Online Reinforcement Learning and Regret

Christina Lee Yu, Sean Sinclair
Cornell University
Main Question

“Given” an MDP, how do we find the optimal policy?
First setting

**Fully Known Model**

- **Known** transitions + rewards
- **Q**: *Computational complexity* of finding good policies?
First setting

Fully Known Model

- **Known** transitions + rewards
- **Q:** Computational complexity of finding good policies?
First setting

Fully Known Model

- **Known** transitions + rewards
- **Q**: Computational complexity of finding good policies?

Value Iteration
Policy Iteration

Function Approximation

[Dai,Gluzman,2022] [Liu,Xie,Modiano,2019] [Shah,Xie,Xu,2020] [Zhang,Gurvich,2020]
Second Setting

Generative Model

- **Unknown** transitions + rewards
- Can sample arbitrary (state, action)
- Q: *Sample complexity* of finding good policies?
Second Setting

Generative Model

- **Unknown** transitions + rewards
- Can sample arbitrary (state, action)
- Q: *Sample complexity* of finding good policies?

Physics Simulators

[Zhang,Zhang,Maguluri,2021] [Chen,Maguluri,Shakkottai,Shanmugam,2020]
[Agarwal,Kakade,Yang,2020] [Srikant,Ying,2019]
Second Setting

Q Learning
TD Learning

Generative Model

- *Unknown* transitions + rewards
- Can sample arbitrary (state, action)
- Q: *Sample complexity* of finding good policies?

[Zhang, Zhang, Maguluri, 2021] [Chen, Maguluri, Shakkottai, Shanmugam, 2020]
[Agarwal, Kakade, Yang, 2020] [Srikant, Ying, 2019]
Do we need another setting?

Some problems have “restricted” interaction with environment.

**Fully Known Model**
- Known transitions + rewards
- Q: *Computational complexity* of finding good policies?

**Generative Model**
- Can *sample* arbitrary (state, action)
- Q: *Sample complexity* of finding good policies?
An example

Optimizing drug dosages

Decide dosage for next three days

[Bastani et al, 2022] [Padmanabhan, Meskin, Haddad, 2017] [Kallus, Uehara, 2020]
An example

Drug dosage model

Decide dosage for next three days

[Bastani et al, 2022] [Padmanabhan, Meskin, Haddad, 2017]
[Kallus, Uehara, 2020]
An example

Drug dosage model

Decide dosage for next three days

[Bastani et al, 2022] [Padmanabhan, Meskin, Haddad, 2017] [Kallus, Uehara, 2020]
An example

Drug dosage model

Decide dosage for next three days

[Bastani et al, 2022] [Padmanabhan, Meskin, Haddad, 2017] [Kallus, Uehara, 2020]
An example

Some problems have “restricted” interaction with environment

**Fully Known Model**
- Fully understand interaction of medication and patient covariates

**Generative Model**
- Able to simulate what “would” happen for any given dosage sequence
Third Setting (this talk)

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
- “Most complex”: constrained exploration, correlated estimates,
Third Setting (this talk)

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
- “Most complex”: constrained exploration, correlated estimates,

Complicated demand dynamics

[Ipek, Mutlu, Martinez, Caruana, 2008]
Finite Horizon

A MDP is defined by: \( \mathcal{M} = \{S, A, r, T, s_0, H\} \)

- \( S \) : State space
- \( A \) : Action space
- \( r_h : S \times A \to [0, 1] \) : Reward
- \( T_h : S \times A \to \Delta(S) \) : Transitions
- \( H \) : Time horizon
- \( \pi_h : S \to \Delta(A) \) : Policy
Finite Horizon

A MDP is defined by: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, T, s_0, H\}$

- $\mathcal{S}$: State space
- $\mathcal{A}$: Action space
- $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$: Reward
- $T_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$: Transitions
- $H$: Time horizon
- $\pi_h : \mathcal{S} \to \Delta(\mathcal{A})$: Policy

Example:
- From 5 mg to 15 mg to 5 mg
- Time horizon $H = 3$
Bellman Equations

The Bellman Equations note that:

\[ V_{h+1}^\pi(s) = \mathbb{E}_{A \sim \pi_h(s)}[r_h(s, A) + \mathbb{E}_{S' \sim T_h(\cdot|s, A)}[V_{h+1}^\pi(S')]] \]

\[ Q_{h}^\pi(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot|s, a)}[V_{h+1}^\pi(S')] \]
Main Question

Unknown transition + reward

Over sequence of episodes:

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
Main Question

**Online Model**

- Can only **sample trajectories** under some chosen policy
- Q: **Regret** incurred over time compared to optimal policy

Unknown transition + reward

Over sequence of **episodes**:
- Pick current policy $\pi^k$
Main Question

Unknown transition + reward

Over sequence of episodes:
- Pick current policy $\pi^k$
- Execute over $H$ steps (episode)

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
Main Question

Unknown transition + reward

Over sequence of episodes:
- Pick current policy \( \pi^k \)
- Execute over \( H \) steps (episode)
- Collect dataset and update policy
  \[ \{(S^k_1, A^k_1, R^k_1), \ldots, (S^k_H, A^k_H, R^k_H)\} \]

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
Main Question

**Online Model**
- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy

Unknown transition + reward

Over sequence of episodes:
- Pick current policy $\pi^k$
- Execute over $H$ steps (episode)
- Collect dataset and update policy

\[
\{(S_1^k, A_1^k, R_1^k), \ldots, (S_H^k, A_H^k, R_H^k)\}
\]

Horizon $H = \text{Number of dosage decisions}$
Episodes $K = \text{Number of homogenous patients}$

Drug dosage model
Main Question

Unknown transition + reward

Over sequence of episodes:
- Pick current policy $\pi^k$
- Execute over $H$ steps (episode)
- Collect dataset and update policy
  $$\{ (S_1^k, A_1^k, R_1^k), \ldots, (S_H^k, A_H^k, R_H^k) \}$$

Goal: Minimize regret:

$$\text{REGRET}(K) = \sum_{k=1}^{K} V_1^*(s_0) - V_1^{\pi^k}(s_0)$$

Online Model

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy
Not into regret?

**Goal:** Minimize regret:

\[
\text{REGRET}(K) = \sum_{k=1}^{K} V_1^*(s_0) - V_1^{\pi_k}(s_0)
\]

**Theorem:** If regret is sublinear in \( K \), can obtain PAC-style sample complexity bound for learning a good policy:

\[
\text{REGRET}(K) \leq K^{1-\alpha}
\]

\[
\text{NUMBER SAMPLES}(\epsilon) \leq \epsilon^{-1/\alpha}
\]
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to “compensate”?
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to "compensate"?
Policy Comparison

Compare two policies \( \pi^{Alg}, \pi^{Ref} \)

Two policies differ on chosen action, how much to "compensate"?

\[
\begin{align*}
\pi^{Alg}(s_h) & \quad r(s_h, \pi^{Alg}(s_h)) \\
\pi^{Ref}(s_h) & \quad r(s_h, \pi^{Ref}(s_h)) \\
\pi^{Alg}(s_h) & \quad \pi^{Ref}(s_h)
\end{align*}
\]

\[
\begin{align*}
2 \cdot [r(s_h, \pi^{Ref}(s_h)) - r(s_h, \pi^{Alg}(s_h))]
\end{align*}
\]
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to “compensate”?

$Q_h^{\pi^{Alg}}(s_h, \pi^{Ref}(s_h)) - V_h^{\pi^{Alg}}(s_h)$
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to “compensate”?

$$Q_{h}^{\pi^{Alg}}(s_{h}, \pi^{Ref}(s_{h})) - V_{h}^{\pi^{Alg}}(s_{h})$$

Play $\pi^{Ref}$ now, then $\pi^{Alg}$

Value $\pi^{Alg}$
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to "compensate"?

$$Q_h^{\pi^{Alg}}(s_h, \pi^{Ref}(s_h)) - V_h^{\pi^{Alg}}(s_h) = A_h^{\pi^{Alg}}(s_h, \pi^{Ref}(s_h))$$

"Advantage" function
Recall...

Given an MDP, how do we find the optimal policy?

**Fully Known Model**
- Known transitions + rewards
- Q: Computational complexity of finding good policies?

**Two Approaches**
- Value Iteration
- Policy Iteration
Given an MDP, how do we find the optimal policy?

**Fully Known Model**
- **Known** transitions + rewards
- **Q:** Computational complexity of finding good policies?

**Two Approaches**
- Value Iteration
- Policy Iteration
Given an MDP, how do we find the optimal policy?

**Online Model**

- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy

**Two Approaches**

Value Based Policy Based
Given an MDP, how do we find the optimal policy?

**Online Model**
- Can only **sample trajectories** under some chosen policy
- Q: **Regret** incurred over time compared to optimal policy

**Two Approaches**
- Value Based
- Policy Based
Goal: Find the best “base-stock” policy:

\[
\pi_S(I_t) = \begin{cases} 
I_t - S & I_t \leq S \\
0 & I_t \geq S
\end{cases}
\]
Policy Based

**Goal**: Maximize $\sup_{\theta \in \Theta} V^{\pi_{\theta}}(s_0)$
Goal: Maximize $\sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta)$
Policy Based

**Goal**: Maximize \( \sup_{\theta \in \Theta} V^{\pi_{\theta}}(s_0) = \sup_{\theta \in \Theta} J(\theta) \)

Use existing *stochastic* optimization algorithms
Goal: Maximize \( \sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta) \)

Use existing **stochastic** optimization algorithms

- Zero-Order (Gradient-Free)  
  [Berahas, Byrd, Nocedal, 2019]  
  [Lei, Chen, Li, Zheng, 2022]  
  [Qian, Yu, 2021]

- First-Order (Gradient-Based)  
  [Bhandari, Russo, 2019]

- Second-Order (Hessian-Based)  
  [Wu, Mansimov, Grosse, Liao, Ba, 2017]
Policy Based

Goal: Maximize $\sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta)$

Primitive:
How can we compare value of two policies?
Policy Comparison

Compare two policies $\pi^{Alg}$, $\pi^{Ref}$

Two policies differ on chosen action, how much to "compensate"?

$$Q_h^{\pi^{Alg}}(s_h, \pi^{Ref}(s_h)) - V_h^{\pi^{Alg}}(s_h) = A_h^{\pi^{Alg}}(s_h, \pi^{Ref}(s_h))$$

Sum over trajectories from $\pi^{Ref}$
Goal: Maximize $\sup_{\theta \in \Theta} V^{\pi_{\theta}}(s_0) = \sup_{\theta \in \Theta} J(\theta)$

Policy Difference Lemma:

$$V^{\pi_{Ref}} - V^{\pi_{Alg}} = \sum_{h=1}^{H} \mathbb{E}_{(S,A) \sim \text{Pr}^{Ref}_{\pi_h}} [A^{\pi_{Ref}}_{h}(S, A)]$$

Can evaluate using Monte-Carlo roll outs under current policy

Used to guarantee one-step improvement

[Kakade2002]
Policy Based

**Goal:** Maximize \( \sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta) \)

Use existing **stochastic** optimization algorithms

- Zero-Order (Gradient-Free) [Berahas, Byrd, Nocedal, 2019] [Lei, Chen, Li, Zheng, 2022] [Qian, Yu, 2021]
- First-Order (Gradient-Based) [Bhandari, Russo, 2019]
- Second-Order (Hessian-Based) [Wu, Mansimov, Grosse, Liao, Ba, 2017]
Goal: Maximize \[ \sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta) \]

What even is \( \nabla J(\theta) \)?

Policy Gradient Theorem:

\[ \nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_\theta} [\nabla_{\theta} \log \pi_\theta(a | s) A^{\pi_\theta}(s, a)] \]

Can evaluate using Monte-Carlo roll outs under current policy

[Williams1992]
Restricted Policies

**Goal:** Maximize $\sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta)$

Use prior domain knowledge to find restricted class of policies

[Agrawal, Jia, 2019]
Restricted Policies

**Goal:** Maximize \( \sup_{\theta \in \Theta} V^{\pi_{\theta}}(s_0) = \sup_{\theta \in \Theta} J(\theta) \)

Use prior domain knowledge to find restricted class of policies

**Inventory Control**

"Base Stock" policies are provably near-optimal

[Agrawal,Jia,2019]
Goal: Maximize $\sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta)$

Use prior domain knowledge to find restricted class of policies

Exploit structured properties of $J(\theta)$:
- Strongly convex
- Evaluate exactly over traces

Inventory Control

“Base Stock” policies are provably near-optimal

[Agrawal, Jia, 2019]
"Dual" Approach

Goal: Maximize \( \sup_{\theta \in \Theta} V^{\pi_\theta}(s_0) = \sup_{\theta \in \Theta} J(\theta) \)

Characterizes conditions when objective has no local maxima

Assumption 1: Differentiability / continuity of objective

Assumption 2: Closure of policy space under policy-iteration steps

[Bhandari, Russo, 2019]
“Dual” Approach

Goal: Maximize $\sup_{\theta \in \Theta} V^{\pi_0}(s_0) = \sup_{\theta \in \Theta} J(\theta)$

Characterizes conditions when objective has no local maxima

Assumption 1: Differentiability / continuity of objective

Assumption 2: Closure of policy space under policy-iteration steps

Stochastic Queueing Network

Linear Quadratic Regulator

[Bhandari, Russo, 2019]
Given a MDP, how do we find the optimal policy?

**Online Model**
- Can only sample trajectories under some chosen policy
- Q: Regret incurred over time compared to optimal policy

**Two Approaches**
- Value Based
- Policy Based
Value Based

The Bellman Optimality Equations note that:

\[ V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s, a) \]

\[ Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot | s, a)}[V_{h+1}^*(S')] \]
Value Based

The Bellman Optimality Equations note that:

\[ V_h^*(s) = \max_{a \in A} Q_h^*(s, a) \]

\[ Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot | s, a)}[V_{h+1}^*(S')] \]

**Model-Based**
- Estimate \( r \) and \( T \)
- Compute \( Q^* \)
- Play greedy w.r.t. \( Q^* \)
- Time complexity / storage scales \( HS^2 A \)

**Model-Free**
- Estimate \( Q^* \) directly
- Play greedy w.r.t. \( Q^* \)
- Better time complexity / storage (only \( HSA \))
Optimism Principle

$H - h$

$Q_h^*$

$a_1$

$a_2$

$a_3$

Optimistic estimate

True value

[Simchowitz, Jamieson, 2019]
Optimism Principle

Optimistic estimate

True value

$H - h$

$Q^*_h$

$a_1$

$a_2$

$a_3$

[Simchowitz, Jamieson, 2019]
Optimism Principle

$H - h$

$Q_h^*$

$a_1$

$a_2$

$a_3$

Optimistic estimate

True value

[Simchowitz, Jamieson, 2019]
Optimism Principle

\[ H - h \]

\[ Q_h^* \]

\[ a_1 \] \hspace{1cm} \[ a_2 \] \hspace{1cm} \[ a_3 \]

Optimistic estimate

True value

[Simchowitz, Jamieson, 2019]
Optimism Principle

Estimates converge to true value for optimal actions on sample paths

Optimistic estimate
True value

[Simchowitz, Jamieson, 2019]
Three Invariants

1. Optimistic Estimates

\[ \overline{Q}_h(s, a) \geq Q^*_h(s, a) \]

2. Monotone non-increasing, decrease “fast enough”

\[ \overline{Q}_h(s, a) - Q^*_h(s, a) \sim \frac{1}{\sqrt{t}} \]

3. Play greedy

\[ \pi_h^k(s) = \arg\max_{a \in \mathcal{A}} \overline{Q}_h(s, a) \]
Value Based

The Bellman Optimality Equations note that:

\[ V_h^*(s) = \max_{a \in A} Q_h^*(s, a) \]

\[ Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(s, a)}[V_{h+1}(S')] \]

---

**Model-Based**

- Estimate \( r \) and \( T \)
- Compute \( Q^* \)
- Play greedy w.r.t. \( Q^* \)
- Time complexity / storage scales \( HS^2A \)

---

**Model-Free**

- Estimate \( Q^* \) directly
- Play greedy w.r.t. \( Q^* \)
- Better time complexity / storage (only \( HSA \))
Model Based

The Bellman Optimality Equations note that:

\[ V^*_h(s) = \max_{a \in A} Q^*_h(s,a) \]

\[ Q^*_h(s,a) = r_h(s,a) + \mathbb{E}_{S' \sim T_h(\cdot|s,a)}[V^*_{h+1}(S')] \]

At start of episode \( k \), have collected data: \( \mathcal{D}^k \)

[Azar,Osband,Munos,2017]
Model Based

The **Bellman Optimality Equations** note that:

\[
V^*_h(s) = \max_{a \in \mathcal{A}} Q^*_h(s, a)
\]

\[
Q^*_h(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot | s, a)}[V^*_{h+1}(S')]
\]

At start of episode k, have collected data: \(\mathcal{D}^k\)

Estimate reward and transition via empirical: \(\bar{r}_h\) \(\overline{T}_h(\cdot | s, a)\)

[Azar,Osband,Munos,2017]
Model Based

The **Bellman Optimality Equations** note that:

\[
V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s, a)
\]

\[
Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim \mathcal{T}_h(\cdot | s, a)} [V_{h+1}^*(S')]\]

At start of episode \( k \), have collected data: \( \mathcal{D}^k \)

Estimate reward and transition via empirical:

\( \overline{r}_h \quad \overline{T}_h(\cdot | s, a) \)

Plug estimates into Bellman Optimality Equations

[Azar,Osband,Munos,2017]
Model Based
Estimate reward and transition via empirical: $\bar{r}_h$, $\bar{T}_h(\cdot | s, a)$

Plug estimates into Bellman Optimality Equations

\[
\bar{V}_h(s) = \max_{a \in \mathcal{A}} \bar{Q}_h(s, a)
\]

\[
\bar{Q}_h(s, a) = \bar{r}_h(s, a) + \mathbb{E}_{S' \sim \bar{T}_h(\cdot | s, a)}[\bar{V}_{h+1}(S')] + \lambda \frac{1}{\sqrt{t}}
\]

\[
\pi_h(s) = \arg\max_{a \in \mathcal{A}} \bar{Q}_h(s, a)
\]

[Azar, Osband, Munos, 2017]
Model Based

Estimated value iteration

\[ \overline{V}_h(s) = \max_{a \in \mathcal{A}} \overline{Q}_h(s, a) \]
\[ \overline{Q}_h(s, a) = \overline{r}_h(s, a) + \mathbb{E}_{s' \sim T_h(\cdot | s, a)}[\overline{V}_{h+1}(s')] + \lambda \frac{1}{\sqrt{t}} \]
\[ \pi_h(s) = \arg\max_{a \in \mathcal{A}} \overline{Q}_h(s, a) \]

True value iteration

\[ V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s, a) \]
\[ Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim T_h(\cdot | s, a)}[V_{h+1}^*(s')] \]
\[ \pi_h^*(s) = \arg\max_{a \in \mathcal{A}} Q_h^*(s, a) \]

Empirical value iteration with reward and transition estimates

[Azar, Osband, Munos, 2017]
Three Invariants

1. Optimistic Estimates
   \[ \overline{Q}_h(s, a) \geq Q^*_h(s, a) \]

2. Monotone non-increasing, decrease “fast enough”
   \[ \overline{Q}_h(s, a) - Q^*_h(s, a) \sim \frac{1}{\sqrt{t}} \]

3. Play greedy
   \[ \pi^k_h(s) = \arg\max_{a \in A} \overline{Q}_h(s, a) \]

[Simchowitz, Jamieson, 2019]
Reduction to bandit

If horizon $H = 1$, estimates reduce:

$$
\overline{Q}_1(s, a) = \overline{r}_1(s, a) + \lambda \frac{1}{\sqrt{t}}
$$

$$
\pi_1(s) = \argmax_{a \in A} \overline{Q}_1(s, a)
$$
Reduction to bandit

If horizon $H = 1$, estimates reduce:

$$
\bar{Q}_1(s, a) = \bar{r}_1(s, a) + \lambda \frac{1}{\sqrt{t}}
$$

$$
\pi_1(s) = \text{argmax}_{a \in A} \bar{Q}_1(s, a)
$$

Reduces to UCB algorithm in $H = 1$ setting
Model Based

**Theorem:** In a $H$-step MDP we have that:

$$\text{Regret}(K) \leq H^{3/2} \sqrt{SAK}$$

- Optimal dependence on $K$
- Suboptimal time + space complexity
- Dependence on $H$ still current research

[Jaksch, Ortner, Auer, 2010]  
[Azar, Osband, Munos, 2017]  
[Agrawal, Jia, 2017]
Model Based

Regret guarantees are worst case, don’t capture specific problem structure

In practice: exploration is done via $\epsilon$ exploration or bonus terms are tuned for performance

[Azar,Osband,Munos,2017]
Value Based

The Bellman Optimality Equations note that:

\[ V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s, a) \]

\[ Q_h^*(s, a) = r_h(s, a) + \mathbb{E}_{S' \sim T_h(\cdot|s, a)}[V_{h+1}^*(S')] \]

Model-Based

- Estimate \( r \) and \( T \)
- Compute \( Q^* \)
- Play greedy w.r.t. \( Q^* \)
- Time complexity / storage scales \( HS^2A \)

Model-Free

- Estimate \( Q^* \) directly
- Play greedy w.r.t. \( Q^* \)
- Better time complexity / storage (only \( HSA \))
Model Free

Follows update procedure:

\[
\bar{V}_h(s) = \max_{a \in \mathcal{A}} \bar{Q}_h(s, a)
\]

\[
\bar{Q}_h(S_h, A_h) = (1 - \alpha_t)\bar{Q}_h(S_h^k, A_h^k) + \alpha_t(R + \bar{V}_{h+1}(S_{h+1}) + \lambda \frac{1}{\sqrt{t}})
\]

\[
\pi_h(s) = \operatorname{argmax}_{a \in \mathcal{A}} \bar{Q}_h(s, a)
\]

Empirical fixed point iteration with exploration bonuses

\[\text{[Jin,Allen-Zhu,Bubeck,Jordan,2018]}\]
Model Free

Follows update procedure:

\[
\overline{V}_h(s) = \max_{a \in A} \overline{Q}_h(s, a)
\]

\[
\overline{Q}_h(S_h, A_h) = (1 - \alpha_t)\overline{Q}_h(S_h^k, A_h^k) + \alpha_t (R + \overline{V}_{h+1}(S_{h+1}) + \lambda \frac{1}{\sqrt{t}})
\]

\[
\pi_h(s) = \arg\max_{a \in A} \overline{Q}_h(s, a)
\]

Learning rate favors later updates

\[
\alpha_t = \frac{H+1}{H+t}
\]

[Jin, Allen-Zhu, Bubeck, Jordan, 2018]
**Model Free**

**Informal Theorem:** In a H-step MDP we have that:

\[ \text{Regret}(K) \leq H^{5/2} \sqrt{SAK} \]

- Strong relation to theory of Stochastic Approximation (Robbins Munro)
- Optimal dependence on K
- Better time + space complexity than model-based algorithms
- Dependence on H still current research

[Jin, Allen-Zhu, Bubeck, Jordan, 2018]
Model Free vs Model Based

Some folklore comparisons:

- Performance Model Based > Model Free
- Model Based more compute, easier implementation
- **Open Question**: Tradeoff minimax regret and storage/compute
Refined Regret Guarantees

Regret guarantees are worst case, don’t capture specific problem structure

Logarithmic Regret:  
[Simchowitz, Jamieson, 2019]  
[He, Zhou, Gu, 2020]  
[Yang, Yang, Du, 2021]

“Variance” Dependence:  
[Zanette, Brunskill, 2019]

“Dimension” Dependence:  
[Sam, Cheng, Yu, 2022]  
[Osband, Roy, 2014]  
[Jiang, Krishnamurthy, Agarwal, Langford, Schapire, 2017]  
[Sun, Jiang, Krishnamurthy, Agarwal, Langford, 2018]
So far:

Saw algorithms designed with value and policy iteration for tabular (discrete) MDPs.

However, even if problem is tabular:

```
MemoryError: Unable to allocate 31.9 GiB for an array with shape (3094, 720, 1280, 3) and data type float32
```

\[ H = 3 \]
\[ S = 3094 \times 720 \]
\[ A = 1280 \]
So far:

To design algorithms that scale...

Function Approximation  Modelling Assumptions
Online Reinforcement Learning and Regret

Sean Sinclair
Cornell University