Convergence of regularized agent-state based Q-learning in POMDPs

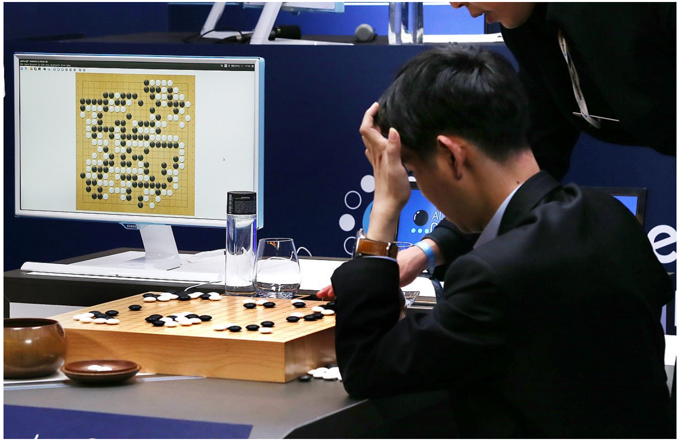
Amit Sinha (McGill), Matthieu Geist (Earth Species Project), Aditya Mahajan (McGill)

CDC 2025

- email: aditya.mahajan@mcgill.ca
- web: https://adityam.github.io

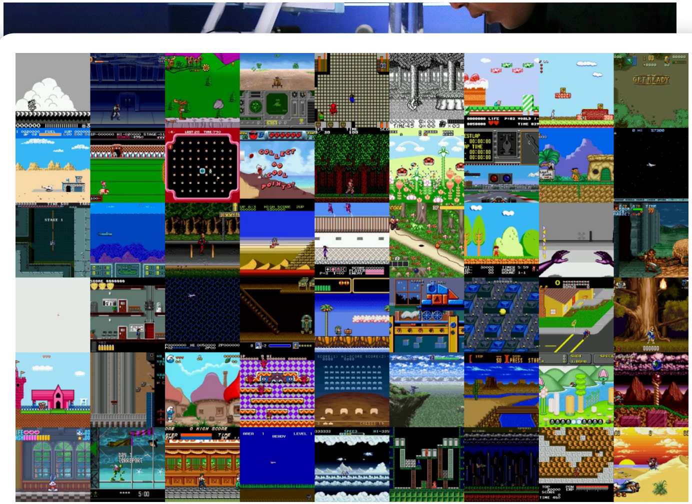
Recent successes of RL

Recent successes of RL



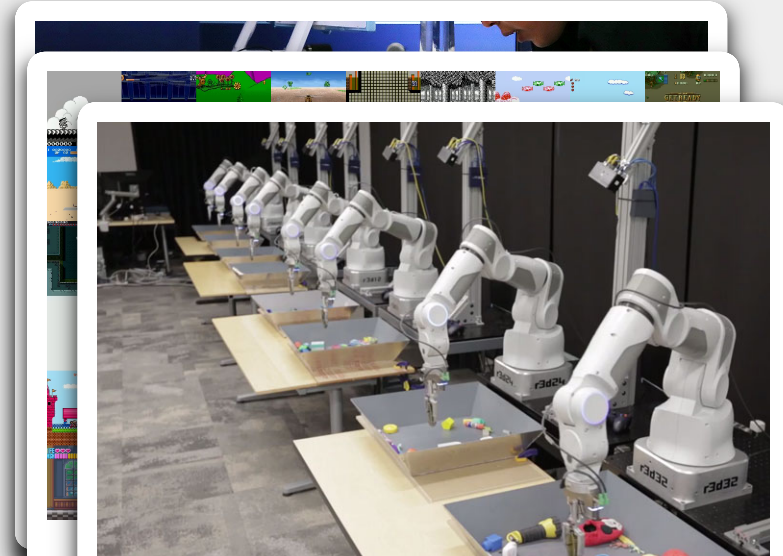
Alpha Go

Recent successes of RL

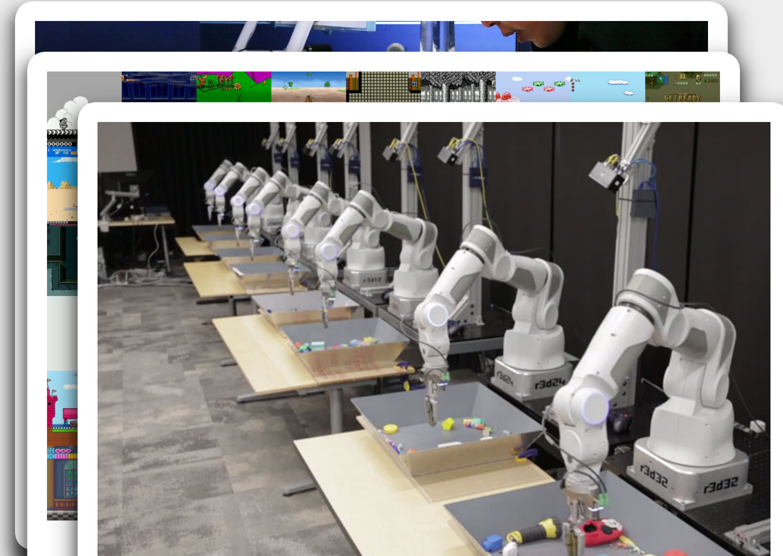


Arcade games

Recent successes of RL



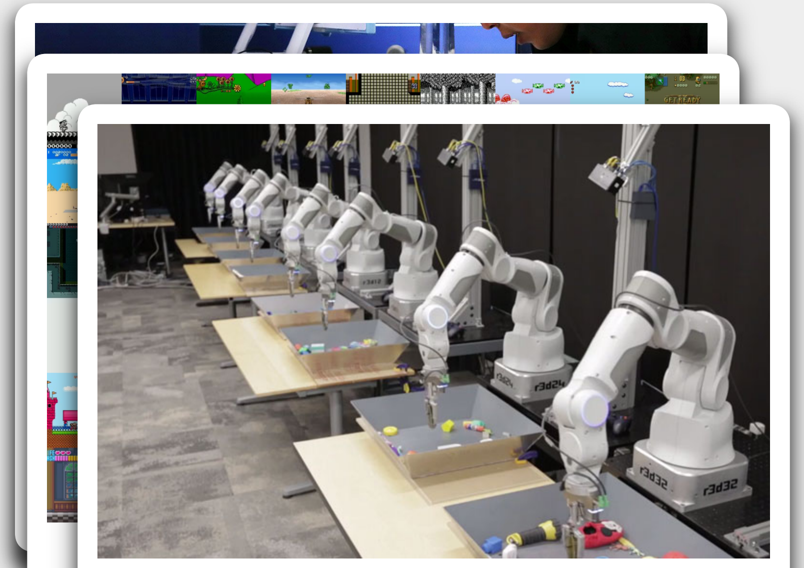
Robotic grasping



Robotic grasping

Recent successes of RL

- Algorithms based on comprehensive theory
- The theory is restricted almost exclusively to systems with perfect state observations



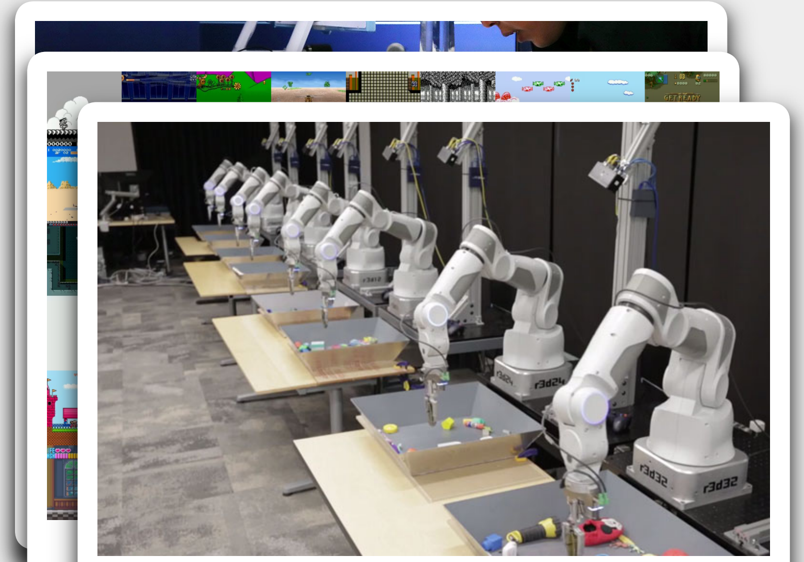
Robotic grasping

Recent successes of RL

- Algorithms based on comprehensive theory
- The theory is restricted almost exclusively to systems with perfect state observations

Many real-world applications are partially observed

- Healthcare
- Autonomous driving
- Finance (portfolio management)
- Retail and marketing



Robotic grasping

Recent successes of RL

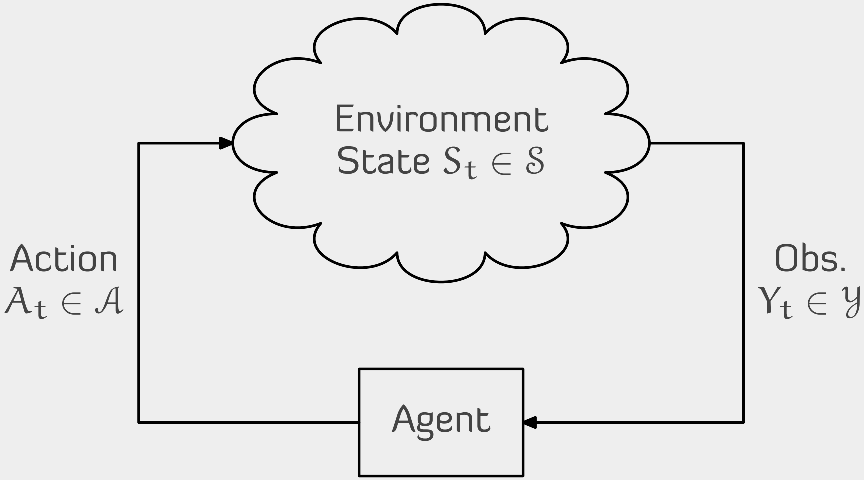
- Algorithms based on comprehensive theory
- The theory is restricted almost exclusively to systems with perfect state observations

Many real-world applications are partially observed

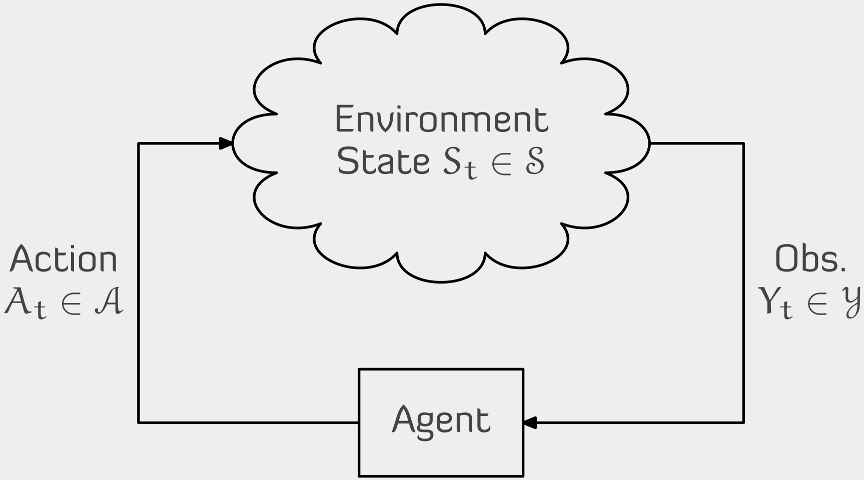
- Healthcare
- Autonomous driving
- Finance (portfolio management)
- Retail and marketing

How do we develop a theory for RL for partially observed systems?

POMDPs: Partially observable Markov decision processes



POMDPs: Partially observable Markov decision processes



$$\mathbb{P}(S_{t+1}, Y_{t+1} | S_{1:t}, Y_{1:t}, A_{1:t}) = \mathbb{P}(S_{t+1}, Y_{t+1} | S_t, A_t)$$

Reward: $R_t = r(S_t, A_t)$.

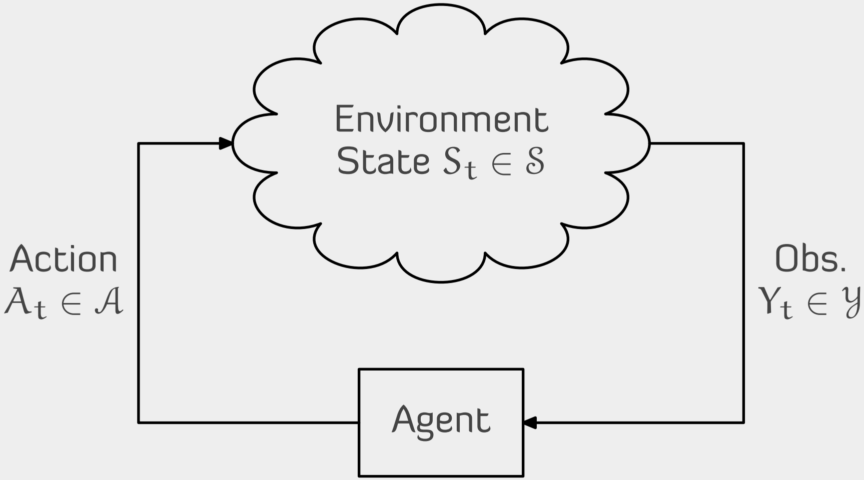
Policy: $\vec{\boldsymbol{\pi}} = (\vec{\pi}_1, \vec{\pi}_2, \dots)$ where

$$A_t \sim \vec{\pi}_t(Y_{1:t}, A_{1:t-1})$$

Performance:

$$J(\vec{\boldsymbol{\pi}}) := \mathbb{E}^{\vec{\boldsymbol{\pi}}}\Bigg[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \,\Bigg|\, S_1 \sim \xi_1\Bigg]$$

POMDPs: Partially observable Markov decision processes



$$\mathbb{P}(S_{t+1}, Y_{t+1} | S_{1:t}, Y_{1:t}, A_{1:t}) = \mathbb{P}(S_{t+1}, Y_{t+1} | S_t, A_t)$$

Reward: $R_t = r(S_t, A_t)$.

Policy: $\vec{\boldsymbol{\pi}} = (\vec{\pi}_1, \vec{\pi}_2, \dots)$ where

$$A_t \sim \vec{\pi}_t(Y_{1:t}, A_{1:t-1})$$

Performance:

$$J(\vec{\boldsymbol{\pi}}) := \mathbb{E}^{\vec{\boldsymbol{\pi}}}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \,\Big|\, S_1 \sim \xi_1 \Big]$$

Objective: Find the (history-dependent) policy $\vec{\boldsymbol{\pi}}$ that maximizes $J(\vec{\boldsymbol{\pi}})$

Review: Belief-state based planning

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner: $B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t)$.
- Belief state is sufficient to evaluate rewards: $\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t)$.

Thus, $\{B_t\}_{t \geqslant 1}$ is a perfectly observed controlled Markov process.

Astrom, “Optimal control of Markov processes with incomplete information,” JMAA 1965.
Stratonovich, “Conditional Markov Processes,” TVP 1960.

Review: Belief-state based planning

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner: $B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t)$.
- Belief state is sufficient to evaluate rewards: $\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t)$.

Thus, $\{B_t\}_{t \geqslant 1}$ is a perfectly observed controlled Markov process. Therefore:

Structure of optimal policy	There is no loss of optimality in choosing the action A_t as a function of the belief state B_t
Dynamic Program	The optimal control policy is given a DP with belief B_t as state.

Implications of the POMDP modeling framework

Implications for planning

- Allows the use of the MDP machinery for partially observed sys.
- Various exact and approximate algorithms to efficiently solve DP.

  Exact: incremental pruning, witness algorithm, linear support algo

  Approximate: QMDP, point based methods, SARSOP, DESPOT, . . .

Implications of the POMDP modeling framework

- Allows the use of the MDP machinery for partially observed sys.

Implications for learning

- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.

Implications of the POMDP modeling framework

▷ Allows the use of the MDP machinery for partially observed sys.

Implications for learning

- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.
- **On the theoretical side:**
  - Propose alternative methods: PSRs (predictive state representations), bisimulation metrics, . . .
  - Good theoretical guarantees, but difficult to scale.

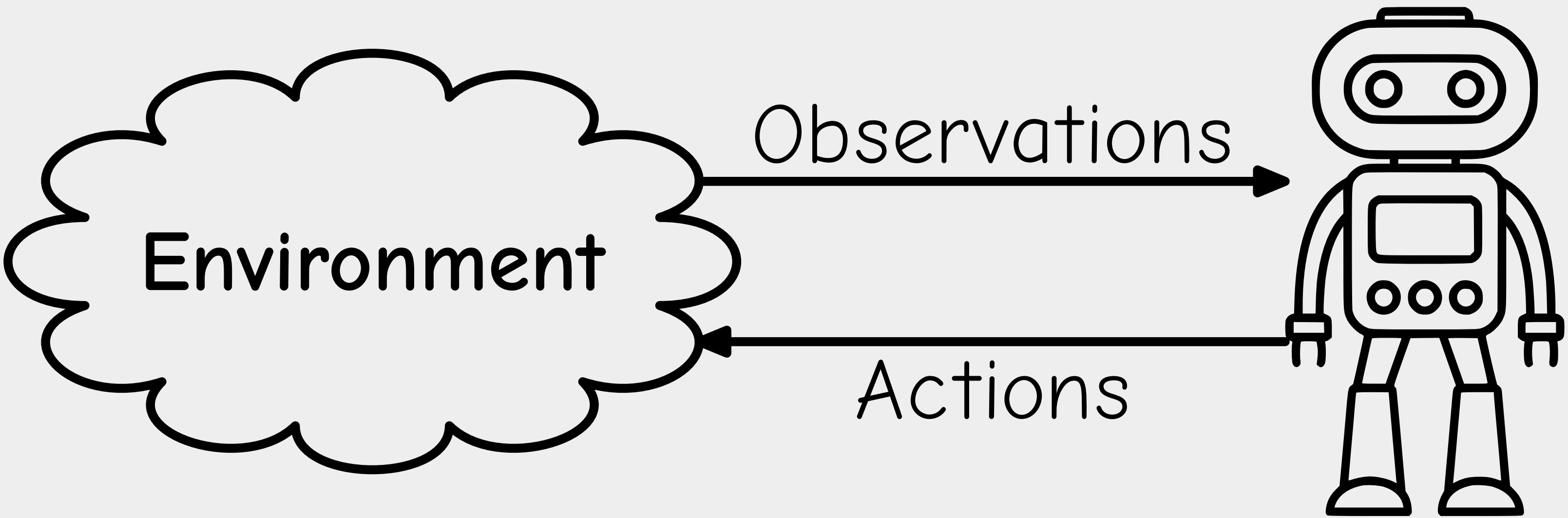
Implications of the POMDP modeling framework

- Allows the use of the MDP machinery for partially observed sys.

Implications for learning

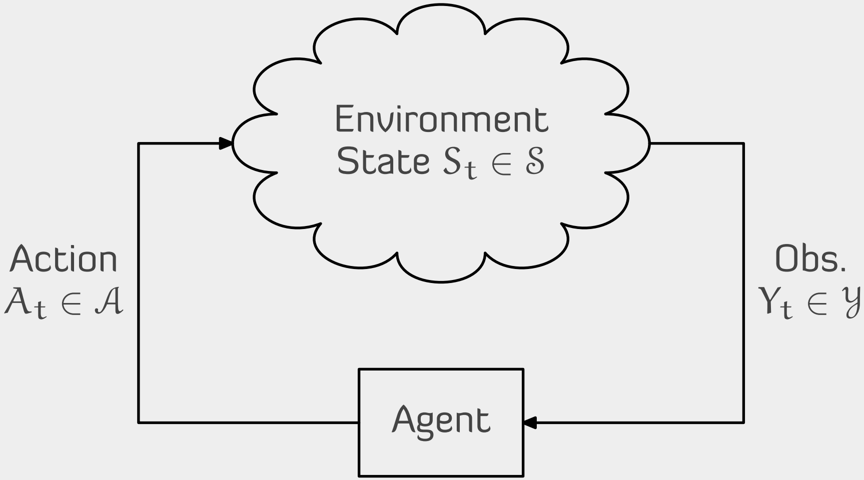
- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.
- **On the theoretical side:**
  - Propose alternative methods: PSRs (predictive state representations), bisimulation metrics, . . .
  - Good theoretical guarantees, but difficult to scale.
- **On the practical side:**
  - Simply stack the previous k observations and treat it as a “state”.
  - Instead of a CNN, use an RNN to model policy and action-value fn.
  - Can be made to work but lose theoretical guarantees and insights.

Deep RL learns agent-state based policies



Can we understand the convergence of such algorithms?

Abstract model of agent-state based policies



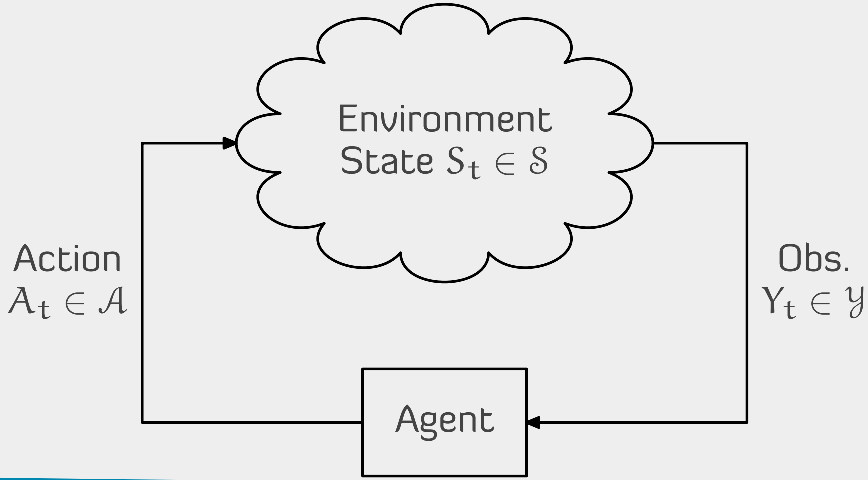
Agent state: $Z_t \in \mathcal{Z}$, where

$$Z_{t+1} = \phi(Z_t, Y_{t+1}, A_t)$$

Examples:

- $Z_t = (Y_{t-n:t}, A_{t-n:t-1})$
- Finite-state controllers
- Recurrent neural networks

Abstract model of agent-state based policies



Agent state: $Z_t \in \mathcal{Z}$, where

$$Z_{t+1} = \phi(Z_t, Y_{t+1}, A_t)$$

Examples:

- $Z_t = (Y_{t-n:t}, A_{t-n:t-1})$
- Finite-state controllers
- Recurrent neural networks

Regularized agent-state based Q-learning (RASQL)

- Act according to behavior policy $\mu \colon \mathcal{Z} \to \Delta(\mathcal{A})$
- Recursively update:

$$Q_{t+1}(z, a) = Q_t(z, a) + \alpha_t(z, a)\left[r_t + \gamma \Omega^*(Q_t(z_{t+1}, \cdot)) - Q_t(z, a)\right]$$

where Ω^* is **reward regularization** (replaces standard max operator).

Why regularization?

All emprirical algorithms use regularization in some form.

- to encourage exploration
- to stabilize the learning process by smoothing the optimization landscape
- to constrain policy update to stay similar to previous policy

Why regularization?

All emprirical algorithms use regularization in some form.

- to encourage exploration
- to stabilize the learning process by smoothing the optimization landscape
- to constrain policy update to stay similar to previous policy

Existing analysis of QL for POMDPs

Restricted to unregularized QL

- QL with finite window memory [Jaakkola Singh Jordan 1994, Kara Yuksel 2022]
- periodic QL with agent state [Sinha Geist Mahajan 2024]
- QL for non-Markovian environments [Kara Yuksel 2024, Chandak Shah Borkar 2024]

This paper: Analyze convergence of regularized agent-state based Q-learning (RASQL) for POMDPs

Key Tool: Legendre-Fenchel transform

Convex conjugate

For a strongly convex function $\Omega \colon \mathbb{R}^n \to \mathbb{R}$, its convex conjugate $\Omega^* \colon \mathbb{R}^n \to \mathbb{R}$ is defined as:

$$\Omega^*(q) = \max_{p \in \mathbb{R}^n} \left\{ \langle p, q \rangle - \Omega(p) \right\}.$$

- Key tool for analyzing convergence of regularized MDPs [Geist et al 2019]

Key Tool: Legendre-Fenchel transform

Convex conjugate

For a strongly convex function $\Omega\colon \mathbb{R}^n \to \mathbb{R}$, its convex conjugate $\Omega^*\colon \mathbb{R}^n \to \mathbb{R}$ is defined as:

$$\Omega^*(q) = \max_{p \in \mathbb{R}^n} \left\{ \langle p, q \rangle - \Omega(p) \right\}.$$

- Key tool for analyzing convergence of regularized MDPs [Geist et al 2019]

Example: Entropy regularization

- $\Omega(p) = \frac{1}{\beta} \sum p(a) \ln p(a)$
- $\Omega^*(q) = \frac{1}{\beta} \ln\left(\sum \exp(\beta q(a))\right)$

Key Tool: Legendre-Fenchel transform

Convex conjugate

For a strongly convex function $\Omega\colon \mathbb{R}^n \to \mathbb{R}$, its convex conjugate $\Omega^*\colon \mathbb{R}^n \to \mathbb{R}$ is defined as:

$$\Omega^*(q) = \max_{p \in \mathbb{R}^n} \left\{ \langle p, q \rangle - \Omega(p) \right\}.$$

- Key tool for analyzing convergence of regularized MDPs [Geist et al 2019]

Example: Entropy regularization

- $\Omega(p) = \frac{1}{\beta} \sum p(a) \ln p(a)$
- $\Omega^*(q) = \frac{1}{\beta} \ln\left(\sum \exp(\beta q(a)) \right)$

Example: KL regularization

- $\Omega(p) = \frac{1}{\beta} \sum p(a) \ln \frac{p(a)}{p_{\mathsf{ref}}(a)}$
- $\Omega^*(q) = \frac{1}{\beta} \ln\left(\sum p_{\mathsf{ref}}(a) \exp(\beta q(a)) \right)$

Key Tool: Legendre-Fenchel transform

Convex conjugate

For a strongly convex function $\Omega\colon \mathbb{R}^n \to \mathbb{R}$, its convex conjugate $\Omega^*\colon \mathbb{R}^n \to \mathbb{R}$ is defined as:

$$\Omega^*(q) = \max_{p \in \mathbb{R}^n} \{ \langle p, q \rangle - \Omega(p) \}.$$

- Key tool for analyzing convergence of regularized MDPs [Geist et al 2019]

Example: Entropy regularization

- $\Omega(p) = \frac{1}{\beta} \sum p(a) \ln p(a)$
- $\Omega^*(q) = \frac{1}{\beta} \ln \big(\sum \exp(\beta q(a)) \big)$

Example: KL regularization

- $\Omega(p) = \frac{1}{\beta} \sum p(a) \ln \frac{p(a)}{p_{\text{ref}}(a)}$
- $\Omega^*(q) = \frac{1}{\beta} \ln \big(\sum p_{\text{ref}}(a) \exp(\beta q(a)) \big)$

- **Important property:** The optimal regularized policy p^* (the argmax) can be obtained directly from the gradient of the convex conjugate:

$$p^* = \nabla \Omega^*(q)$$

Main step: construction of a limit MDP

Assumptions:

- The behavior policy μ induces an invariant distribution ζ_μ over (S_t, Y_t, Z_t, A_t).
- The learning rates $\{\alpha_t\}_{t \geqslant 1}$ satisfy: $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.

Main step: construction of a limit MDP

Assumptions:

- The behavior policy μ induces an invariant distribution ζ_μ over (S_t, Y_t, Z_t, A_t).
- The learning rates $\{\alpha_t\}_{t \geqslant 1}$ satisfy: $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.

Limit MDP

Construct a limit MDP with state space $\mathcal{Z}$ and action space $\mathcal{A}$ and

- Reward: $r_\mu(z, a) := \sum_{s \in \mathcal{S}} r(s, a)\, \zeta_\mu(s \mid z)$
- Dynamics: $P_\mu(z' | z, a) := \sum_{(s, y')} \mathbb{1}_{\{z' = \phi(z, y', a)\}} P(y' | s, a)\, \zeta_\mu(s | z)$

Main step: construction of a limit MDP

Assumptions:

- The behavior policy μ induces an invariant distribution ζ_μ over (S_t, Y_t, Z_t, A_t).
- The learning rates $\{\alpha_t\}_{t \geqslant 1}$ satisfy: $\sum_{t=1}^{\infty} \alpha_t = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$.

Limit MDP

Construct a limit MDP with state space $\mathcal{Z}$ and action space $\mathcal{A}$ and

- Reward: $r_\mu(z, a) := \sum_{s \in \mathcal{S}} r(s, a)\, \zeta_\mu(s \mid z)$
- Dynamics: $P_\mu(z'|z, a) := \sum_{(s, y')} \mathbb{1}_{\{z' = \phi(z, y', a)\}} P(y'|s, a)\, \zeta_\mu(s|z)$
- Let Q_μ be the unique fixed point of the regularized Bellman operator of this limit MDP:

$$Q_\mu(z, a) := r_\mu(z, a) + \gamma \sum_{z' \in \mathcal{Z}} P_\mu(z' \mid z, a)\, \Omega^*(Q_\mu(z', \cdot))$$

Main convergence result

Theorem

Under the stated assumptions, RASQL converges to Q_μ almost surely.

Main convergence result

Theorem

Under the stated assumptions, RASQL converges to Q_μ almost surely.

Salient features

- The converged policy is the greedy regularized policy with respect to Q_μ:

$$\pi^*(\cdot \mid z) = \nabla\Omega^*(Q_\mu(z, \cdot))$$

- This policy is typically stochastic (e.g., softmax for entropy regularization)
- Significant advantage over (unregularized) ASQL, which converges to deterministic policy as stochastic stationary policies can outperform deterministic stationary policies in POMDPs with agent-states.

Generalization: Regularized periodic Q-learning (RePASQL)

Why periodic policies?

- Time-varying policies can outperform time-invariant ones (as agent state is not info state)
- Periodic policies are simplest time-varying policies.

Generalization: Regularized periodic Q-learning (RePASQL)

Why periodic policies?

- Time-varying policies can outperform time-invariant ones (as agent state is not info state)
- Periodic policies are simplest time-varying policies.

Regularized periodic Q-learning (RePASQL)

Fix a period L and pick a periodic behavior policy μ with period L. Recursively update:

$$Q^{\ell}_{t+1}(z, a) = Q^{\ell}_{t}(z, a) + \alpha^{\ell}_{t}(z, a)\left[r_t + \gamma\Omega^{*}(Q^{[\![\ell+1]\!]}_{t}(z', \cdot)) - Q^{\ell}_{t}(z, a)\right]$$

where $[\![\ell]\!] = \ell \bmod L$.

Generalization: Regularized periodic Q-learning (RePASQL)

Why periodic policies?

- Time-varying policies can outperform time-invariant ones (as agent state is not info state)
- Periodic policies are simplest time-varying policies.

Regularized periodic Q-learning (RePASQL)

Fix a period L and pick a periodic behavior policy μ with period L. Recursively update:

$$Q^{\ell}_{t+1}(z,a) = Q^{\ell}_t(z,a) + \alpha^{\ell}_t(z,a)\left[r_t + \gamma\Omega^*(Q^{[\![\ell+1]\!]}_t(z',\cdot)) - Q^{\ell}_t(z,a)\right]$$

where $[\![\ell]\!] = \ell \bmod L$

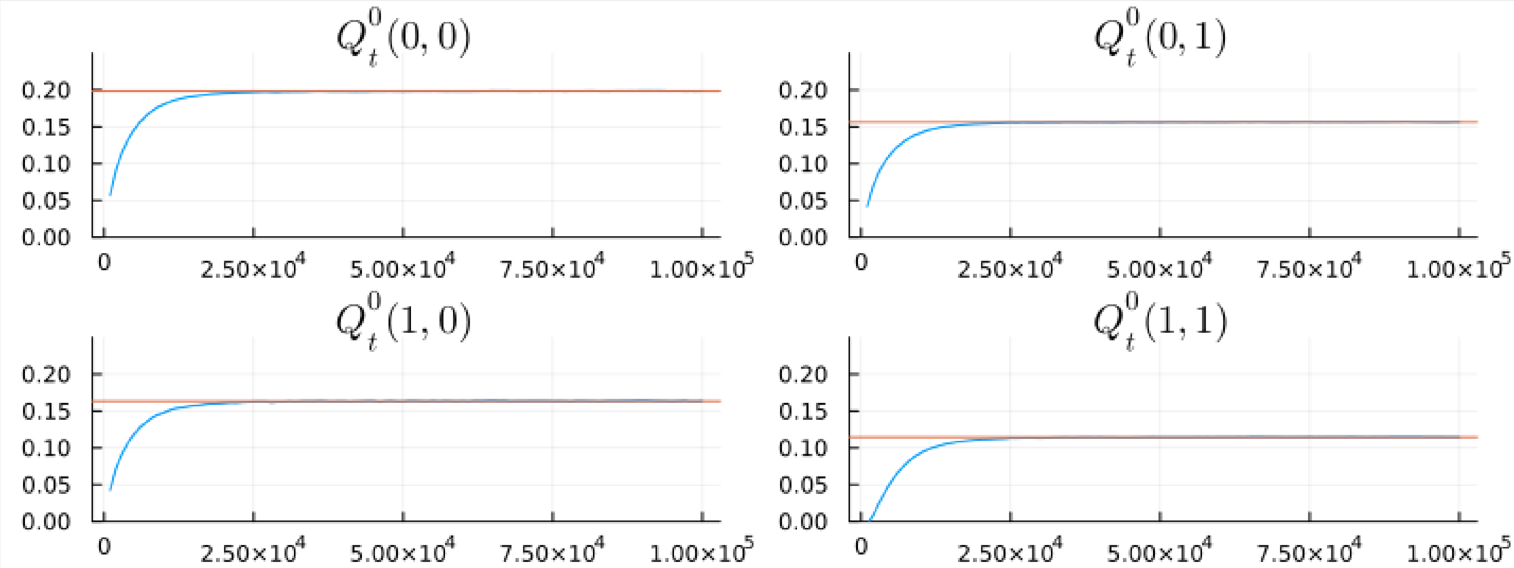
Theorem

Assuming (S_t, Y_t, Z_t, A_t) has periodic limit, then RePASQL converges to the fixed point of a periodic regularized MDP almost surely.

Numerical experiments

Verifying convergence of RASQL

- Small POMDP with $|\mathcal{S}| = 4$, $|\mathcal{Y}| = 2$, $|\mathcal{A}| = 2$
- Agent state $Z_t = Y_t$
- Median and quantile over 25 seeds

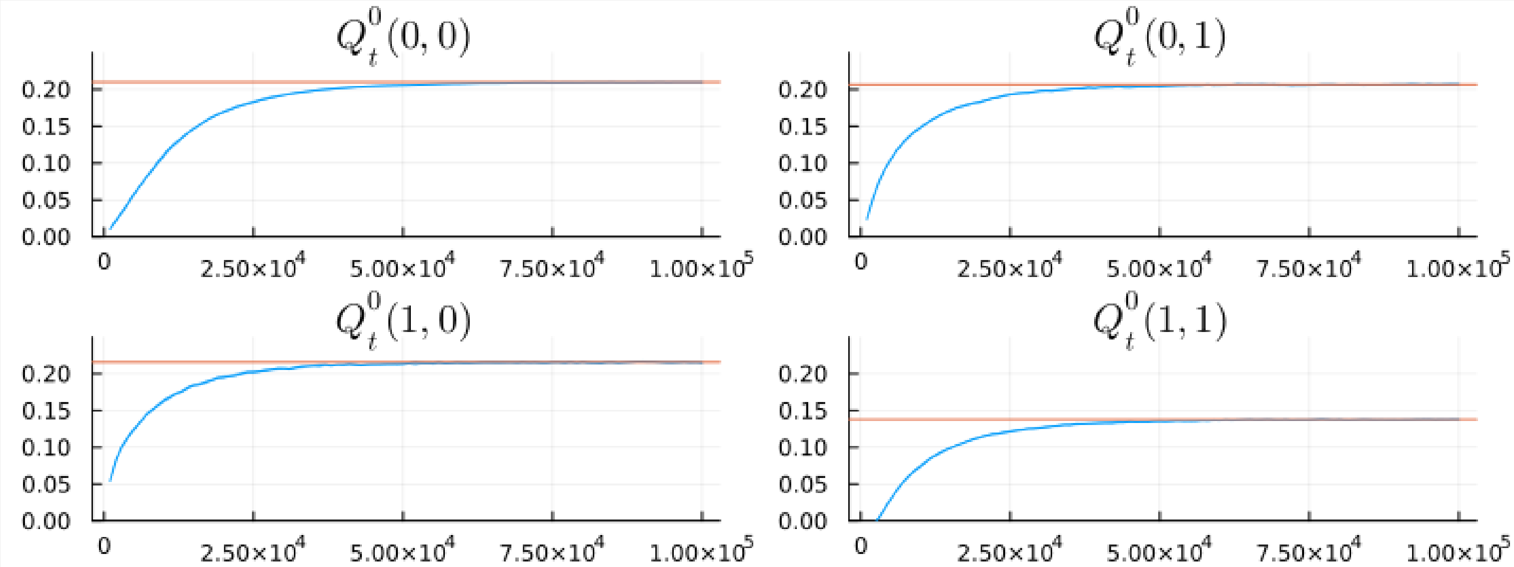


Blue: RASQL iterates Red: Theoretical limit.

Verifying convergence of RePASQL

- Same model as before
- Period $L = 2$

- $\ell = 0$

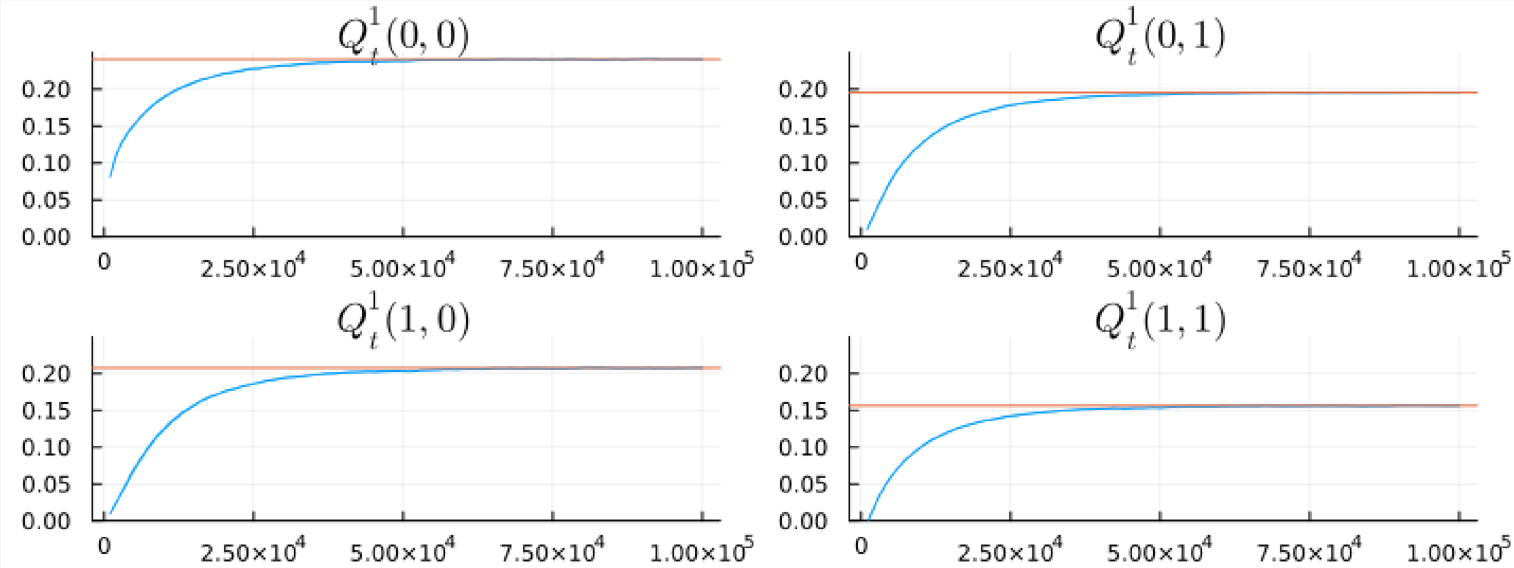


Blue: RePASQL iterates Red: Theoretical limit.

Verifying convergence of RePASQL

- Same model as before
- Period $L = 2$

- $\ell = 1$



Blue: RePASQL iterates Red: Theoretical limit.

Conclusion

- Proved almost sure convergence of **regularized** agent-state based Q-learning
- Iterates converge to the fixed point of a limit MDP (or a periodic limit MDP), which y depend on the invariant distribution (or periodic limit distribution) induced by the exploration policy.
- Regularization helps in learning stochastic policies, which can outperform deterministic ones in POMDPs with agent-states.

Conclusion

- Proved almost sure convergence of **regularized** agent-state based Q-learning
- Iterates converge to the fixed point of a limit MDP (or a periodic limit MDP), which y depend on the invariant distribution (or periodic limit distribution) induced by the exploration policy.
- Regularization helps in learning stochastic policies, which can outperform deterministic ones in POMDPs with agent-states.

- email: aditya.mahajan@mcgill.ca
- web: https://adityam.github.io

Thank you