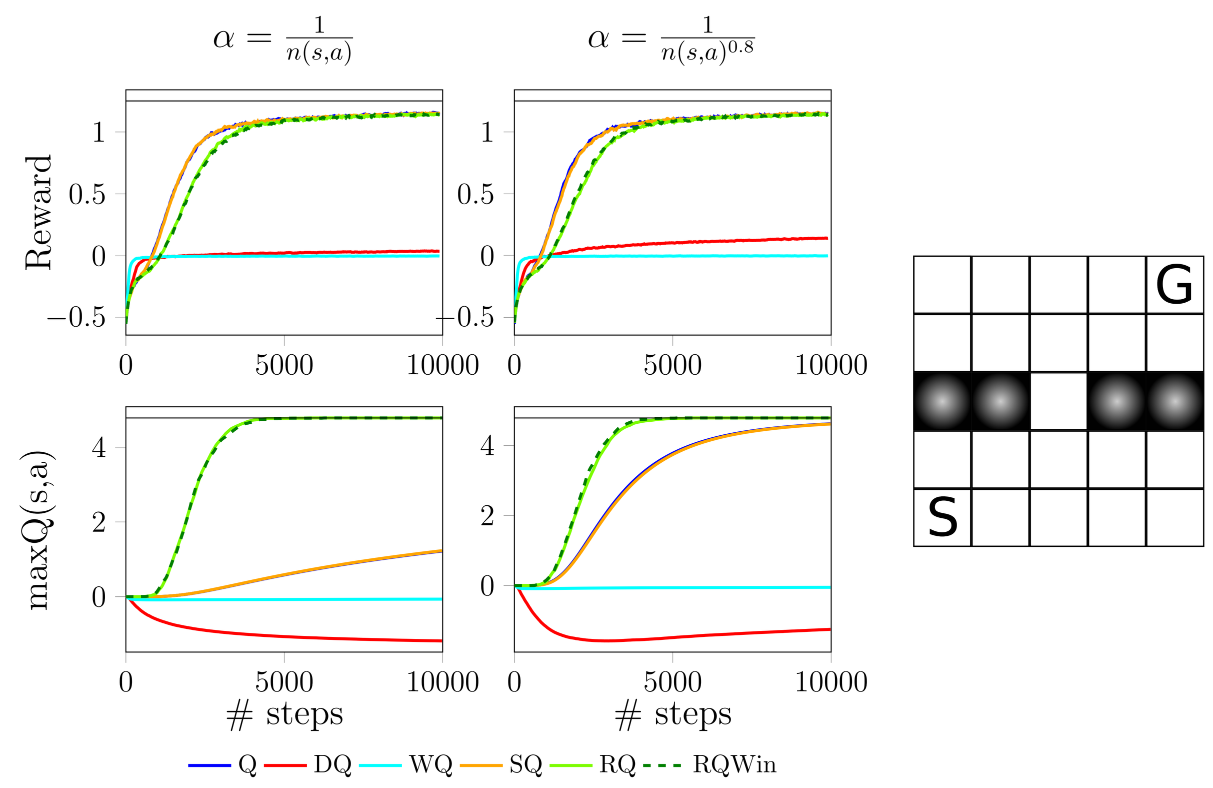
Q-Learning: $\alpha_t = \beta_t$ — RQ-Learning: $\beta_t \neq \alpha_t$ — RQ$_\delta$-Learning: $\beta_t = \alpha_t\delta_t$

Empirical Results

Noisy Gridworld



Gridworld with Holes



Double Chain