On Optimization Formulations and Algorithms of Markov Decision Problems

Lexing Ying

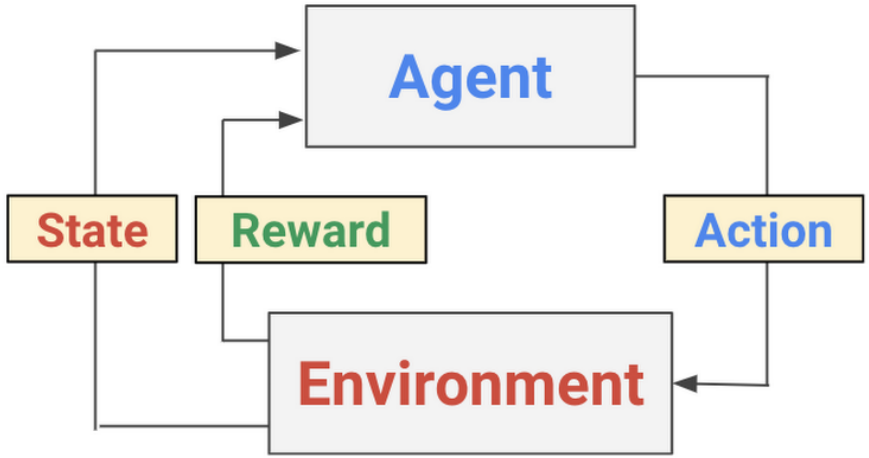
work with Haoya Li, Yuhua Zhu, Samarth Gupta, Hsiangfu Yu, Inderjit Dhillon

MATH-IMS Joint Applied Mathematics Colloquium

Chinese University of Hong Kong

Feb 4, 2022

Markov decision problems



- Reinforcement learning (RL), online decision making
- Game playing: AlphaGo, AlphaZero
- Robotics
- Science: AlphaFold

This talk: optimization formulations and algorithms

MDP

Consider a Markov decision process (MDP) $M = (S, A, P, r, \gamma)$

- S: state space
- A: action space
- P: transition tensor, P^a_{st} probability from s to t under action a
- r: reward, r^a_s reward at s under action a

Policy π

- π^a_s: probability of taking action a at state s
- $\sum_{a \in A} \pi^a_s = 1$

Under a fixed π, the behavior follows P^π and r^π

$$P^\pi_{st} = \sum_{a \in A} P^a_{st} \pi^a_s, \quad r^\pi_s = \sum_{a \in A} r^a_s \pi^a_s$$

The value function $v^\pi_s \equiv \mathbb{E}\left[\sum_{m=0}^{\infty} \gamma^m r^{a_m}_{s_m} | s_0 = s\right]$ where

- $(s_0, a_0, s_1, a_1, s_2, a_2, \ldots), \quad a_m \sim \pi^A_{s_m}, s_{m+1} \sim P^{a_m}_{s_m, S}$

$v^\pi \in \mathbb{R}^{|S|}$ satisfies

$$v^\pi = r^\pi + \gamma P^\pi v^\pi$$

Goal: maximize the value function!

Content

Optimization formulations

- Primal: value (policy) iteration
- Primal-dual: actor-critic
- Dual: policy gradient

New optimization algorithms

- Quasi-Newton policy gradient (for dual form)
- Accelerated primal-dual algorithm (for primal-dual form)

Background: linear programming forms

Primal (e positive)

$$\min_{v: b - Av \leq 0} e^{\mathsf{T}} v$$

Primal dual

$$\min_{v} \max_{u \geq 0} e^{\mathsf{T}} v + u^{\mathsf{T}} (b - Av)$$

(by minimax theorem)

$$\max_{u \geq 0} \min_{v} e^{\mathsf{T}} v + u^{\mathsf{T}} (b - Av)$$

Dual

$$\max_{u \geq 0, A^{\mathsf{T}} u = e} b^{\mathsf{T}} u$$

Background: entropy regularized version

(Genearalized) Shannon entropy, convex

$$h(u) = \sum_i u_i \log \frac{u_i}{\sum_j u_j} = \sum_i u_i \log u_i - \left(\sum_i u_i\right) \log\left(\sum_i u_i\right)$$

Primal dual

$$\min_v \max_{u>0} e^\mathsf{T} v + u^\mathsf{T}(b - Av) - h(u)$$

$$\max_{u>0} \min_v e^\mathsf{T} v + u^\mathsf{T}(b - Av) - h(u)$$

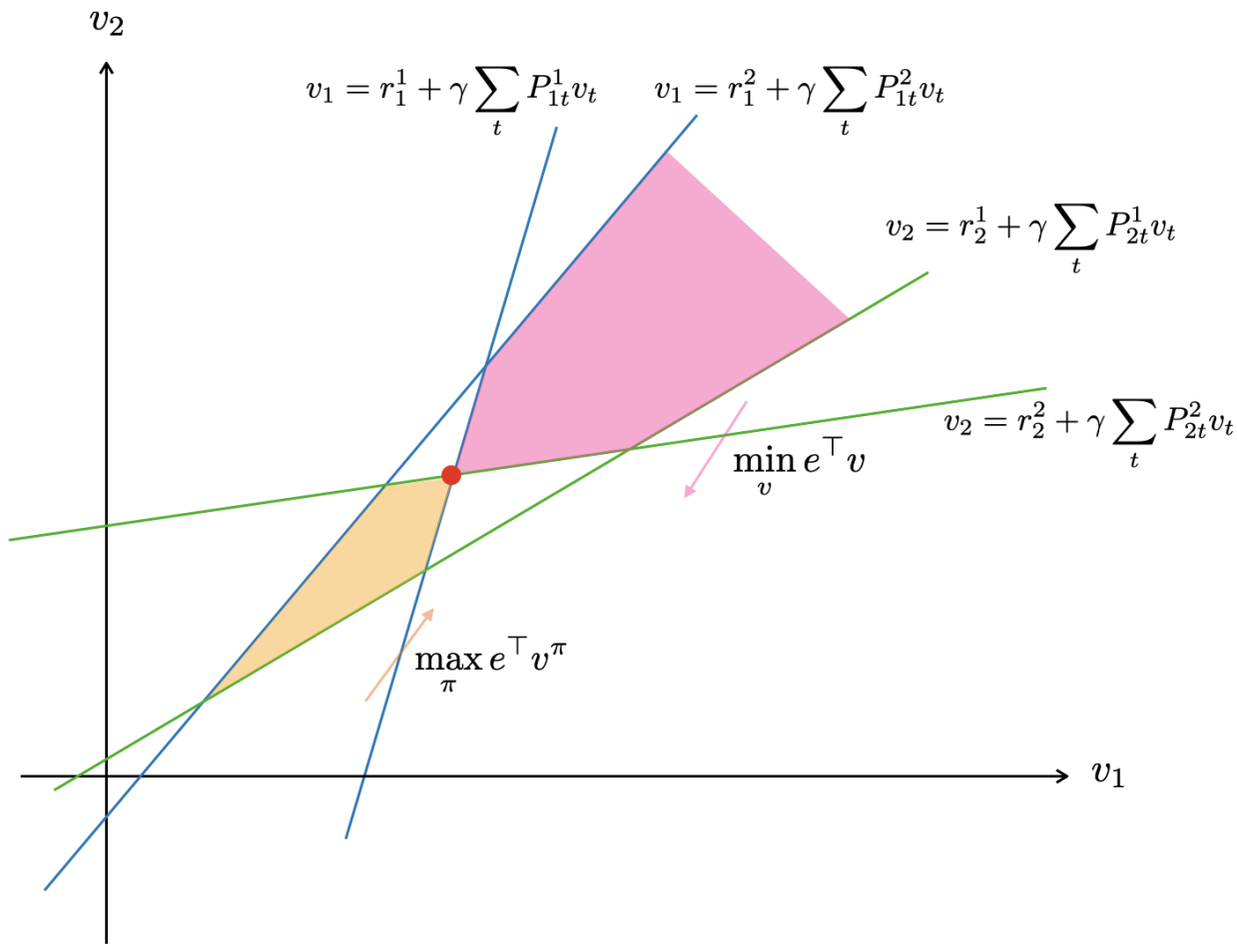
Primal

$$\min_{1^\mathsf{T} \exp(b-Av) \leq 1} e^\mathsf{T} v$$

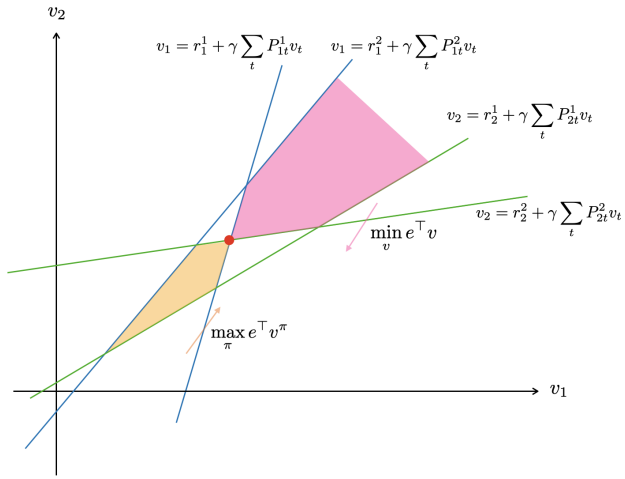
Dual

$$\max_{A^\mathsf{T} u = e} b^\mathsf{T} u - h(u)$$

Going back to MDP



Non-regularized MDP



Primal

$$\min_v \sum_{s \in S} e_s v_s, \text{ s.t. } \forall a, s, \; r_s^a - \left(v_s - \gamma \sum_{t \in S} P_{st}^a v_t \right) \leq 0$$

Primal-dual (u_s^a Lagrange multipilers)

$$\min_{v_s} \max_{u_s^a \geq 0} \sum_{s \in S} e_s v_s + \sum_{s,a} (r_s^a + \gamma \sum_{t \in S} P_{st}^a v_t - v_s) u_s^a$$

$$\min_v \max_{u^a \geq 0} \; e^\mathsf{T} v + \sum_{a \in A} (u^a)^\mathsf{T} (r^a + \gamma P^a v - v)$$

Dual

$$\max_{u^a \geq 0} \sum_{a \in A} (r^a)^\mathsf{T} u^a, \text{ s.t. } \sum_{a \in A} (I - \gamma (P^a)^\mathsf{T}) u^a = e$$

Connections with RL algorithms

Primal:

$$\min_{v} \sum_{s\in S} e_s v_s, \text{ s.t. } \forall a, s, \ r_s^a - \left(v_s - \gamma \sum_{t\in S} P_{st}^a v_t \right) \le 0$$

- KKT condition is equivalent to Bellman equation

$$\begin{cases} r_s^a + \gamma \sum_{t\in S} P_{st}^a v_t^* \le v_s^*, & \text{for } \forall s, a \\ \sum_{a\in A} (u_s^a - \gamma \sum_{t\in S} P_{ts}^a u_t^a) = e_s, & \text{for } \forall s \\ u_s^a (r_s^a + \gamma \sum_{t\in S} P_{st}^a v_t^* - v_s^*) = 0 & \text{for } \forall s, a \end{cases} \Leftrightarrow v_s^* = \max_{a\in A} \left(r_s^a + \gamma \sum_{t\in S} P_{st}^a v_t^* \right)$$

- Algorithm: value/policy iteration

Dual:

$$\max_{u^a \ge 0} \sum_{a\in A} (r^a)^\mathsf{T} u^a, \text{ s.t. } \sum_{a\in A} (I - \gamma (P^a)^\mathsf{T}) u^a = e$$

- Let $u_s^a = w_s \pi_s^a$. Then $(I - \gamma (P^\pi)^\mathsf{T}) w = e$, $\sum_{a\in A} (r^a)^\mathsf{T} u^a = (r^\pi)^\mathsf{T} w^\pi$

$$\max_{\pi} e^\mathsf{T} (I - \gamma P^\pi)^{-1} r^\pi$$

- Algorithm: policy gradient

Entropy regularized MDP

$$h(u_s) = \sum_{a \in A} u_s^a \log \frac{u_s^a}{\sum_{b \in A} u_s^b}$$

Primal-dual

$$\min_{v_s} \sup_{u_s^a > 0} \sum_{s \in S} e_s v_s + \sum_{s,a} \left(r_s^a + \gamma \sum_{t \in S} P_{st}^a v_t - v_s \right) u_s^a - \tau \sum_{s \in S} h(u_s)$$

Primal

$$\min_{v_s} \; e^{\mathsf{T}} v, \; \text{s.t.} \; \forall s, v_s \geq \tau \log \left(\sum_{a \in A} \exp \left(\frac{r_s^a + \gamma \sum_{t \in S} P_{st}^a v_t}{\tau} \right) \right)$$

Dual

$$\sup_{u^a > 0} \sum_{a \in A} (r^a)^{\mathsf{T}} u^a - \sum_{s \in S} h(u_s), \; \text{s.t.} \; \sum_{a \in A} (I - \gamma (P^a)^{\mathsf{T}}) u^a = e$$

Connections with RL algorithms

Primal

$$\min_{v_s} \; e^{\mathsf{T}} v, \;\; \text{s.t. } \forall s, v_s \geq \tau \log \left(\sum_{a \in A} \exp \left(\frac{r_s^a + \gamma \sum_{t \in S} P_{st}^a v_t}{\tau} \right) \right)$$

- KKT condition is equivalent to Bellman equation

$$v_s^* = \tau \log \left(\sum_{a \in A} \exp \left(\frac{r_s^a + \gamma \sum_{t \in S} P_{st}^a v_t^*}{\tau} \right) \right)$$

- Algorithm: value/policy iteration

Dual

$$\sup_{u^a > 0} \sum_{a \in A} (r^a)^{\mathsf{T}} u^a - \sum_{s \in S} h(u_s), \;\; \text{s.t.} \;\; \sum_{a \in A} (I - \gamma (P^a)^{\mathsf{T}}) u^a = e$$

- Let $u_s^a = w_s \pi_s^a$. Then $(I - \gamma (P^\pi)^{\mathsf{T}}) w = e \; \sum_{a \in A} (r^a)^{\mathsf{T}} u^a = (r^\pi)^{\mathsf{T}} w^\pi$

$$\max_{\pi} e^{\mathsf{T}} (I - \gamma P^\pi)^{-1} (r^\pi - \tau h^\pi)$$

where $h_s^\pi = \sum_{a \in A} \pi_s^a \log \pi_s^a$

- Algorithm: policy gradient

Content

Optimization formulations

- Primal: value (policy) iteration
- Primal-dual: actor-critic
- Dual: policy gradient

New optimization algorithms

- 1. Quasi-Newton policy gradient (for dual form)
- 2. Accelerated natural gradient ascent descent (for primal-dual form)

New algorithm 1

$$\max_{\pi} e^{\mathsf{T}}(I - \gamma P^{\pi})^{-1}(r^{\pi} - \tau h^{\pi})$$

Consider gradient-type dynamics

- 1st order gradient descent: slow
- 2nd order Newton method: Hessian too expensive to estimate

Our contribution: quasi-Newton

- Diagonal approximation of Hessian
- As cheap as a 1st-order method
- Converges like 2nd-order methods: super exponential convergence

Hessian approximation

Gradient

$$\frac{\partial E}{\partial \pi_s^a} = (r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v^\pi]_s + c_s)(w^\pi)_s,$$

Hessian

$$\frac{\partial^2 E}{\partial \pi_s^a \partial \pi_t^b} = H_{st}^{ab} + \text{remainder}$$

Theorem

For the choice

$$H_{st}^{ab}(\pi) = -\tau \delta_{st}^{ab} \frac{(w_\pi)_s}{\pi_s^a},$$

$$\|\partial^2 E(\pi) - H(\pi)\| = O(\|\pi - \pi^*\|), \quad \pi^* \textit{ exact soln}$$

Preconditioned gradient descent

$$\frac{\mathrm{d}\pi_s^a}{\mathrm{d}t} = -(H^{-1}\nabla_\pi E)_s^a = \pi_s^a(r_s^a - \tau(\log \pi_s^a + 1) - [(I - \gamma P^a)v^\pi]_s + c_s)/\tau.$$

In terms of $\theta_s^a = \log \pi_s^a$

$$\frac{\mathrm{d}\theta_s^a}{\mathrm{d}t} = (r_s^a - \tau(\theta_s^a + 1) - [(I - \gamma P^a)v^\pi]_s + c_s)/\tau.$$

Quasi-Newton algorithm

$$\frac{\mathrm{d}\theta_s^a}{\mathrm{d}t} = (r_s^a - \tau(\theta_s^a + 1) - [(I - \gamma P^a)v^\pi]_s + c_s)/\tau.$$

Taking explicit Euler with step size η

$$\theta_s^a \leftarrow \eta(r_s^a - \tau - [(I - \gamma P^a)v^\pi]_s + c_s)/\tau + (1-\eta)\theta_s^a$$

Back in terms of π

$$\pi_s^a \leftarrow (\pi_s^a)^{1-\eta} \exp(\eta(r_s^a + (\gamma P^a v^\pi)_s)/\tau)$$

When $\eta = 1$

$$\pi_s^a \leftarrow \exp((r_s^a + (\gamma P^a v^\pi)_s)/\tau)$$

and this is entropy regularized natural policy gradient method

Super exponential convergence

Theorem

With step size $\eta = 1$,

$$\left\| \pi^{(k+1)} - \pi^* \right\| \leq C \left\| \pi^k - \pi^* \right\|^2,$$

Our approach generalizes to entropies of form

$$(h^\pi)_s = \sum_a \phi\left(\frac{\pi_s^a}{\gamma_a}\right) \gamma_a,$$

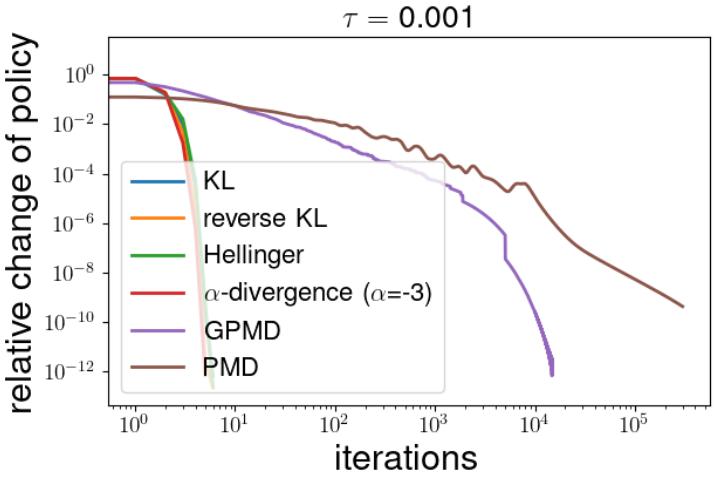
where

- ϕ is convex on $(0, +\infty)$ and $\phi(1) = 0$,
- γ is a prior distribution over A.

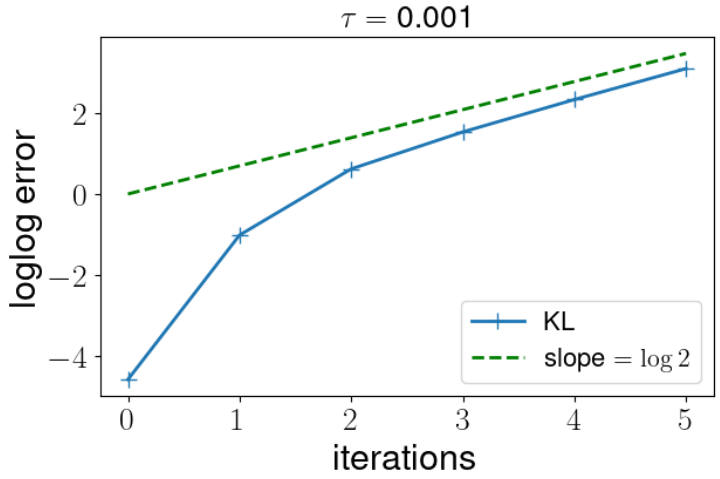
Examples

- $\phi(x) = x \log x$, KL-divergence
- Reverse KL-divergence
- Hellinger divergences

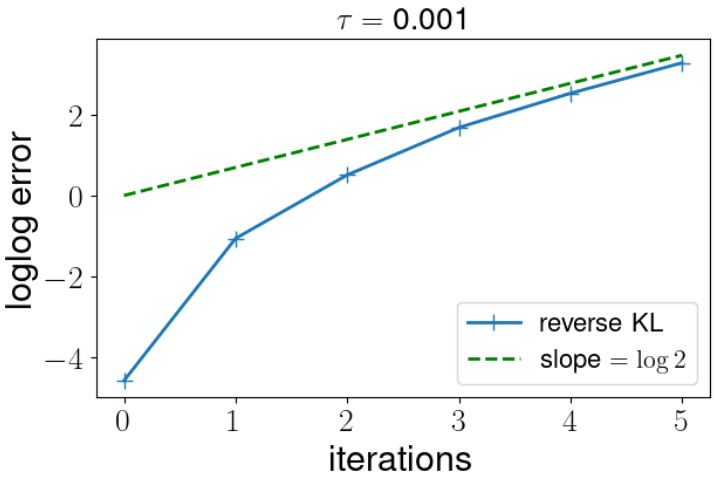
Example 1



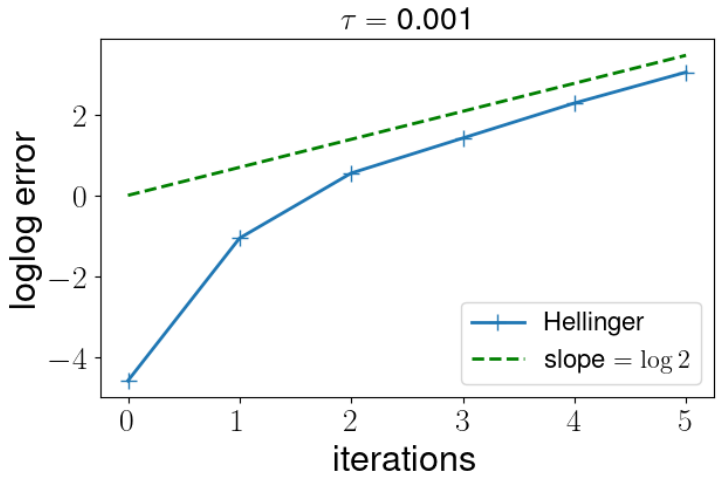
Relative policy error



loglog KL



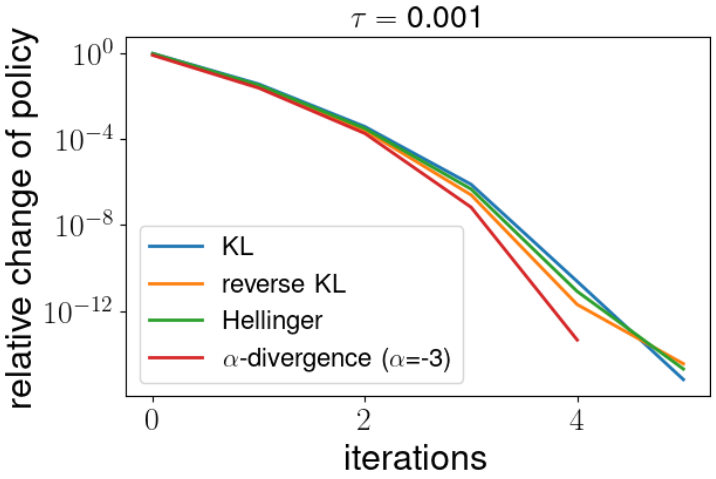
loglog rKL



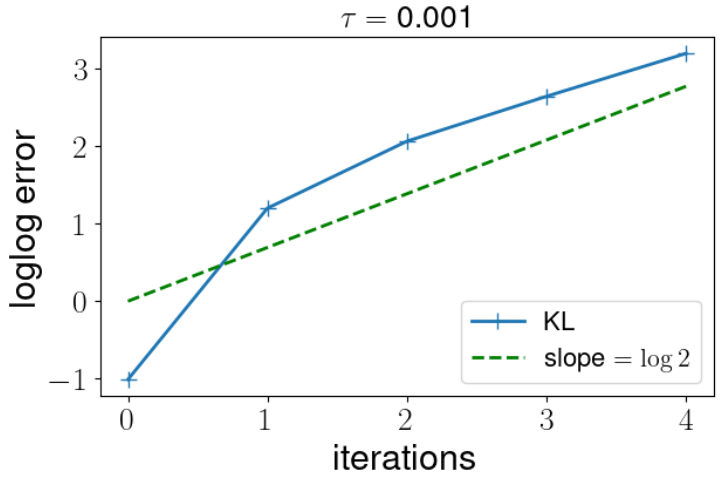
loglog Hellinger

Green line slope $= \log 2$ (Netwon quadratic convergence)

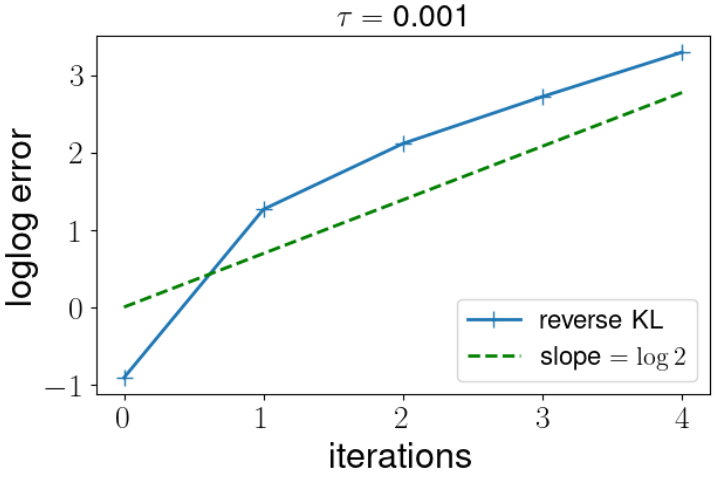
Example 2



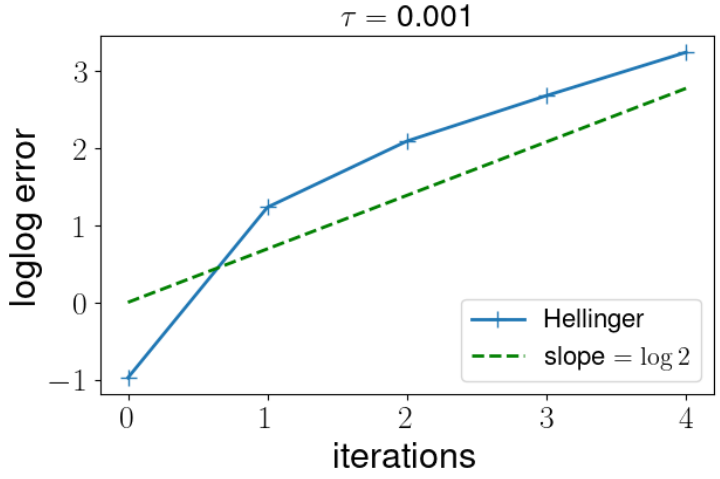
Relative policy error



loglog KL



loglog rKL



loglog Hellinger

Green line slope $= \log 2$ (Newton quadratic convergence)

New algorithm 2

$$\min_v \max_u \sum_{s\in S} e_s v_s + \sum_{s\in S, a\in A} u_s^a (r_s^a - ((I-\gamma P^a)v)_s) - \tau \sum_{s\in S, a\in A} u_s^a \log(u_s^a/\tilde{u}_s)$$

where $\tilde{u}_s := \sum_{a\in A} u_s^a$. By introducing $K^a = I - \gamma P^a$, in short

$$\min_v \max_u \sum_s e_s v_s + \sum_s^a u_s^a (r_s^a - (K^a v)_s) - \tau \sum_{sa} u_s^a \log(u_s^a/\tilde{u}_s).$$

Consider 1st order methods

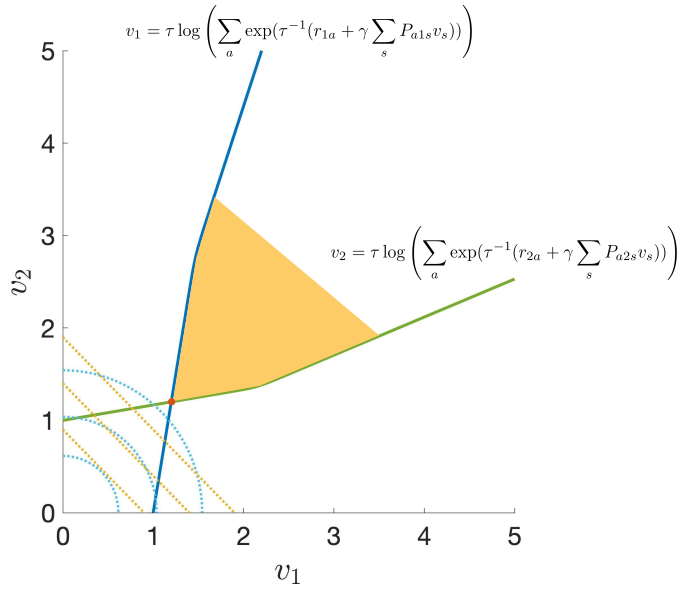
- Simple gradient descent (GD) in v and gradient ascent (GA) in u fail to converge due to lack of strict convexity/concavity
- Example: $\min_v \max_u v \cdot u \Rightarrow \begin{cases} \frac{\mathrm{d}v}{\mathrm{d}t} = -u \\ \frac{\mathrm{d}u}{\mathrm{d}t} = v \end{cases}$
- Improved versions gives $1/T^c$ convergence

Our contribution

- Quadratically convexified in v
- Improved preconditioning in u
- Fast exponential convergence

Quadratic convexification

Main idea in the primal form



For primal-dual, the following are equivalent

$$\min_v \max_u E_0(v,u) = \sum_s e_s v_s + \sum_{sa} u_s^a \left(r_s^a - \sum_{s'} K_{ss'}^a v_{s'} \right) - \tau \sum_{sa} u_s^a \log \frac{u_s^a}{\tilde{u}_s}$$

$$\min_v \max_u E(v,u) = \frac{\alpha}{2} \sum_s v_s^2 + \sum_{sa} u_s^a \left(r_s^a - \sum_{s'} K_{ss'}^a v_{s'} \right) - \tau \sum_{sa} u_s^a \log \frac{u_s^a}{\tilde{u}_s}$$

$E(v,u)$ is quadratically convexified in v

Natural GAD

For $E(v,u)$, apply natural GAD

$$\frac{\partial E}{\partial v_{s'}} = \alpha v_{s'} - \sum_{sa} K^a_{ss'} u^a_s,$$

$$\frac{\partial E}{\partial u^a_s} = \left(r^a_s - \sum_{s'} K^a_{ss'} v_{s'} \right) - \tau \log \frac{u^a_s}{\tilde{u}_s}$$

$$\frac{\partial^2 E}{\partial v_s \partial v_{s'}} = \alpha \delta_{ss'},$$

$$\frac{\partial^2 E}{\partial u^a_s \partial u^{a'}_{s'}} = -\tau \delta_{ss'} \left(\frac{\delta^{aa'}}{u^a_s} - \frac{1}{\tilde{u}_s} \right) \approx -\tau \delta_{ss'} \frac{\delta^{aa'}}{u^a_s}$$

Natural GAD (NGAD)

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\left(v_{s'} - \frac{1}{\alpha} \sum_{sa} K^a_{ss'} u^a_s \right)$$

$$\frac{\mathrm{d}u^a_s}{\mathrm{d}t} = -u^a_s \left(\log \frac{u^a_s}{\tilde{u}_s} - \frac{1}{\tau} \left(r^a_s - \sum_{s'} K^a_{ss'} v_{s'} \right) \right)$$

Convergence

Let (v^*, u^*) be the exact solution.

Theorem

$$L(v,u) = \frac{\alpha}{2}\sum_{s\in S}|v_s - v_s^*|^2 + \tau \sum_{s\in S, a\in A}\left(u_s^{*a}\log\frac{u_s^{*a}}{u_s^a} + u_s^a - u_s^{*a}\right)$$

is a Lyapunov function of the NGAD dynamics

Theorem

The NGAD dynamics converges globally to (v^*, u^*).

Theorem

The NGAD dynamics converges to (v^*, u^*) *at rate* $O(e^{-ct})$.

Q: In practice, the convergence is still slow. Can this be improved?

Another look at the Hessian

$$\frac{\partial^2 E}{\partial u^2} = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_{|S|} \end{bmatrix}, \quad H_s = \operatorname{diag}\left((u_{s\cdot})^{-1}\right) - \frac{1}{\tilde{u}_s}\mathbf{1}_{|A|}\mathbf{1}_{|A|}^{\mathsf{T}}, \quad s \in S.$$

Natural grad: $\frac{\partial^2 E}{\partial u^2} \approx \begin{bmatrix} \operatorname{diag}\left((u_{1\cdot})^{-1}\right) & & \\ & \ddots & \\ & & \operatorname{diag}\left((u_{|S|\cdot})^{-1}\right) \end{bmatrix}$ with inverse

$$\begin{bmatrix} \operatorname{diag}(u_{1\cdot}) & & \\ & \ddots & \\ & & \operatorname{diag}(u_{|S|\cdot}) \end{bmatrix} \equiv \begin{bmatrix} \tilde{u}_1(\operatorname{diag}(\pi_{1\cdot})) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\operatorname{diag}(\pi_{|S|\cdot})) \end{bmatrix}$$

But the pseudo-inverse of $\frac{\partial^2 E}{\partial u^2}$ is

$$\begin{bmatrix} \tilde{u}_1(\operatorname{diag}(\pi_{1\cdot}) - \pi_{1\cdot}\pi_{1\cdot}^{\mathsf{T}}) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\operatorname{diag}(\pi_{|S|\cdot}) - \pi_{|S|\cdot}\pi_{|S|\cdot}^{\mathsf{T}}). \end{bmatrix}$$

Interpolate

An interpolation of the two gives

$$\begin{bmatrix} \tilde{u}_1(\mathrm{diag}(\pi_{1\cdot}) - c\pi_{1\cdot}\pi_{1\cdot}^{\mathsf{T}}) & & \\ & \ddots & \\ & & \tilde{u}_{|S|}(\mathrm{diag}(\pi_{|S|\cdot}) - c\pi_{|S|\cdot}\pi_{|S|\cdot}^{\mathsf{T}}) \end{bmatrix}$$

where $c \in (0, 1)$

This gives interpolated NGAD (I-NGAD)

$$\frac{\mathrm{d}v_{s'}}{\mathrm{d}t} = -\left(v_{s'} - \frac{1}{\alpha}\sum_{sa} K^a_{ss'} u^a_s\right)$$

$$\frac{\mathrm{d}\dot{u_s}}{\mathrm{d}t} = -\tilde{u}_s\left(\mathrm{diag}(\dot{\pi_s}) - c\dot{\pi_s}(\dot{\pi_s})^{\mathsf{T}}\right)\left(\log\frac{\dot{u_s}}{\tilde{u}_s} - \frac{1}{\tau}\left(\dot{r_s} - \sum_{s'}\dot{K_{ss'}}v_{s'}\right)\right)$$

Convergence

Let (v^*, u^*) be the exact solution.

Theorem

$$L_c(v,u) = \frac{\alpha}{2}\sum_s |v_s - v_s^*|^2 + \tau\left(\sum_{sa}\left(u_s^{*a}\log\frac{u_s^{*a}}{u_s^a} + u_s^a - u_s^{*a}\right) + \frac{c}{1-c}\sum_s\left(\tilde{u}_s^*\log\frac{\tilde{u}_s^*}{\tilde{u}_s} + \tilde{u}_s - \tilde{u}_s^*\right)\right)$$

is a Lyapunov function of the I-NGAD dynamics

Theorem

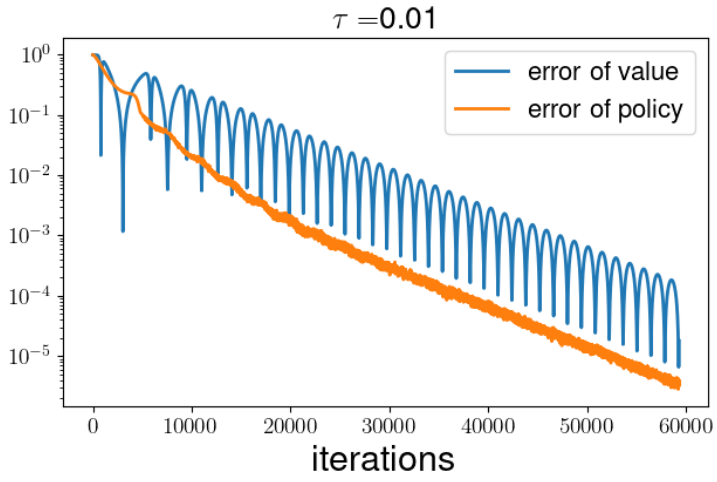
The I-NGAD dynamics converges globally to (v^*, u^*).

Theorem

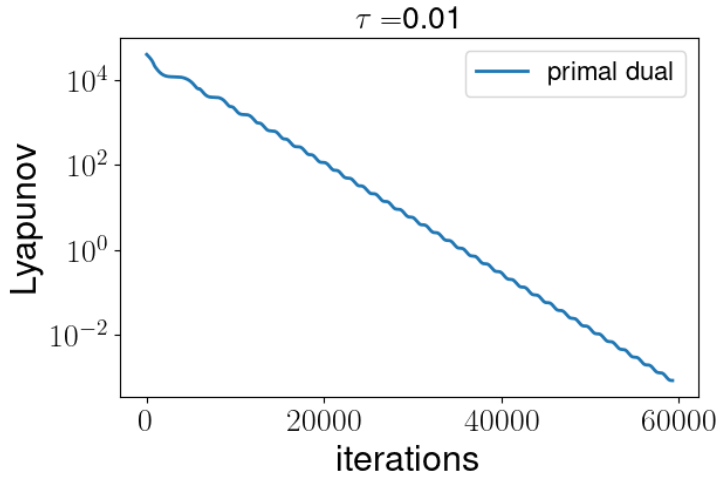
The I-NGAD dynamics converges to (v^*, u^*) *at rate* $O(e^{-ct})$.

In practice, the convergence is FAST

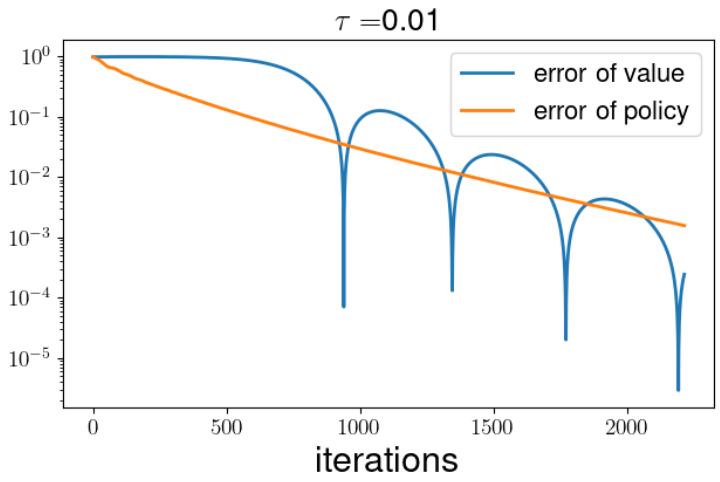
Example 1 (no noise)



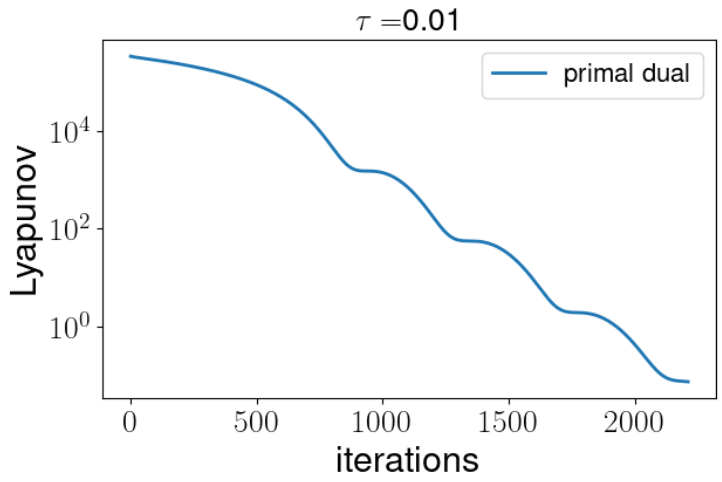
NGAD: Errors



NGAD: Lyapunov function



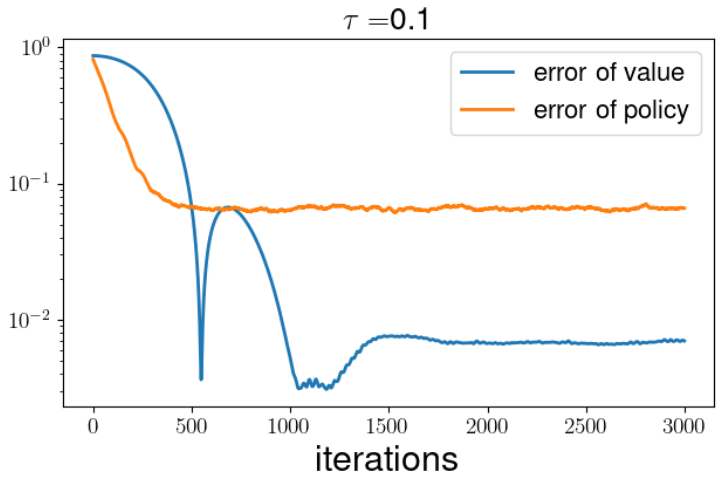
I-NGAD: Errors



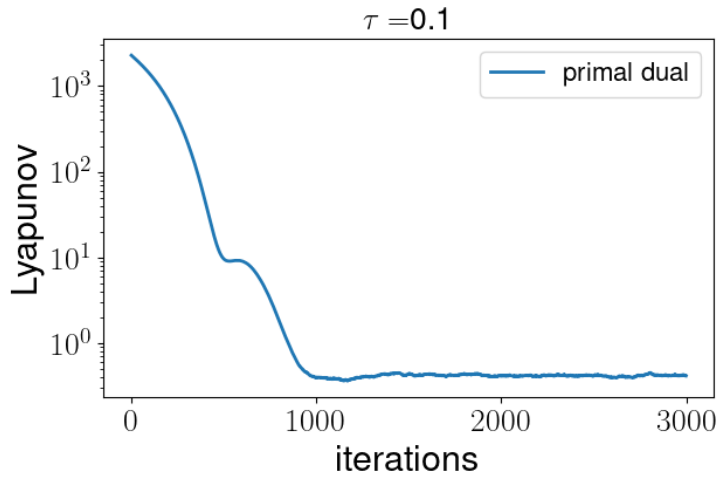
I-NGAD: Lyapunov function

Example 2 (Gaussian noise)

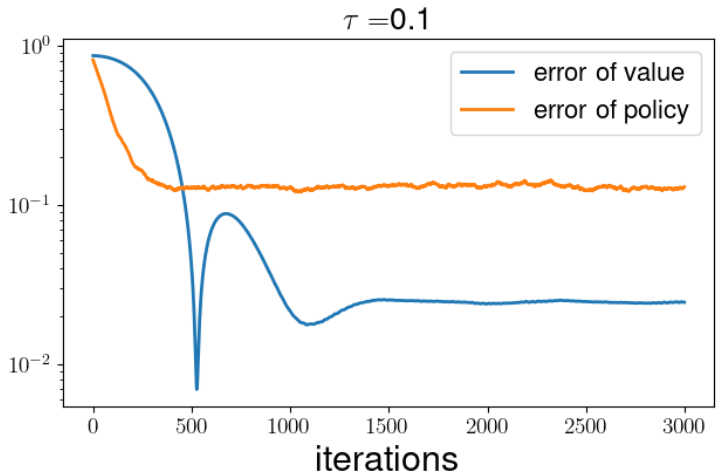
NGAD converges slowly. I-NGAD works well (σ noise level)



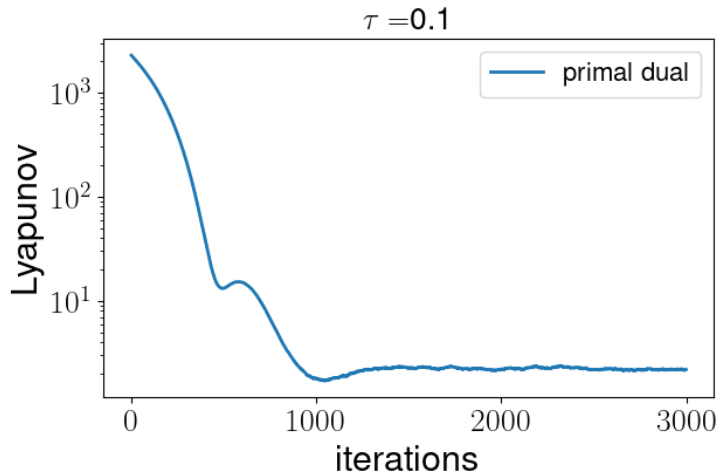
I-NGAD: Errors ($\sigma = 0.1$)



I-NGAD: Lyapunov function ($\sigma = 0.1$)



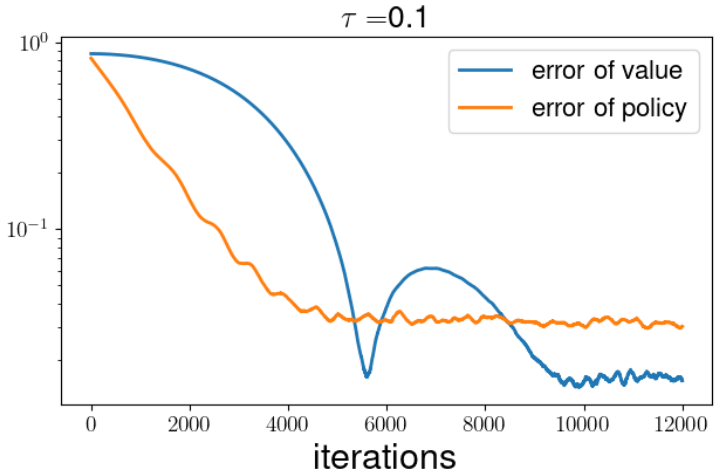
I-NGAD: Errors ($\sigma = 0.2$)



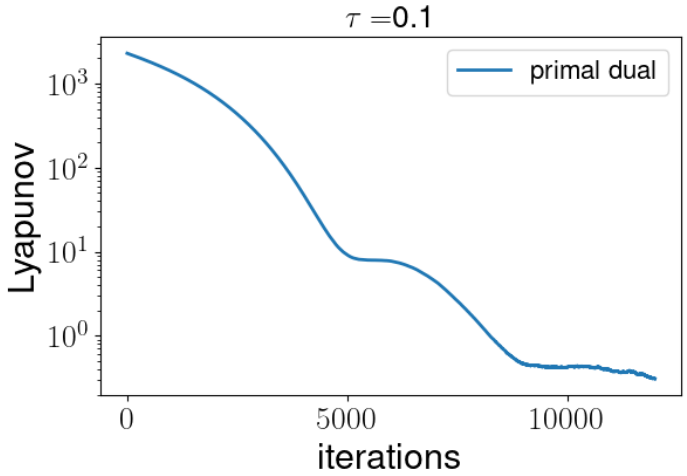
I-NGAD: Lyapunov function ($\sigma = 0.2$)

Example 3 (Sampling noise)

NGAD fails to converge. I-NGAD works well



I-NGAD: Errors



I-NGAD: Lyapunov function

Summary

Linear/convex programming formulations for MDP

- Non-regularized and entropy-regularized
- Primal form, Bellman equation, value (policy) iteration
- Primal-dual form (actor-critic algorithms)
- Dual, policy gradient

New algorithms

- Quasi-Newton policy gradient algorithm, dual form, super exponential convergence
- Interpolated natural gradient ascent descent (I-NGAD), primal-dual form, exponential convergence

Thank you

References

- Lexing Ying, Yuhua Zhu, A Note on Optimization Formulations of Markov Decision Processes. To appear in Communications in Mathematical Sciences.
- Haoya Li, Samarth Gupta, Hsiangfu Yu, Lexing Ying, Inderjit Dhillon. Quasi-Newton policy gradient algorithms. arXiv:2110.0239
- Haoya Li, Hsiangfu Yu, Lexing Ying, Inderjit Dhillon. Accelerating Primal-dual Methods for Regularized Markov Decision Processes.