The Mean-Squared Error of Double Q-Learning

(To appear in NeurIPS 2020)

R. Srikant

ECE/CSL

University of Illinois at Urbana-Champaign

Collaborators

Wentao Weng
Tsinghua University

Harsh Gupta
UIUC

Niao He
UIUC

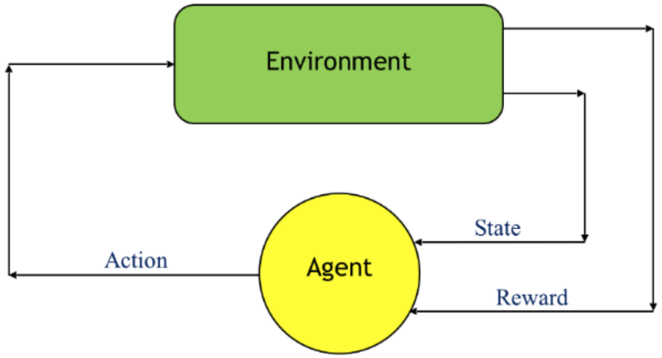
Lei Ying
Michigan

MDPs

- Dynamical System:

$$X_{k+1} = f(X_k, u_k, w_k)$$

X_k: state (finite state space), u_k: control action, w_k: noise



- Reward Function:

$$\sum_{k\geq 0} \gamma^k E(r(X_k, u_k))$$

- Problem: Find $u_k = \mu(X_k)$ to maximize the reward function

Q-learning

The max operator leads to overestimation of the next state Q-value!

- Q-function :

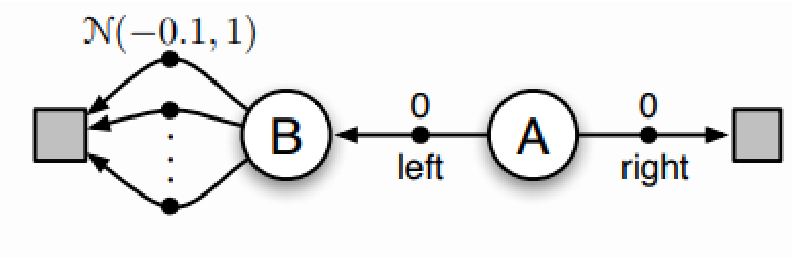
$$Q(i,u) = r(i,u) + \gamma\ max_a\ E(Q \ldots a)|X_k = i)$$

- If the conditional expectation can be calcul d, then this fixed-point equation can be solved
  - This requires knowledge of $p_{ij}(a)$, the probabilit r the Markov chain transitioning from state i to state j under action
- What if the transition probabilities are unknow Q-Learning

$$Q_{k+1}(i_k, u_k) = (1 - \alpha_k)Q_k(i_k, u_k) + \alpha_k\{r(i_k, u_k) + \gamma\ max_a\ Q_k(s_{k+1}, a)\}$$

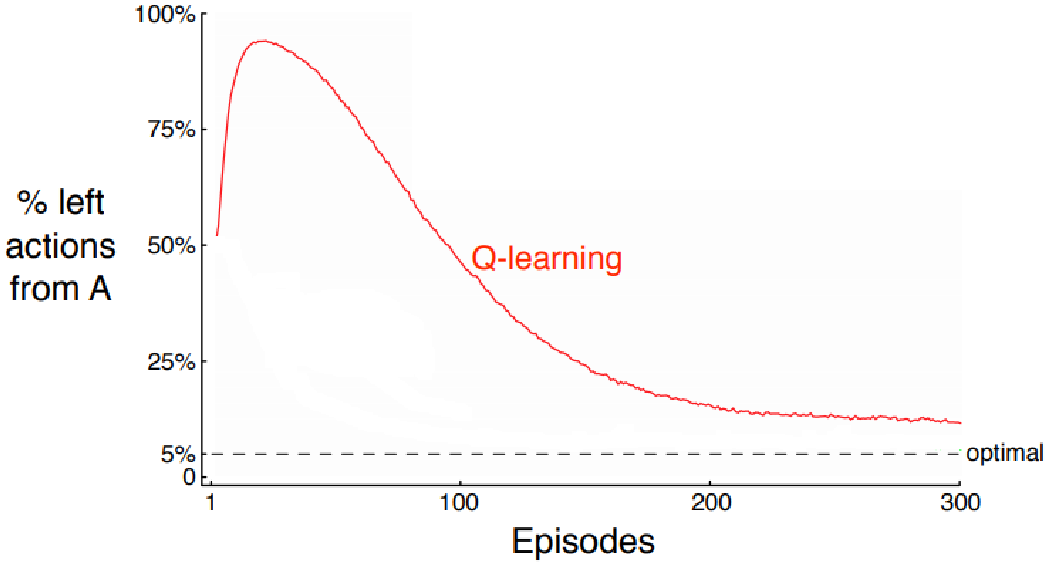
Maximization Bias (van Hasselt, 2010)

- Consider the following MDP (Sutton and Barto, 2nd Edition):



- How does Q-learning do?

➢ Q-learning takes the left action much more often than the right action due to maximization bias!



How to Fix the Problem?

- Need to estimate $\max_i E(Y_i)$
- Suppose we have twenty samples of Y_i for each i
- One estimator is $\max_i \hat{\mu}_i$
- Another estimator
  - Divide the twenty samples into two batches of 10 each, call the corresponding empirical mean estimates $\hat{\mu}_i^1$ and $\hat{\mu}_i^2$
  - Find $i^* = \arg\max \hat{\mu}_i^1$
  - Estimate $\max_i E(Y_i)$ as $\hat{\mu}_{i^*}^2$
- The first estimator overestimates, the second one underestimates

Double Q-learning

The max operator leads to overestimation of the next state Q-value!

- Q-learning :

$$Q_{k+1}(i_k, u_k) = (1 - \alpha_k) Q_k(i_k, u_k) + \alpha_k \{r(i_k, u_k) + \gamma \, max_a \, Q_k(s_{k+1}, a)\}$$

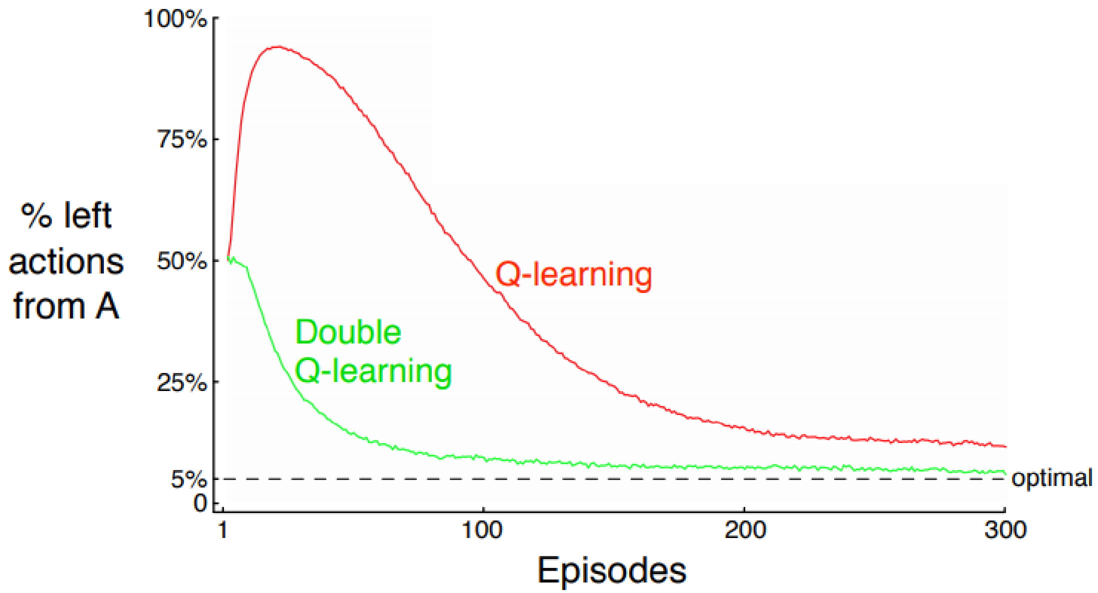
- Bootstrapping leads to maximization bias.
- **Double Q-learning:** Use two Q-learning estimates! $\beta_k \in \{0,1\}$

$$Q^A_{k+1}(i_k, u_k) = Q^A_k(i_k, u_k) + \alpha_k \beta_k \{r(i_k, u_k) + \gamma \, max_a \, Q^{\boldsymbol{B}}_k(s_{k+1}, a) - Q^A_k(i_k, u_k)\}$$

$$Q^B_{k+1}(i_k, u_k) = Q^B_k(i_k, u_k) + \alpha_k (1 - \beta_k) \{r(i_k, u_k) + \gamma \, max_a \, Q^{\boldsymbol{A}}_k(s_{k+1}, a) - Q^B_k(i_k, u_k)\}$$

Double Q-learning

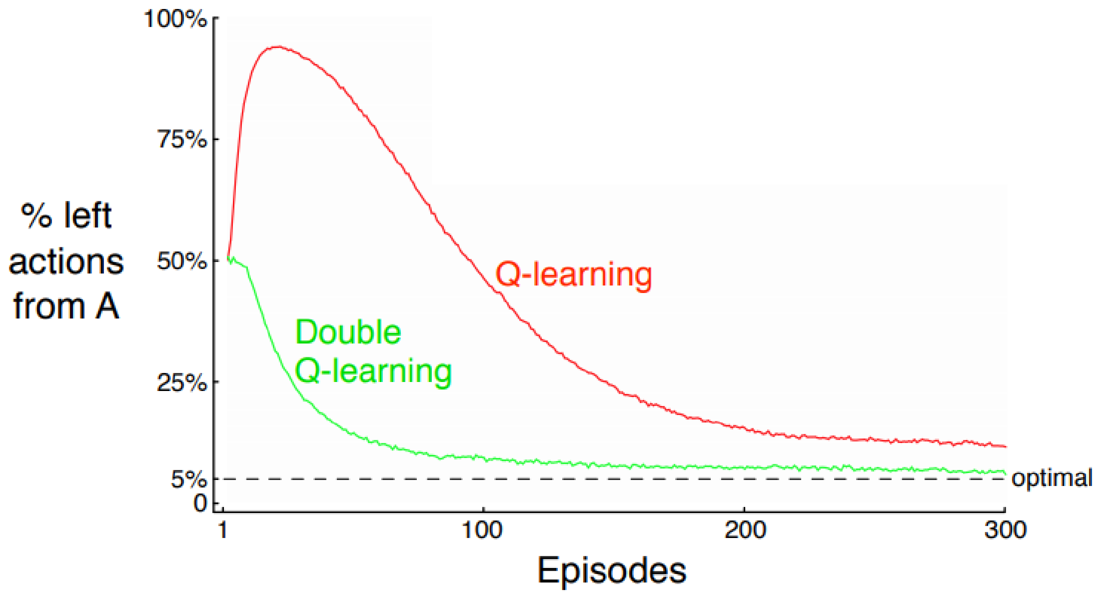
- Advantage: Faster transient performance due to reduced maximization bias.



Double Q-learning

- Advantage: Faster transient performance due to reduced maximization bias.

- Disadvantage: In problems where the maximization bias does not matter, Double Q-learning does not perform well as well as Q-learning and its asymptotic mean-squared error is worse



Goals

- Use asymptotic mean-squared error as the metric (??) and derive conditions on the learning rates such that Double Q-learning has the same asymptotic mean-squared error as Q-learning

- Use these conditions and study using experiments whether the transient performance of Double Q-learning is better without sacrificing asymptotic mean-square error

Simplest Reinforcement Learning Problem

- Motivated by Devraj and Meyn (2017), we analyze a simpler problem
- Fix a policy $u_k = \mu(X_k)$ and evaluate the value function:

$$V(i) = c(i) + \gamma E(V(X_{k+1}) | X_k = i)$$

- Using TD learning
- But which policy? We prove a result for all policies, which then holds for the optimal policy

Value Function Approximation

- Linear function approximation (in this talk)

State i

$$\left(\Phi_1(i), \Phi_2(i)\right)\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} = V(i)$$

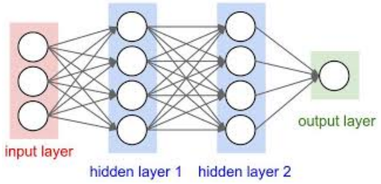
features parameters

Learn $\boldsymbol{\theta}$ instead of V

- Deep neural network (nonlinear)

State i

i



$= V(i)$

Learn $\boldsymbol{\theta}$ (parameters) instead of V

Standard TD Learning

- Assume $\mathrm{V}(i) \approx \theta^T \phi(i)$
  - ϕ is a known feature vector of dimension much smaller than the state space

- TD learning with linear function approximation (Sutton, 1988):

$$\theta_{k+1} = \theta_k - \epsilon_k \phi(X_k)\left(\phi^T(X_k)\theta_k - c(X_k) - \gamma \phi^T(X_{k+1})\theta_k\right)$$

  - Special case: $\phi(i) = e_i$ reduces to tabular TD learning

Standard TD - Linear Stochastic Approximation

- (Tsitisiklis and van Roy, 1998): With appropriate centering, the TD algorithm can be written as

$$\theta_{k+1} = \theta_k + \epsilon_k\big(A(X_k)\theta_k + b(X_k)\big),$$

where

$$\bar{A} = E(A(X_\infty)) \text{ is Hurwitz and } \bar{b} = E\big(b(X_\infty)\big) = 0$$

Double TD Learning

- Double TD learning with linear function approximation:

$$\theta_{k+1}^{\mathrm{A}} = \theta_k^{\mathrm{A}} - \beta_k \delta_k \phi(X_k)\left(\phi^T(X_k)\theta_k^{\mathrm{A}} - c(X_k) - \gamma\phi^T(X_{k+1})\theta_k^{\mathrm{B}}\right)$$

$$\theta_{k+1}^{\mathrm{B}} = \theta_k^{\mathrm{B}} - (1-\beta_k)\delta_k \phi(X_k)\left(\phi^T(X_k)\theta_k^{\mathrm{B}} - c(X_k) - \gamma\phi^T(X_{k+1})\theta_k^{\mathrm{A}}\right)$$

Double TD – Linear Stochastic Approximation

- Linear Stochastic Approximation (LSA):

$$\mathrm{U}_{k+1} = \mathrm{U}_k + \delta_k\big(A_D(X_k)\mathrm{U}_k + b_D(X_k)\big),$$

where $U_k = [\theta_k^{\mathrm{A}}, \theta_k^{\mathrm{B}}], \bar{A}_D = E\big(A_D(X_\infty)\big)$ is Hurwitz and $\bar{b}_D = E\big(b_D(X_\infty)\big) = 0$

- The asymptotic mean-squared error of LSA has been studied recently in Chen, Devraj, Busic, Meyn (2020) for linear stochastic approximation

Asymptotic Mean-Squared Error

- Assume $\theta^* = \mathbf{0}$
- AMSE of Double TD-Learning:
  - $AMSE(\theta^A) = \lim_{k\to\infty} kE\left[\left(\theta_k^A\right)^T \theta_k^A\right]$
  - $AMSE(\theta^B) = \lim_{k\to\infty} kE\left[\left(\theta_k^B\right)^T \theta_k^B\right]$
  - $AMSE\left(\frac{\theta^A+\theta^B}{2}\right) = \lim_{n\to\infty} \frac{k}{4} E\left[\left(\theta_k^A + \theta_k^B\right)^T \left(\theta_k^A + \theta_k^B\right)\right]$
- AMSE of Standard TD-Learning:
  - $AMSE(\theta) = \lim_{k\to\infty} kE\left[\theta_k^T \theta_k\right]$

Main Result (Double Q-learning)

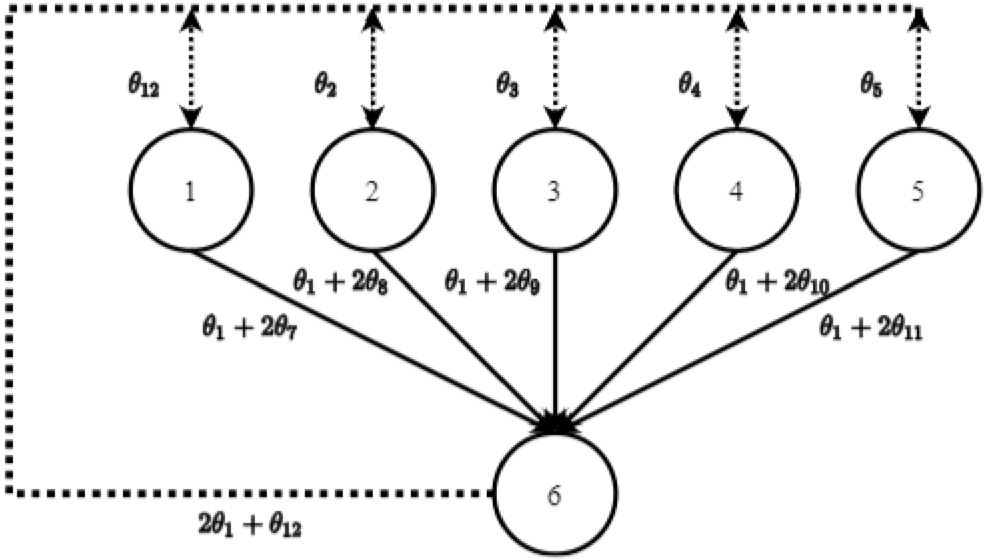
- Let step-size(Double Q) = 2 × step-size(Standard Q):

$$AMSE(\theta^A) = AMSE(\theta^B) > AMSE(\theta)$$
$$AMSE\left(\frac{\theta^A + \theta^B}{2}\right) = AMSE(\theta)$$

- Double Q-learning with twice the step as Q-learning, with its two outputs averaged has the same AMSE as Q-learning

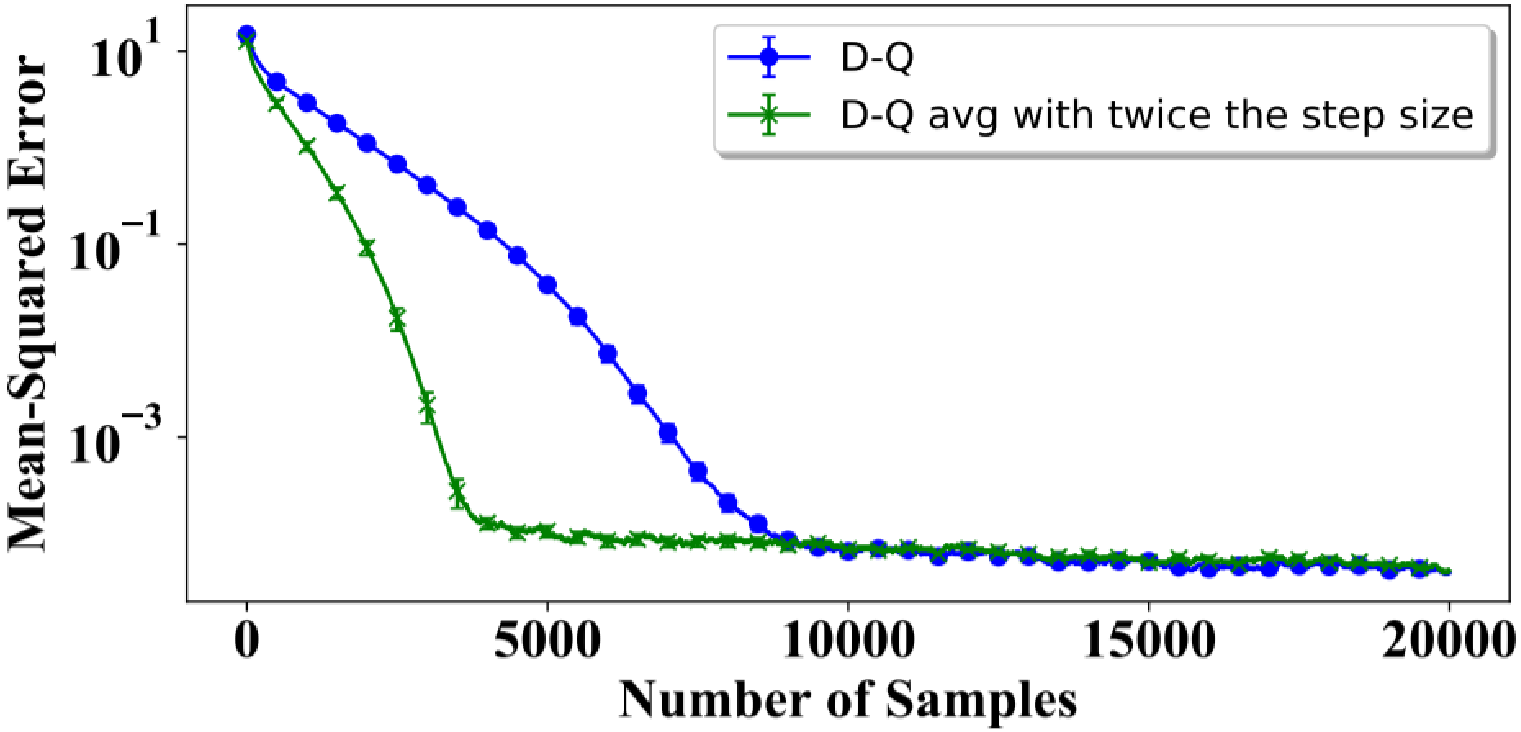
Baird's Example

- MDP with six states
- Action: dashed or solid transitions
- Reward: randomly sampled from [-0.05,0.05]
- Linear function approximation
  - As specified in the graph
  - Example: $Q(6, dashed) \approx 2\theta_1 + \theta_{12}$
- Discount factor $\gamma = 0.8$



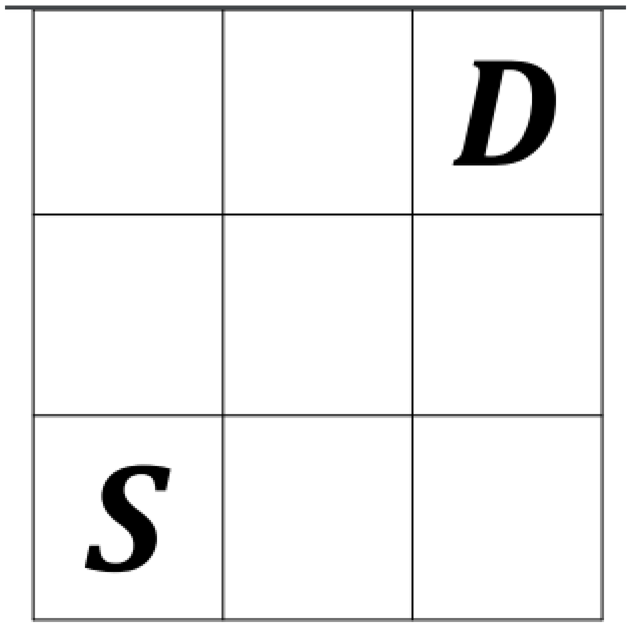
Baird's Example – Results

- Mean-squared error: $||\theta - \theta^*||_2^2$ (Step sizes for Q and D-Q are $\frac{1000}{\#samples+10000}$)



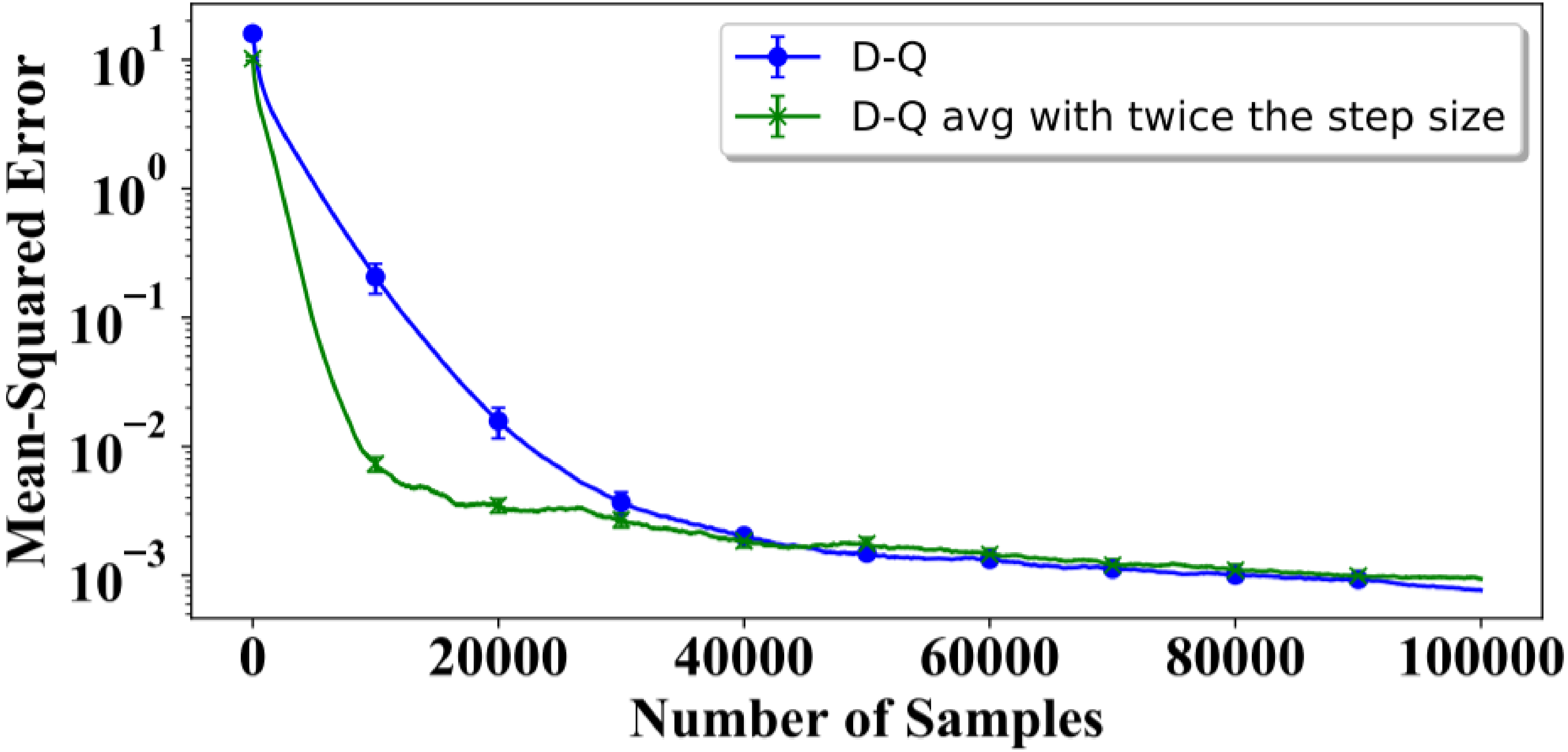
GridWorld

- Environment: $n \times n$ grid
- Actions: up, down, left, right
  - Each action has a 30% error probability.
- Reward: +1 at D, -0.001 for other steps
- Termination: walk outside the grid, or arrive at D
- Tabular Q-Learning: $\phi(s, a) = e_{s,a}$

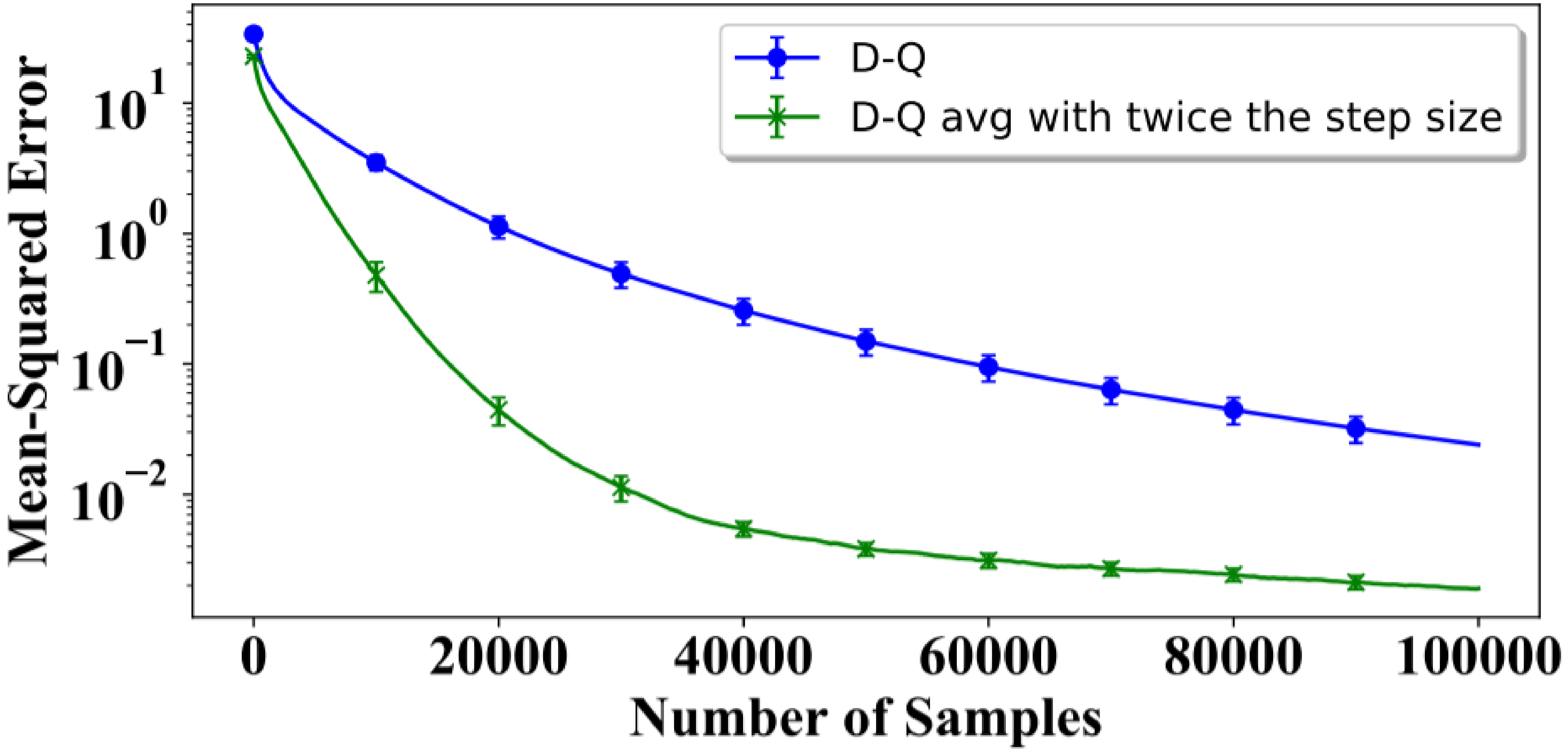


A 3x3 GridWorld

GridWorld (3x3) – Results



GridWorld (4x4) – Results



Linear Stochastic Approximation (LSA)

- General Linear Stochastic Approximation (Chen et al., 2020):

$$\xi_{k+1} = \xi_k + \frac{g}{k}\left(A(Y_k)\xi_k + b(Y_k)\right)$$

- Assume $\xi_k \to \mathbf{0}$
- $\Sigma_\infty = \lim_{k\to\infty} kE\left[\xi_k \xi_k^T\right]$, $\Sigma_\mathrm{b} = \sum_{k=2}^{\infty} E[b(Y_k)b(Y_1)^T]$, $\bar{A} = E[A(Y(\infty)]$

Main Result: If $\frac{1}{2}I + gA$ is Hurwitz,

$$\left(\frac{1}{2}I + gA\right)\Sigma_\infty + \Sigma_\infty\left(\frac{1}{2}I + gA^T\right) + g^2\Sigma_\mathrm{b} = 0$$

Outline of the Proof: Standard TD-Learning

- Recall $\theta_{k+1} = \theta_k + \frac{g}{k}\left(A(X_k)\theta_k + b(X_k)\right)$

- Lyapunov equation:

$$\left(\frac{1}{2}I + g\bar{A}\right)\Sigma_\infty^S + \Sigma_\infty^S\left(\frac{1}{2}I + g\bar{A}\right)^T + g^2\Sigma_b = 0$$

where $\Sigma_\infty^S = \lim_{k\to\infty} kE[\theta_k\theta_k^T]$.

Outline of the Proof: Double TD-learning

- Recall $U_k = [\theta_k^A; \theta_k^B], U_{k+1} = U_k + \frac{2g}{k}\left(A_D\,(X_k)U_k + b_D(X_k)\right)$
- Lyapunov equation:

$$\left(\frac{1}{2}I + 2g\bar{A}_D\right)\Sigma_\infty^D + \Sigma_\infty^D\left(\frac{1}{2}I + 2g\bar{A}_D\right)^T + g^2\Sigma_b^D = 0$$

where $\Sigma_\infty^D = \lim_{k\to\infty} kE\left[U_k U_k^T\right]$.

- "Guess": for some matrix V, C,

$$\Sigma_\infty^D = \begin{bmatrix} V & C \\ C & V \end{bmatrix}$$

Connection between $\bar{A}, \bar{A}_D$

- $\mathrm{E}[A(X_k)] = \gamma E[\phi(X_k)\phi(X_{k+1})^T] - E[\phi(X_k)\phi(X_k)^T] = A_2 - A_1$

- $E[A_D(X_k)] = E\begin{bmatrix} -\beta_k\phi(X_k)\phi(X_k)^T & \beta_k\gamma\phi(X_k)\phi(X_{k+1})^T \\ (1-\beta_k)\gamma\phi(X_k)\phi(X_{k+1})^T & (1-\beta_k)\phi(X_k)\phi(X_k)^T \end{bmatrix}$

$$= \frac{1}{2}\begin{bmatrix} -A_1 & A_2 \\ A_2 & -A_1 \end{bmatrix}$$

Proof outline: compare Lyapunov equations

- Double TD-learning: with some manipulations

$$\left(\frac{1}{2}I + g\bar{A}\right)\frac{V+C}{2} + \frac{V+C}{2}\left(\frac{1}{2}I + g\bar{A}\right)^T + g^2\Sigma_b = 0$$

- Recall that for TD-Learning:

$$\left(\frac{1}{2}I + g\bar{A}\right)\Sigma_\infty^S + \Sigma_\infty^S\left(\frac{1}{2}I + g\bar{A}\right)^T + g^2\Sigma_b = 0$$

- Uniqueness implies $\frac{V+C}{2} = \Sigma_\infty^S$.

Proof outline: back to AMSE

- $AMSE\left(\frac{\theta^A+\theta^B}{2}\right) = \lim_{k\to\infty} \frac{k}{4} E\left[\left(\theta_k^A + \theta_k^B\right)^T\left(\theta_k^A + \theta_k^B\right)\right]$
$= \frac{1}{2}\lim_{k\to\infty} \mathrm{trace}\left(kE\left[\theta_k^A\left(\theta_k^A\right)^T\right] + kE\left[\theta_k^A\left(\theta_k^B\right)^T\right]\right)$
$= \frac{1}{2}\mathrm{trace}(V + C)$
$= \mathrm{trace}\left(\Sigma_\infty^S\right)$
$= AMSE(\theta)$
- $AMSE\left(\theta^A\right) > AMSE(\theta)$?
  - Show $\mathrm{trace}(V) > \mathrm{trace}(C)$

Conclusions

- Showed that an averaged estimator of Double Q-Learning with twice the step-size has the same (asymptotic) mean-squared error as Q-Learning
  - But each estimator from Double Q-Learning is not as good
- Possible step-size guideline for Double Q-Learning
  - Doubling the step size
- Transient Analysis, Nonlinear Function Approximation??
  - Finite time analysis of Double Q-learning: Xiong, Zhao, Liang, Zhang (NeurIPS 2020)