Structured Policy Iteration for Linear Quadratic Regulator

Youngsuk Park[1] with R. Rossi[2], Z. Wen[3], G. Wu[2], and H. Zhao[2]

Stanford University[1], Adobe research[2], Deepmind[3]

July 14, 2020

Introduction

- reinforcement learning (RL) is about learning from interaction with delayed feedback
  - decide action to take, which affects the next state of environment
  - need sequential decision making
- most of discrete RL algorithms scales poorly for tasks in continuous space
  - discretize state or/and action space
  - curse of dimensionality
  - sample inefficiency

Linear Quadratic Regulator

- Linear Quadratic Regulator (LQR) has rich applications for continuous space task
  - e.g., motion planning, trajectory optimization, portfolio
- Infinite horizon (undiscounted) LQR problem

$$\begin{aligned} \underset{\pi}{\text{minimize}} \quad & \mathbb{E}\left(\sum_{t=0}^{\infty} x_t^T Q x_t + u_t^T R u_t\right) \\ \text{subject to} \quad & x_{t+1} = Ax_t + Bu_t, \\ & u_t = \pi(x_t),\ x_0 \sim \mathcal{D}, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, Q \succeq 0$, and $R \succ 0$.

  - quadratic cost Q, R and linear dynamics A, B
  - Q, R set relative weights of state deviation and input usage

Linear Quadratic Regulator (Continued)

- LQR problem

$$\begin{array}{ll} \underset{\pi}{\text{minimize}} & \mathbb{E}\left(\sum_{t=0}^{\infty} x_t^T Q x_t + u_t^T R u_t\right) \\ \text{subject to} & x_{t+1} = Ax_t + Bu_t, \\ & u_t = \pi(x_t),\ x_0 \sim \mathcal{D}, \end{array}$$

  where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, Q \succeq 0$, and $R \succ 0$.

- well-known facts
  - **linear** optimal policy (or control gain) $\pi^\star(x) = Kx$,
  - **quadratic** optimal value function (cost-to-go) $V^\star(x) = x^T P x$ s.t.

$$P = A^T PA + Q - A^T PB(B^T PB + R)^{-1} B^T PA,$$

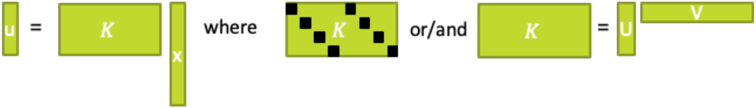
$$K = -(B^T PB + R)^{-1} B^T PA.$$

  - P can be derived efficiently, e.g., Riccati recursion, SDP, etc

- many variants and extensions
  - e.g., time-varying, averaged or discounted, jumping LQR etc.

Structured Linear Policy

- can we find the **structured linear policy** for LQR?
- **structure** can mean (block) sparse, low-rank, *etc*



  - *more interpretable, memory and computationally efficient, well-suited for distributed setting*
  - Often, structure policy is related to physical decision system
    - e.g., data cooling system need to install/arrange cooling infrastructure
- To tackle this, we develop
  - formulation, algorithm, theory and practice

Formulation

- regularized LQR problem

$$\begin{array}{ll} \underset{K}{\text{minimize}} & \overbrace{\mathbb{E}\left(\sum_{t=0}^{\infty} x_t^T Q x_t + u_t^T R u_t\right)}^{f(K)} + \lambda r(K) \\ \text{subject to} & x_{t+1} = A x_t + B u_t, \\ & u_t = K x_t, \quad x_0 \sim \mathcal{D}, \end{array} \tag{2}$$

  - explicitly restrict policy to linear class, i.e., $u_t = K x_t$
  - value function is still quadratic, i.e., $V(x) = x^T P x$ for some P
  - convex regularizer with (scalar) parameter $\lambda \geq 0$

- regularizer $r(K)$ induces the policy structure
  - lasso $\|K\|_1 = \sum_{i,j} |K_{i,j}|$ for sparse structure
  - group lasso $\|K\|_{\mathcal{G},2} = \sum_{g \in \mathcal{G}} \|K_g\|_2$ for block-diagonal structure
  - nuclear-norm $\|K\|_* = \sum_i \sigma_i(K)$ for low-rank structure
  - proximity $\|K - K^{\text{ref}}\|_F^2$ for some $K^{\text{ref}} \in \mathbb{R}^{n \times m}$,

Structured Policy Iteration (S-PI)

- When model is known, S-PI repeats
  - (1) Policy (and covariance) evaluation
    - solve Lyapunov equations to return (P^i, Σ^i)

$$
\begin{aligned}
(A + BK^i)^T P^i (A + BK^i) - P^i + Q + (K^i)^T R K^i = 0, \\
(A + BK^i) \Sigma^i (A + BK^i)^T - \Sigma^i + \Sigma_0 = 0.
\end{aligned}
$$

  - (2) Policy improvement
    - compute gradient $\nabla_K f(K^i) = 2\left(\left(R + B^T P^i B\right) K^i + B^T P^i A\right) \Sigma^i$
    - apply proximal gradient step under linesearch

- note that
  - Lyapunov equation requires $O(n^3)$ to solve
  - (almost) no hyperparameter to tune under linesearch (LS),
  - LS make stability $\rho(A + BK^i) < 1$ satisfied

Convergence

Theorem (Park et al. '20) Assume K^0 s.t. $\rho(A + BK^0) < 1$. K^i from S-PI Algorithm converges to the stationary point $K^\star$. Moreover, it converges linearly, i.e., after N iterations,

$$\left\| K^N - K^\star \right\|_F^2 \leq \left(1 - \frac{1}{\kappa}\right)^N \left\| K^0 - K^\star \right\|_F^2 .$$

Here, $\kappa = 1/\left(\eta_{\min}\sigma_{\min}(\Sigma_0)\sigma_{\min}(R)\right) > 1$ where

$$\begin{aligned} \eta_{\min} = h_\eta \Big(& \sigma_{\min}(\Sigma_0), \sigma_{\min}(Q), \frac{1}{\lambda}, \\ & \frac{1}{\|A\|}, \frac{1}{\|B\|}, \frac{1}{\|R\|}, \frac{1}{\Delta}, \frac{1}{F(K^0)} \Big), \end{aligned} \tag{3}$$

for some non-decreasing function h_η on each argument.

- Riccati recursion can give stabilizing initial policy K^0
- (global bound on) fixed stepsize $\eta_{\min}$ depends on model parameters
- note $\eta_{\min} \propto 1/\lambda$
- in practice using LS, stepsize does have to be tuned or calculated

Model-free Structured Policy Iteration

- when model is unknown, S-PI repeats
  - (1) Perturbed policy evaluation
    - get perturbation and (perturbed) cost-to-go $\{\hat{f}^j, U^j\}_{j=1}^{N_{\text{traj}}}$
      for each $j = 1, \ldots, N_{\text{traj}}$
      sample $U^j \sim \text{Uniform}(\mathbb{S}_r)$ to get a perturbed $\hat{K}^i = K^i + U^j$
      roll out $\hat{K}^i$ over the horizon H to estimate the cost-to-go

$$\hat{f}^j = \sum_{t=0}^{H} g(x_t, \hat{K}^i x_t)$$

  - (2) Policy improvement
    - compute the (noisy) gradient

$$\widehat{\nabla_K f(K^i)} = \frac{1}{N_{\text{traj}}} \sum_{j=1}^{N_{\text{traj}}} \frac{n}{r^2} \hat{f}^j U^j$$

    - apply proximal gradient step

- note that
  - smoothing procedure adapted to estimate noisy gradient
  - (N_{traj}, H, r) are additional hyperparameters to tune
  - LS is not applicable

Convergence

Theorem (Park et al. '20)
Suppose $F(K^0)$ is finite, $\Sigma_0 \succ 0$, and that $x_0 \sim \mathcal{D}$ has norm bounded by D almost surly. Suppose the parameters in Algorithm **??** are chosen from

$$(N_{\text{traj}}, H, 1/r) = h\left(n, \frac{1}{\epsilon}, \frac{1}{\sigma_{\min}(\Sigma_0)\sigma_{\min}(R)}, \frac{D^2}{\sigma_{\min}(\Sigma_0)}\right).$$

for some polynomials h. Then, with the same stepsize in Eq. (3), there exist iteration N at most $4\kappa \log\left(\frac{\|K^0 - K^\star\|_F}{\epsilon}\right)$ such that $\left\|K^N - K^\star\right\| \leq \epsilon$ with at least $1 - o(\epsilon^{n-1})$ probability. Moreover, it converges linearly,

$$\left\|K^i - K^\star\right\|^2 \leq \left(1 - \frac{1}{2\kappa}\right)^i \left\|K^0 - K^\star\right\|^2,$$

for the iteration $i = 1, \ldots, N$, where $\kappa = \eta\sigma_{\min}(\Sigma_0)\sigma_{\min}(R) > 1$.

- Assume K^0 is stabilizing policy but cannot use Riccati here
- here (N_{traj}, H, r) are hyperparameters to tune

Experiment (Setting)

- Consider unstable Laplacian system $A \in \mathbb{R}^{n\times n}$ where

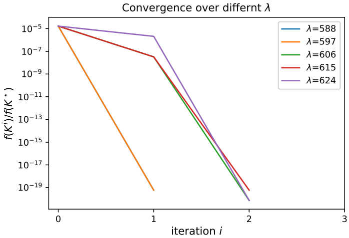
$$A_{ij} = \begin{cases} 1.1, & i = j \\ 0.1, & i = j+1 \text{ or } j = i+1 \\ 0, & \text{otherwise} \end{cases}$$

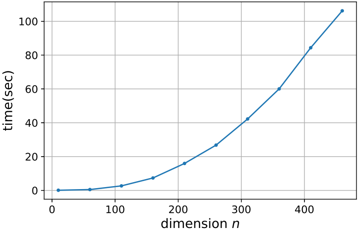
  $B = Q = I_n \in \mathbb{R}^{n\times n}$ and $R = 1000 \times I_n \in \mathbb{R}^{n\times n}$.
  - unstable open loop system, i.e., $\rho(A) \geq 1$
  - extremely sensitive to parameters (even under known model setting)
  - less in favor of the generic model-free RL approaches to deploy

- Model and S-PI algorithm parameter under known model
  - system size $n \in [3, 500]$
  - lasso penalty with $\lambda \in [10^{-2}, 10^{6}]$
  - LS with initial stepsize $\eta = \frac{1}{\lambda}$ with backtracking factor $\beta = \frac{1}{2}$
  - For fixed stepsize, select $\eta = \mathcal{O}\left(\frac{1}{\lambda}\right)$

Experiment (Continued)

- **Convergence behavior under LS and scalability**

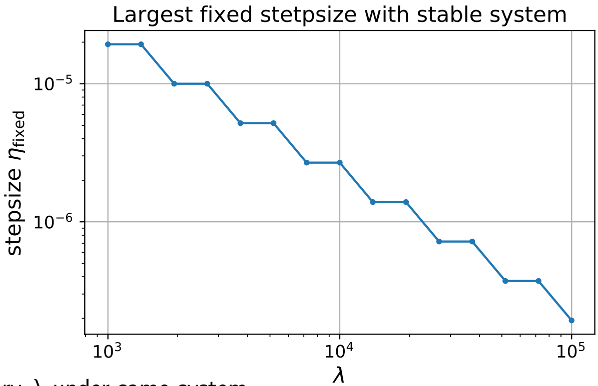




  - S-PI with LS converges very fast over various n and λ
  - scales well for large system, even with computational bottleneck on solving Lyapunov equation
  - For $n = 500$, takes less than 2 mins (MacBook Air)

Experiment (Continued)

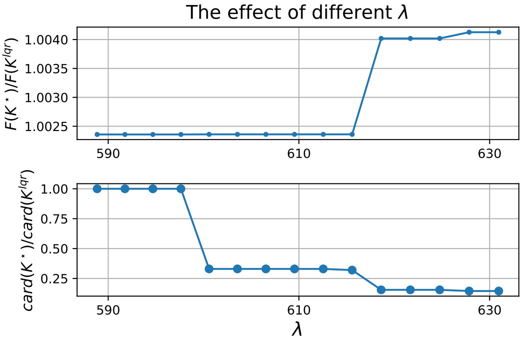
- **Dependency of stepsize η on λ.**



  - vary λ under same system
  - largest (fixed) stepsize for stable (closed) system, i.e., $A + BK^i < 1$ is non-increasing, i.e., $\eta_{\text{fixed}} \propto \frac{1}{\lambda}$

Experiment (Continued)

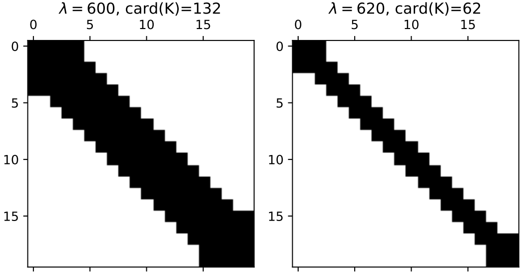
- **Trade off between LQR performance and structure K**



  - LQR solution K^{lqr}, and S-PI solution $K^{\star}$
  - λ increases, LQR cost $f(K^{\star})$ increases whereas cardinality decreases (sparsitiy is improved).
  - In this range, S-PI barely changes LQR cost but improved the sparsity more than 50%.

Experiment (Continued)

- **sparsity pattern of policy matrix**



  – sparsity pattern (location of non-zero elements) of the policy matrix with $\lambda = 600$ and $\lambda = 620$.

Challenge on model-free approach

- model-free approach is challenging and unstable
  - especially unstable open loop system $\rho(A) < 1$
  - suffer similar difficulty to the model-free policy gradient method [Fazel et al., 2018] for LQR
  - finding stabilizing initial policy K^0 is non-trivial unless $\rho(A) < 1$
  - suffer high variance, especially sensitive to smoothing parameter r

- open problems and algorithmic efforts needed in practice
  - variance reduction
  - rule of thumb to tune hyperparamters

- still, promising as a different class of model-free approach
  - no discretization
  - no need to compute $Q(s, a)$ pair (like in REINFORCE)
  - seems to work for averaged cost of LQR (easier class of LQR)
  - more in longer version of paper

Summary

- formulate regularized LQR problem to derive structured policy
- develop S-PI algorithm for both model-based and model-free approach with theoretical guarantees
- model-based S-PI works well in practice with (almost) no hyperparameter tuning
- model-free S-PI is still promising but challenging

Thank you!