`Learning to Control’ an Unknown System

Rahul Jain
University of Southern California

Joint work with Mukul Gagrani (USC), Ashutosh Nayyar (USC) and Yi Ouyang (UC Berkeley)
Outline

I. MDPs, Dynamic Programming
II. Bandit Models, Online Learning
III. PSDE: An RL Algorithm for Unknown MDPs
IV. PSDE Algorithm for Unknown Linear Stochastic Systems
A Markov Decision Process

\[ V_{\pi}(\theta) = \liminf_{T \to \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^{T} r(x_t, u_t) \right] \]

*Finite* State space \( X \) *Finite* Action space \( U \)

**Control**

\( \pi(u \mid x; \theta) \)

\( \theta(y \mid x,u) \) **known**

\( u \) \( y \) \( x \) \( r(x,u) \)
Dynamic Programming

- Weakly communicating finite MDP
- Optimal average reward \( V^*(\theta) = \sup_{\pi} V_\pi(\theta) \)

Bellman equation

\[
V^*(\theta) + w^*(x, \theta) = \sup_u \{ r(x, u) + \sum_y \theta(y|x, u)w^*(x, \theta) \} \\
\]

- \( w^*(x, \theta) \) is relative value function
- Solve by average-reward DP algorithms
Unknown Model

★ True $\theta_o$, unknown ~ prior $\mu$

★ Learning policy $\phi_t(h_t)$, history $h_t=$(states,actions)

★ Objective of Learning: To find a nearly optimal policy at the fastest possible rate?

Finite State space $X$  Finite Action space $U$

MDP

$\theta(y | x,u)$ Unknown

Control  Learn

$\pi(u | x; \hat{\theta})$  $\hat{\theta}$

$(x,r)$  [Borkar-Varaiya'82]
Bandit Models and Online Learning

- Reward on Heads = $1, on Tails = 0
- Objective: “max expected long-term total reward”
  $\equiv\min (expected) \text{ Regret}$
  \[
  R_T(\phi) = T\theta_{max} - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]
  \]
- Lai & Robbins (1985) lower bound $O(\log T)$
- UCB1 algorithm achieves $O(\log T)$ [Agrawal’95, Auer, et al’02]$
  g_i(t, t_i) = X_i + \sqrt{2 \log t / t_i}$
  Optimism in the Face of Uncertainty (OFU)
The (Thompson) Posterior Sampling Algorithm

- Maintain a belief (posterior distribution), $\mu_i$ over $\theta_i$
- Sample $\hat{\theta}_i$ from $\mu_i$
- Choose $i^* = \arg \max_i \hat{\theta}_i$
- Achieves (exp) regret $R_T(\phi) = O(\log T)$

- Advantage: superior numerical performance, computationally simpler
- Thompson’33, Chapelle-Li’11, Agrawal-Goyal’12

Posterior Sampling Algorithms
**Learning an Unknown MDP**

- **Learning policy** $\phi_t(h_t)$ to search over space $\Theta$

- **Objective of Learning**:
  \[ \mathcal{R}_T(\phi) = TV^*(\theta_0) - \mathbb{E}\left[\sum_{t=1}^{T} r(x_t, u_t)\right] \]

- **Lower Bound** $= \Omega(\sqrt{T})$ [Tsitsiklis, et al (2010)]

- **OFU v. PS**
The **PSDE** Algorithm: Posterior Sampling with Dynamic Episodes

The PSDE Algorithm:

- **Resample** $\theta$ from posterior $\mu_t$ at end of every episode
  - Compute policy optimal for sampled $\theta$
  - At each $t$, update posterior $\mu_t(\theta) = \mathbb{P}(\theta|h_t)$ using Bayes’ rule

Stopping Rule 1:

$t > t_k + T_{k-1}$

Stopping Rule 2:

$N_t(x, u) > 2N_{t_k}(x, u)$ for some $(x, u)$
Non-asymptotic Regret bound for PSDE

Theorem.*

If the MDP is \textit{weakly communicating} and its \textit{span} \( \leq H \), then

\[ \mathcal{R}_T(PSDE) \leq \tilde{O}(HX \sqrt{UT}) \]

where \( X \) is state space size, and \( U \) is action space size.

- Up to logarithmic factors, exact constants known
- PSDE Algorithm works with approximately optimal policies in each episode also
- Episode length can’t increase faster

Numerical Performance

Riverswim Benchmark problem

- UCRL2: [Jaksch, Ortner, Auer (2010)]
- TSMDP: [Gopalan & Mannor (2015)]
- Lazy-PSRL: [Yadkori & Szepesvari (2015)]
Proof Outline

★ For any function $f$ and RV $X$, algorithm must satisfy

$$E[f(\theta_k, X)] = E[f(\theta_o, X)]$$

★ Upper bounds number of episodes

$$K_T \leq \sqrt{2XUT \log T}$$

★ Upper bound between true and sampled parameters

$$T\mathbb{E}[V^*(\theta_o)] - \sum_{k=1}^{K_T} \mathbb{E}[T_k V^*(\theta_k)] \leq \mathbb{E}[K_T]$$
Unknown Stochastic Linear System

\[ x_{t+1} = Ax_t + Bu_t + w_t \]

\[ u_t = \pi_t(h_t) \]

\[ c_t = x_t^T Q x_t + u_t^T R u_t \]

- **Parameters** \( \theta \) unknown
- **Regret**
  \[ R_T(\pi) = \mathbb{E} \left[ \sum_{t=1}^{T} c_t - T J(\theta) \right] \]
- **Optimal control policy is linear:**
  \[ u = G(\theta)x \text{ where } G(\theta) = -(R + B^\top S(\theta B)^{-1} B^\top S(\theta) A). \]

**Assumption 1:** There is a set \( \Theta \) such that for all \( \theta \in \Theta \), there is a unique p.d. solution to the Ricatti equation
Stochastic Adaptive Control

- Classical Adaptive Control…
  - Certainty equivalence principle
    - Astrom-Wittenmark’94, Sastry’89, Narendra’89

- Cost-biased Max Likelihood approach
  - Campi and Kumar’98, Prandini-Campi’01,…

- Optimism in the Face of Uncertainty (OFU)
  - Yadkori-Szepesvari’11,’15, Van Roy, et al’12,’13,’16 (computation!)
  - Abeile-Lazaric’17 ∼O(T^{2/3})
The Posterior Sampling with Dynamic Episodes (PSDE) Learning Algorithm

★ From data \( z_t = [x_t, u_t] \), estimate parameters \( \theta \):

\[
\hat{\theta}_{t+1}(i) = \hat{\theta}_t(i) + \frac{\Sigma_t z_t (x_{t+1}(i) - \hat{\theta}_t(i)^T z_t)}{1 + z_t^T \Sigma_t z_t}
\]

\[
\Sigma_{t+1} = \Sigma_t - \frac{\Sigma_t z_t z_t^T \Sigma_t}{1 + z_t^T \Sigma_t z_t}
\]

**Posterior Sampling:** Sample parameters \( \tilde{\theta}_{t_k} \) from \( \mu_{t_k}(\hat{\theta}_{t_k}) \)

Solve Ricatti equation

Compute Gain \( G(\tilde{\theta}_{t_k}) \)

---

Dynamic Episodes

\[ T_1 \]

\[ 0 \quad t_2 \quad \ldots \quad t_k \quad t_{k+1} \quad \ldots \]

\[ T_k = T_{k-1} + 1 \]

\[ \det(\Sigma_t) < 0.5 \det(\Sigma_{t_k}) \]
Assumption 2. State space $X$ compact,

This implies for all $\theta \in \Theta$, spectral radius $\rho(A_1+B_1G(\theta)) < \delta < 1$

**Theorem.** Expected regret of PSDE, $R_T(PSDE) \leq \tilde{O}(\sqrt{T})$

Conclusions

★ Simple Posterior Sampling (PS)-based Learning-to-Control Algorithms
  ‣ For MDPs and Linear Stochastic Systems
★ Trades-off `Exploration v. Exploitation’ nearly optimally to get $O(\sqrt{T})$ regret
  ‣ Unlike OFU-type algorithms, computationally simple
  ‣ A natural design
  ‣ A deterministic schedule possible?
★ Extensions
  ‣ Continuous state space MDPs via function approximation
  ‣ Time-varying systems