Renewal theory based Reinforcement Learning for Markov processes with controlled restarts

Jayakumar Subramanian

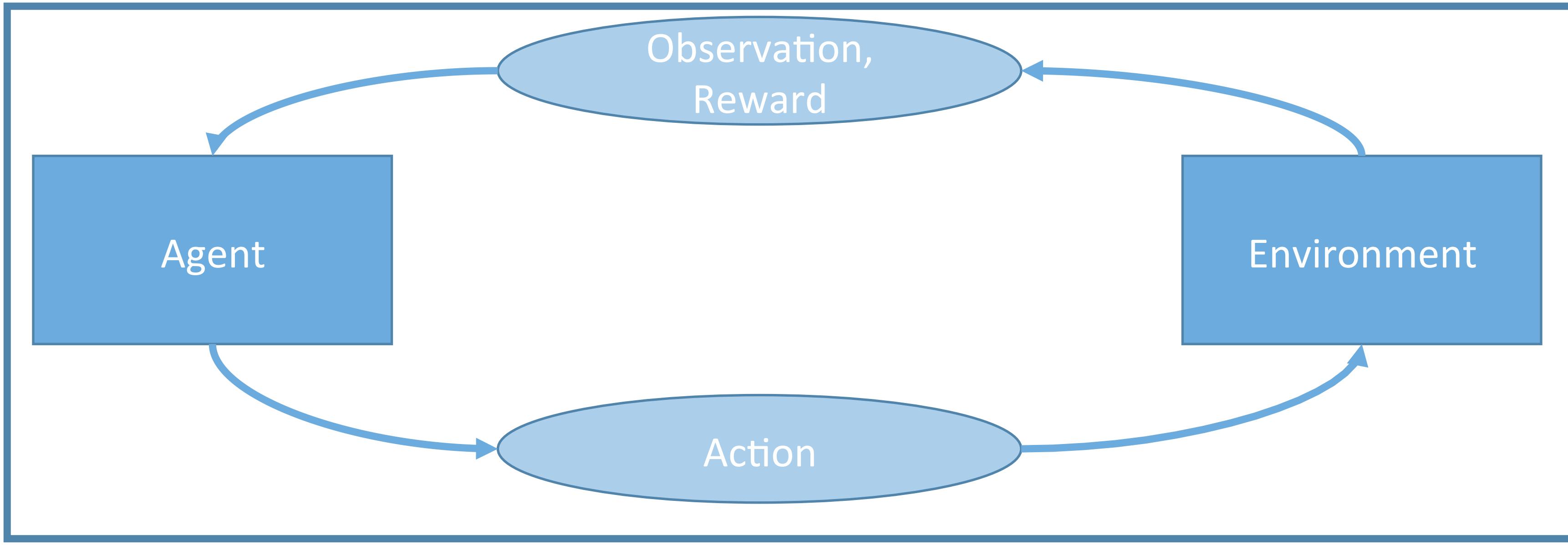
Joint work with Jhelum Chakravorty & Aditya Mahajan

McGill University

JOPT2017, May 10, 2017

Reinforcement Learning

Reinforcement Learning



Some Application Examples

- Self-driving cars
- Smart home applications – NEST
- Game playing – AlphaGO
- Conversational Agents / Chatbots
- Recommender Systems
- Portfolio Management

Types of RL

Markov Decision Processes - MDP

{ State Space | Action Space | Transition Kernel | Reward Fn }

Model-Based RL

- Transitions and rewards are learnt from sample trajectories
- Planning - Standard DP or Approximate DP used to find optimal policy

Model-Free RL

- Value or Action Value function or performance estimated directly without estimating T and R
- Optimal policy then determined using these estimates

Our Focus

Model Free RL

Critic Only

- Only (Action) Value Function (Q or V) modelled and estimated
- Implicit Policy function – Greedy in Q Values

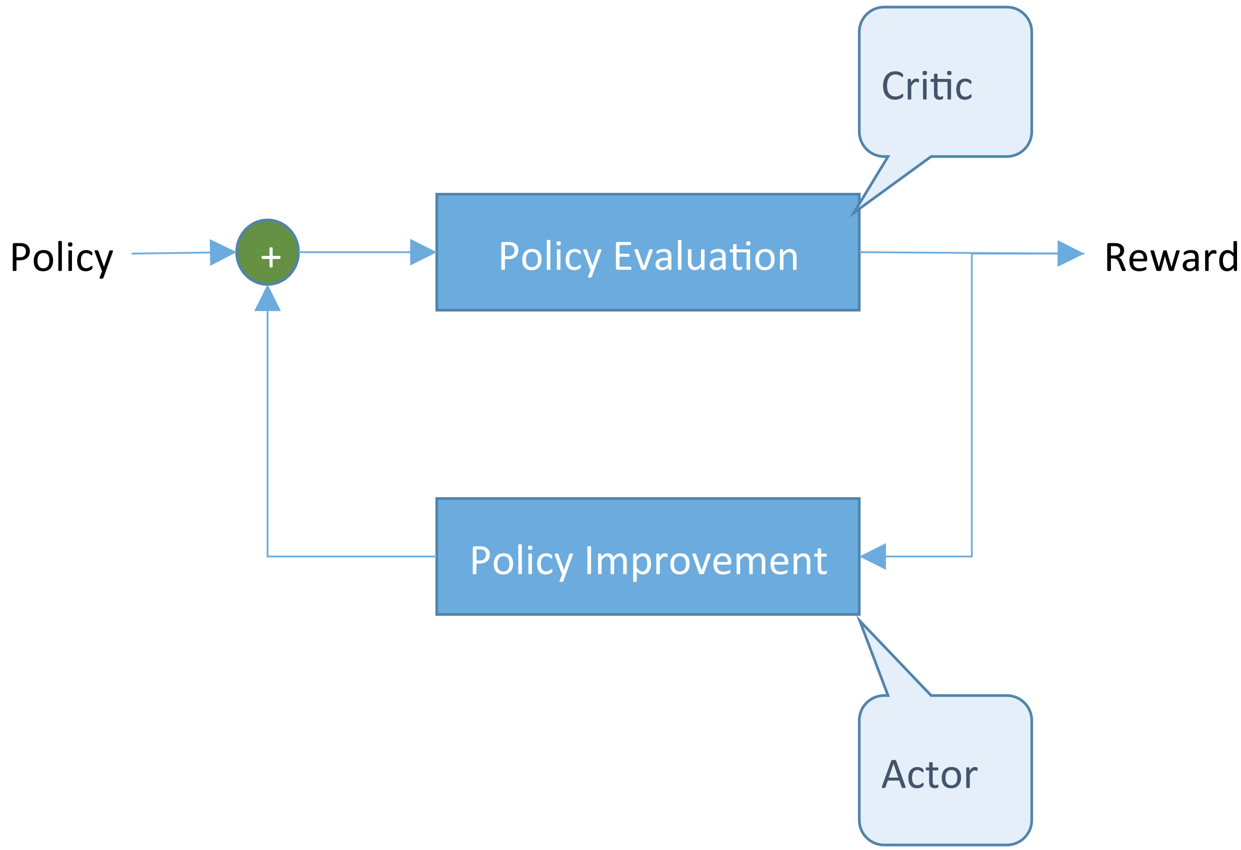
Actor Only

- Only Policy Function (Q or V) modelled – typically a parametrized policy
- Performance is estimated (Implicit Critic)

Actor Critic

- Both (action) value function and policy function modelled and estimated
- Attempts to combine best of both worlds

Basic Algorithm



Policy Evaluation

MC	Minimum Mean Squared Error Estimate	Relative Convergence Speed – Open Question (& problem dependent)
TD	Maximum Likelihood Estimate	

TD (λ) – in between TD and MC methods, $\lambda \in [0,1]$

Methods	Concern
Monte-Carlo (MC) methods	High variance estimates
Temporal Difference (TD) methods	High bias estimates

Policy Improvement

Key Steps

- Policy Parametrized by parameters: θ
- Performance: J_θ
- Policy Improvement: Gradient Ascent: $\theta_{n+1} = \theta_n + \alpha_n \nabla_\theta J_\theta$

Actor Only

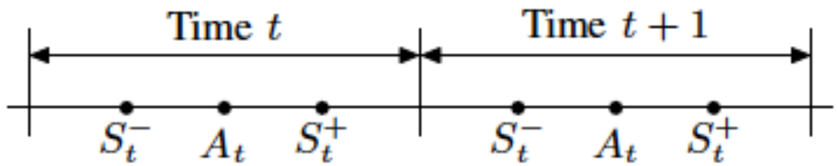
- Stochastic Finite Difference : $\nabla J_\theta \approx \frac{1}{\delta}[J_{\theta+\delta} - J_\theta]$
  - Kiefer-Wolfowitz
  - SPSA
  - SF
- Quotient Rule: $\theta_{n+1} = \theta_n + \alpha_n [T_\theta(s_*^+) R_{\theta+\delta}(s_*^+) - R_\theta(s_*^+) T_{\theta+\delta}(s_*^+)]$

Actor Critic

- Policy Gradient Theorem : $$\nabla J_\theta = \sum_{t=1}^{\tau(s_*^+)} f_\theta^{score}(S_t^-, A_t) V_\pi^+(S_t^-)$$
- Score function: $f_\theta^{score}(s^-, a) = \nabla_\theta \log[\pi_\theta(s^-, a)]$

Model

- Pre-decision State, Action, Post-decision State:



- Controlled transition from Pre-decision state to Post-decision state:

$$\begin{aligned}
&\mathbb{P}(S_t^+ = s_t^+ | S_{1:t}^- = s_{1:t}^-, A_{1:t} = a_{1:t}, S_{1:t-1}^+ = s_{1:t-1}^+) \\
&\quad = \mathbb{P}(S_t^+ = s_t^+ | S_t^- = s_t^-, A_t = a_t) \\
&\quad =: P^+(s_t^+ | s_t^-, a_t)
\end{aligned}$$

- Uncontrolled transition from post-decision state to next pre-decision state:

$$\begin{aligned}
&\mathbb{P}(S_{t+1}^- = s_{t+1}^- | S_{1:t}^- = s_{1:t}^-, A_{1:t} = a_{1:t}, S_{1:t}^+ = s_{1:t}^+) \\
&\quad = \mathbb{P}(S_{t+1}^- = s_{t+1}^- | S_t^+ = s_t^+) \\
&\quad =: P^-(s_{t+1}^- | s_t^+)
\end{aligned}$$

- Per step reward: $r(S_t^-, A_t, S_t^+)$

Value and Action Value Functions

Regenerative MDPs – Post Decision State Variable

- Regenerative in terms of the post-decision state variable
- Regenerative → Process restarts → Same state revisited infinitely often
- Stopping time: Time till restart: $\tau(s^+)$
- Trajectories between stopping times: Regenerative Cycles
- Regenerative cycles are independent of each other
- (Action) Value function estimated in terms of post-decision state variable

$$V_\pi^+(s^+) = \mathbb{E}_{A_t \sim \pi(S_t^-)}\left[\sum^{\infty} \gamma^{t-1} r(S_t^-, A_t, S_t^+) \,\middle|\, S_0^+ = s^+\right]$$

$$V_\pi^+(s^+) = \sum_{s^- \in \mathcal{S}^-} P^-(s^-|s^+) V_\pi^-(s^-)$$

$$V_\pi^-(s^-) = \sum_{a \in \mathcal{A}} \pi(s^-, a) Q_\pi(s^-, a)$$

$$Q_\pi(s^-, a) = \sum_{s+ \in \mathcal{S}+} P^-(s^+|s^-, a)\left[r(s^-, a, s^+) + \gamma V_\pi^+(s^+)\right]$$

Post Decision State: Example

Inventory Control Problem

- Stock Evolution Equation:

$$S^-_{t+1} = f(S^-_t, A_t) - W_t$$

$$S^+_t = f(S^-_t, A_t)$$

- Per step cost:

$$c(S^-_t, A_t) = d(S^-_t, A_t) + p(A_t, \lambda)$$

$$r(S^-_t, A_t) = -c(S^-_t, A_t)$$

- Base-stock strategy (two threshold-based):

$$\theta = [k1, k2]$$

$$A = \mathbb{1}_{S^- > k1}$$

$$f(S^-, A = 0) = S^-$$

$$f(S^-, A = 1) = k2$$

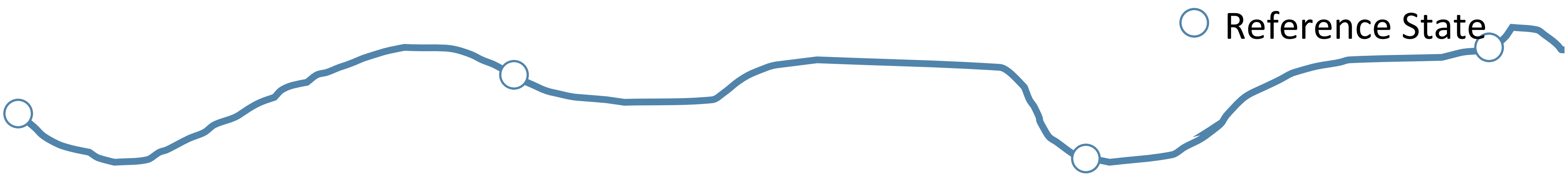
Renewal Relationships

Infinite Horizon Performance – Estimated using (finite horizon) Regenerative Cycle

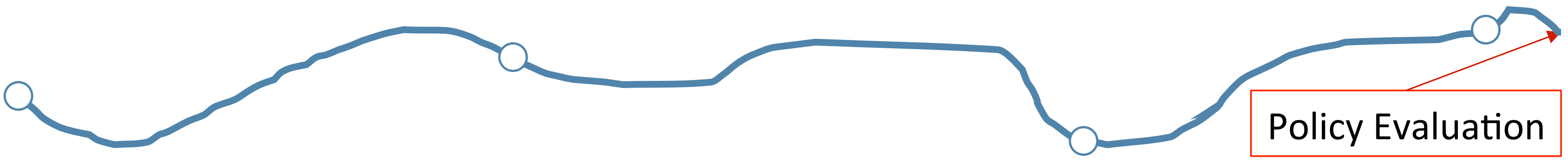
Exp. discounted restart reward	$R_\pi(s^+) = \mathbb{E}_{A_t \sim \pi(S_t^-)}\left[\sum_{t=1}^{\tau(s^+)} \gamma^{t-1} r(S_t^-, A_t, S_t^+) \,\middle|\, S_0^+ = s^+\right]$
Exp. discounted restart time	$T_\pi(s^+) = \mathbb{E}_{A_t \sim \pi(S_t^-)}\left[\sum_{t=1}^{\tau(s^+)} \gamma^{t-1} \,\middle|\, S_0^+ = s^+\right]$
For Reference State	$V_\pi^+(s_*^+) = \dfrac{R_\pi(s_*^+)}{T_\pi(s_*^+)}$
For any other State	$V_\pi^+(s^+) = R_\pi(s^+) + [1 - (1-\gamma)T_\pi(s^+)]V_\pi^+(s_*^+)$

Renewal Theory: Policy Evaluation

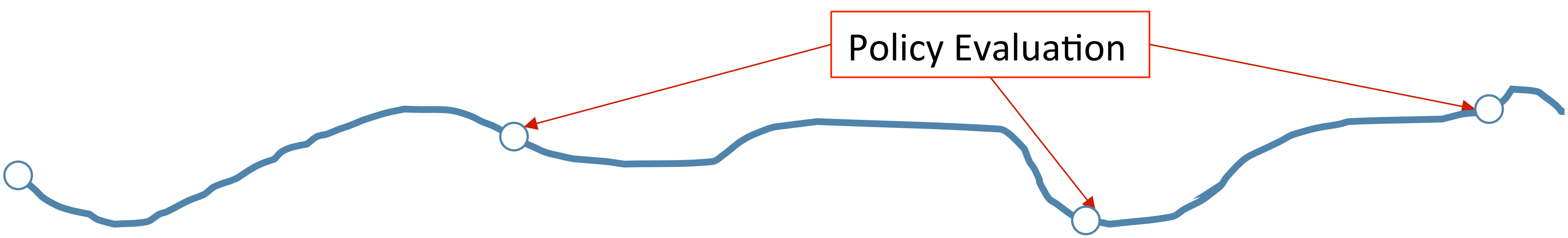
Sample Trajectory



Monte Carlo Evaluation

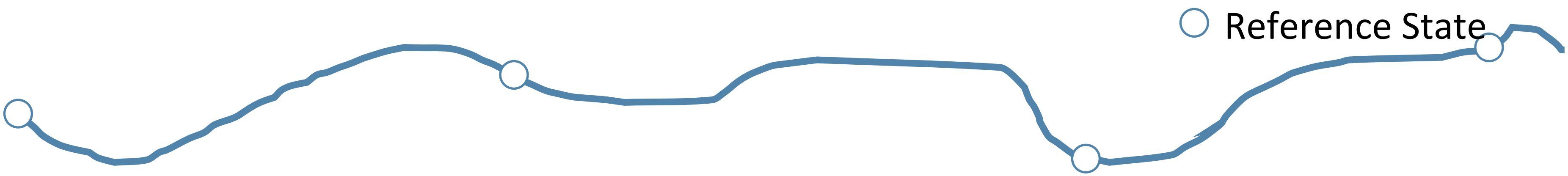


Renewal Monte Carlo Evaluation

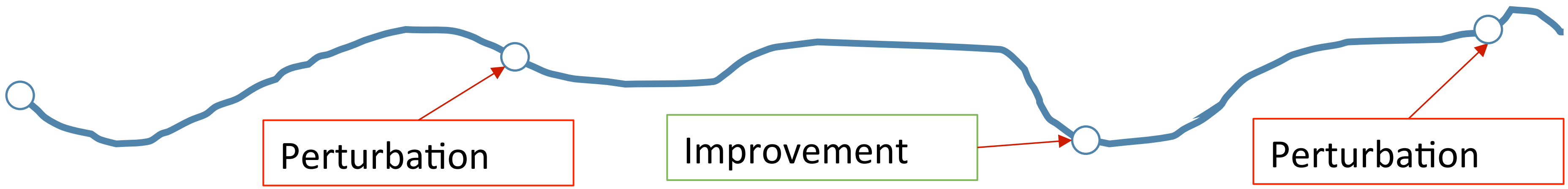


Renewal Theory: Policy Improvement

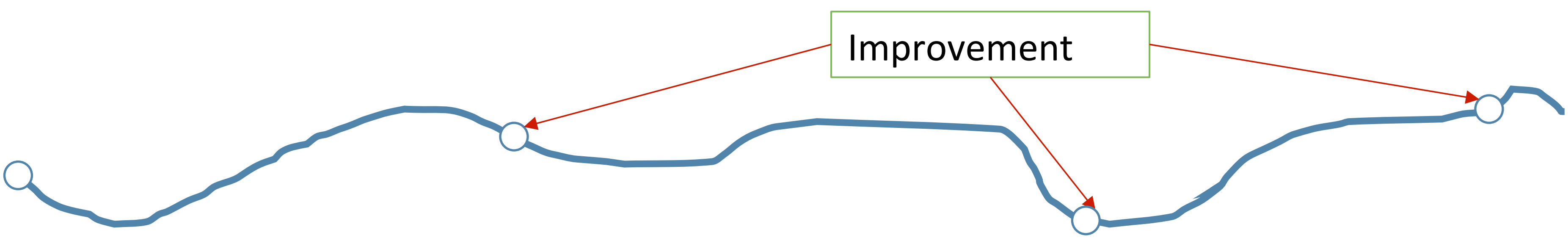
Sample Trajectory



Actor Only Policy Improvement



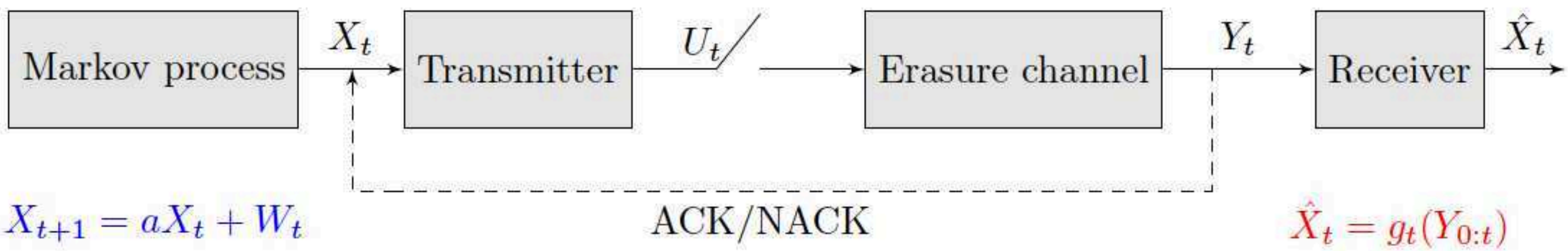
Actor Critic Policy Improvement



Remote Estimation Problem

Model

$$U_t = f_t(X_{0:t}, U_{0:t-1}), \in \{0,1\} \quad H_t \in \{\text{ON}(1\text{-}\varepsilon), \text{OFF}(\varepsilon)\}$$

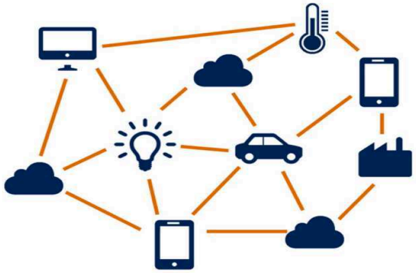


$$X_{t+1} = aX_t + W_t \qquad \hat{X}_t = g_t(Y_{0:t})$$

Applications

Smart Grid

Internet of Things

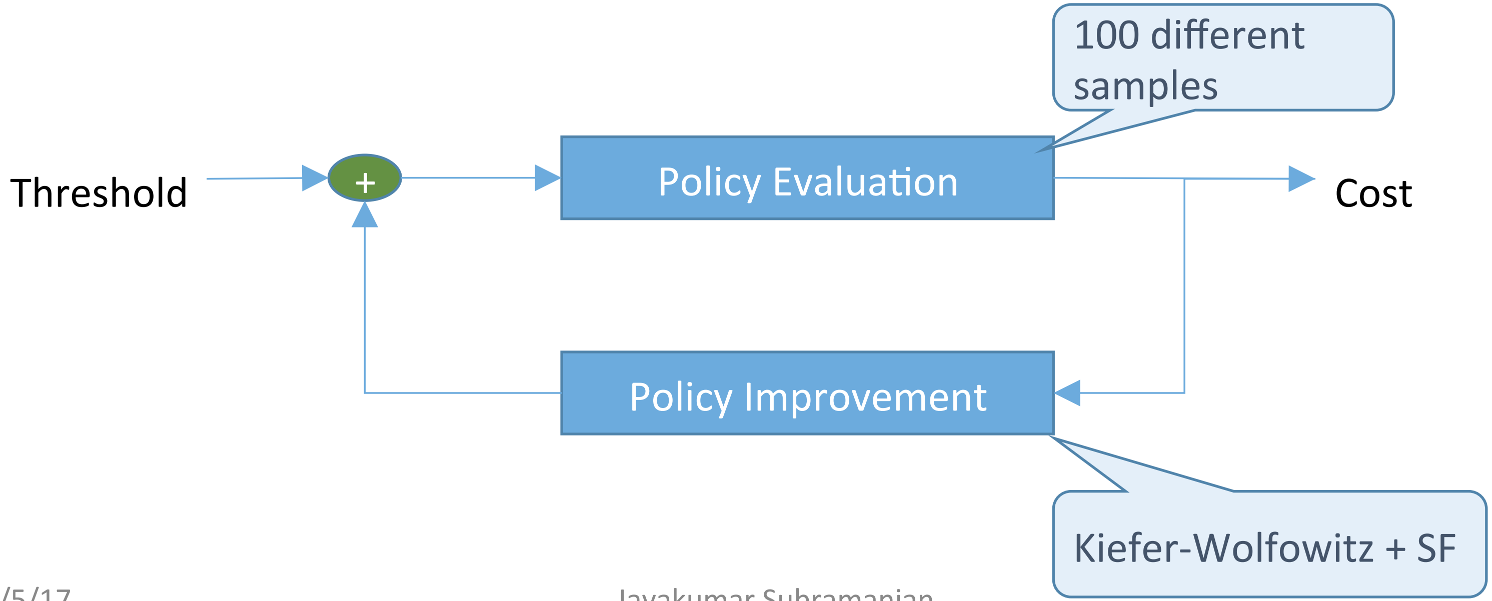


Sensor Network

Algorithm

Optimal Transmitter	$U_t = f_t^*(X_t, U_{0:t-1}) = f^*(X_t) = \begin{cases} 1, & \text{if } \lvert X_t - a\hat{X}_t \rvert \geq k \\ 0, & \text{if } \lvert X_t - a\hat{X}_t \rvert < k \end{cases}$
Optimal Receiver (Estimator)	$\hat{X}_t = g_t^*(Y_t) = g^*(Y_t) = \begin{cases} Y_t, & \text{if } Y_t \neq \mathfrak{E}; \\ a\hat{X}_{t-1}, & \text{if } Y_t = \mathfrak{E}. \end{cases}$

Algorithm

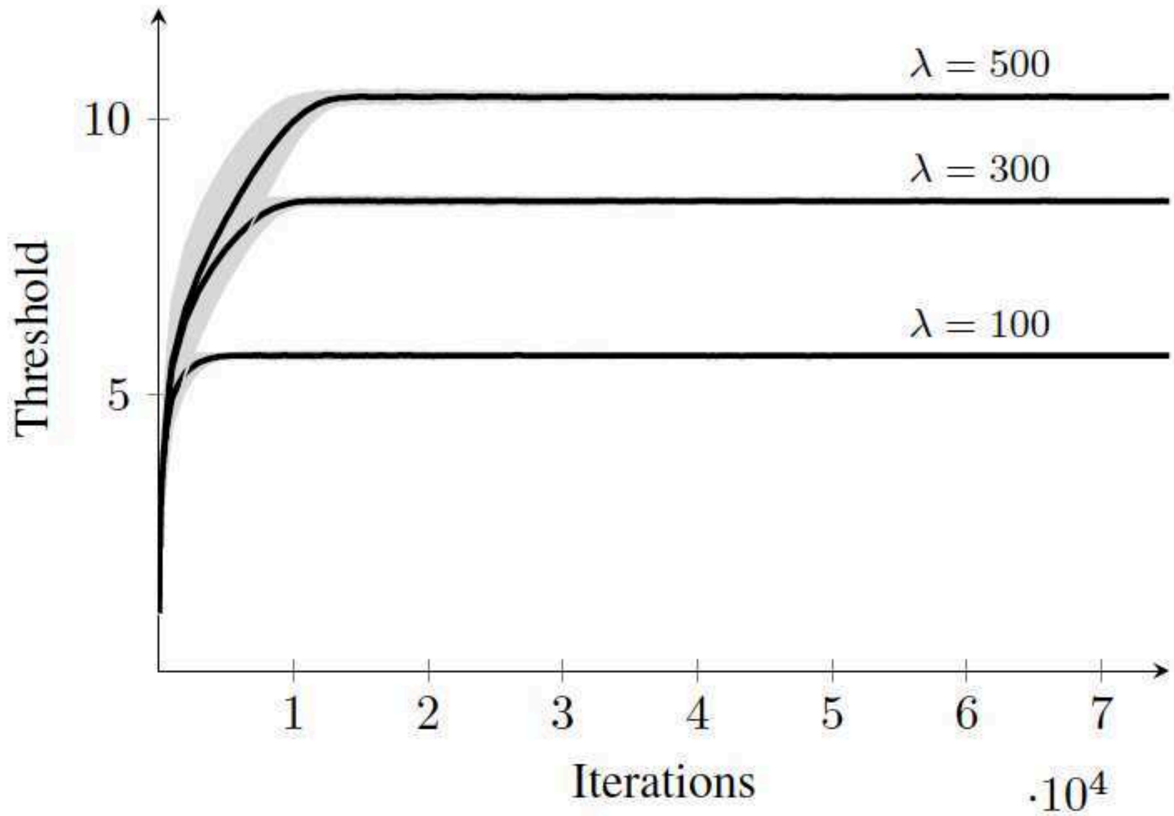


Results – 1/2

λ	Threshold k^* SA	Threshold k^* FIE	Threshold k^* Error (Absolute)	Performance $C^*_\beta(\lambda)$ SA	Performance $C^*_\beta(\lambda)$ FIE	Performance $C^*_\beta(\lambda)$ Error (Absolute)
100	4.9355	4.9298	5.7×10^{-3}	5.2511	5.2511	9.1×10^{-6}
200	6.3221	6.3086	1.4×10^{-2}	6.5221	6.5221	3.5×10^{-5}
300	7.3421	7.3289	1.3×10^{-2}	7.2208	7.2208	2.4×10^{-5}
400	8.2118	8.1764	3.5×10^{-2}	7.6654	7.6652	1.4×10^{-4}
500	8.9469	8.9177	2.9×10^{-2}	7.9700	7.9700	7.2×10^{-5}
600	9.5830	9.5854	2.5×10^{-3}	8.1886	8.1886	4.7×10^{-7}
700	10.0803	10.1984	1.2×10^{-1}	8.3515	8.3507	8.0×10^{-4}

Difference less than 10^{-2} in most cases

Results – 2/2



- Results in line with expectation
- Convergence

Conclusions

- Use of RMC methods in RL
- Applicability in Actor Only and Actor Critic methods
- Networked Control System example
- Extension to other problems

Thank You