RL with continuous action spaces
Linear MDP

Aarti Singh

Machine Learning 10-734
Oct 30, 2025

Slides courtesy: Yuejie Chi, Wen Sun

Continuous action spaces

Bandits:

Reward is linear, Lipshitz, GP, NN, …

e.g. $r^*(x) = x^T\theta^*$ x, θ^* are d-dimensional

Why?

Optimal action a^* depends on reward r^*

MDP:

Optimal policy π^* depends on Q^*

Attempt 1: Value function Q^* is linear

e.g. $Q^*(s,a) = \phi(s,a)^T \theta^*$ ϕ, θ^* are d-dimensional

feature representation of (state, action) pair

Linear Q doesn't suffice

- No, regret bounds:

	Linear Bandits	Q^* Linear
Upper bound	$\tilde{O}(\mathrm{d}\sqrt{T})$ LinUCB	$O(e^H)$
Lower bound	$\Omega(\mathrm{d}\sqrt{T})$	$\Omega(poly(H))$

- Need additional assumptions!

Linear MDP

- Attempt 2: Use a parametric function to approximate $r(s,a), P(s'|s,a)$
  - E.g. Linear $r_h(s,a) = w_h^T \phi(s,a)$

    Low-rank $P_h(s'|s,a) = \mu_h(s')^T \phi(s,a)$

- Also called low-rank MDPs

- For all π, $P^\pi = \Psi^T \Phi^\pi$

  SxS $P(s'|s,\pi(s))$ = Sxd dxS $\phi(s,\pi(s))$

  d-rank

  parameters

Linear MDP

- Use a parametric function to approximate $r(s,a), P(s'|s,a)$
  - E.g. Linear $r_h(s,a) = w_h^T \phi(s,a)$

    Low-rank $P_h(s'|s,a) = \mu_h(s')^T \phi(s,a)$

- Learn the parameters , value iteration, policy iteration

- Training data $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1})$ gathered online by the agent/learning algorithm

Fundamental property of Linear MDP

For any value function V, Bellman update T_h V is linear

where Bellman operator T_h

Bellman update $\quad T_h V(s) = r_h\big(s, \pi(s)\big) + \sum_{s'} P\big(s' \mid s, \pi(s)\big) V^{\pi}(s')$

Proof: $\quad r_h(s,a) = w_h^T \, \phi(s,a)$

$$\sum_{s'} P(s' \mid s,a) V^{\pi}(s') = \sum_{s'} \mu_h(s')^T \phi(s,a) V^{\pi}(s')$$

$$= \sum_{s'} \mu_h(s')^T V^{\pi}(s') \; . \phi(s,a)$$

eqv. parameter for V function

Implies Q* is linear! But stronger

$V^* = T_h V^*$, $Q^* = T Q^*$

LSVI-UCB: Least Square Value Iteration with UCB

Value iteration at episode n using $\{s_h^i, a_h^i, r_h^i, s_{h+1}^i\}_{h=1,i=1}^{H-1,n-1}$

$$\widehat{V}_H^n(s) = 0, \forall s$$

For h = H-1, H-2, …, 1

$$\theta_h^n \leftarrow \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n-1} \left(\left\langle \phi(s_h^i, a_h^i), \theta \right\rangle - r_h^i - \widehat{V}_{h+1}^n(s_{h+1}^i) \right)^2 + \lambda \|\theta\|_2^2$$

$$\widehat{Q}_h^n(s,a) = \min \left\{ b_h^n(s,a) + \left\langle \phi(s,a), \theta_h^n \right\rangle, \quad H \right\}, \forall s, a$$

$\hat{Q} > Q^*$ upper confidence bound

$$\widehat{V}_h^n(s) = \max_a \widehat{Q}_h^n(s,a), \qquad \pi_h^n(s) = \arg\max_a \widehat{Q}_h^n(s,a), \forall s$$

LSVI-UCB: Least Square Value Iteration with UCB

At each episode n

For h = H, H-1, ..., 1

Estimate using data from past episodes

$$\widehat{Q}_h^n(s,a) = \min\left\{ b_h^n(s,a) + \left\langle \phi(s,a), \theta_h^n \right\rangle, \quad H \right\}, \forall s, a$$

Plan

$$\pi_h^n(s) = \arg\max_a \widehat{Q}_h^n(s,a), \forall s$$

Execute

Roll out π_h^n to collect trajectory and add to data

LSVI-UCB: Least Square Value Iteration with UCB

$$\widehat{Q}_h^n(s,a) = \min\left\{ b_h^n(s,a) + \left\langle \phi(s,a), \theta_h^n \right\rangle, H \right\}, \forall s, a$$

$$\theta_h^n \leftarrow \operatorname*{argmin}_\theta \sum_{i=1}^{n-1} \left(\left\langle \phi(s_h^i, a_h^i), \theta \right\rangle - r_h^i - \widehat{V}_{h+1}^n(s_{h+1}^i) \right)^2 + \lambda \|\theta\|_2^2$$

This implies $\theta_h^n = \Lambda_h^{n-1} \sum_{i=1}^{N-1} \phi_h^i (r_h^i + \hat{V}_{h+1}^i(s_{h+1}^i))$

where $\phi_h^i = \phi(s_h^i, a_h^i)$ and $\Lambda_h^n = \sum_{i=1}^{N-1} \phi_h^i \phi_h^{i\,T}$

and bonus $b_h^n(s,a) = \|\phi(s,a)\|_{\Lambda_h^{n-1}} \beta$ (derived later)

Bandits: If H = 1, recover LinUCB for Linear bandits

LSVI-UCB Regret

- With probability at least 1 – δ, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*} - V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3T})$$

Key proof idea: Combine proofs of LinUCB and UCB-VI

bandits

Tabular

LSVI-UCB Regret

- With probability at least 1 – δ, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*} - V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3T})$$

Step 1: Design bonus to guarantee Optimism i.e. $V_h^*(s) \leq \hat{V}_h^n(s)\ \forall h, s$
implies Bonus $b_h^n(s,a)$ = Transition error $e_h^n(s,a)$

Step 2: Characterize transition error

Step 3: Apply simulation lemma

$$\hat{V}_0^n(s_0) - V_0^{\pi^n}(s_0) \leq \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^n}}\left[b_h^n(s,a) + \underbrace{(\hat{P}_h^n(\cdot\,|\,s,a) - P_h(\cdot\,|\,s,a)) \cdot \hat{V}_{h+1}^n}_{e_h^n(s,a)}\right]$$

LSVI-UCB Regret

- With probability at least 1 – δ, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*} - V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3T})$$

Step 1: Design bonus to guarantee Optimism i.e. $V_h^*(s) \leq \hat{V}_h^n(s)\ \forall h, s$

For any policy π, consider

$$\hat{Q}_h^n(s,a) - Q_h^{\pi}(s,a)$$
$$= b_h^n(s,a) + < \phi(s,a), \theta_h^n > - r_h(s,a) - \mathbb{E}_{s' \sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$
$$= b_h^n(s,a) + < \phi(s,a), w_h + \theta_h^{\hat{V}} > + < \phi(s,a), \Lambda_h^{n-1} \sum_{i=1}^{N-1} \phi_h^i \eta_h^i) >$$
$$- r_h(s,a) - \mathbb{E}_{s' \sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$

$$\theta_h^n = \Lambda_h^{n-1} \sum_{i=1}^{N-1} \phi_h^i (r_h^i + \hat{V}_{h+1}^i(s_{h+1}^i))$$
$$= \Lambda_h^{n-1} \sum_{i=1}^{N-1} \phi_h^i (r_h^i + \mathbb{E}_{s' \sim P(.|s_h^i, a_h^i)}[\widehat{V}_{h+1}^i(s')] + \eta_h^i)$$ where η_h^i is transition error
$$= \Lambda_h^{n-1} \sum_{i=1}^{N-1} \phi_h^i (\phi_h^i w_h + \phi_h^i \theta_h^{\hat{V}} + \eta_h^i)$$ Bellman update is linear in linear MDP

LSVI-UCB Regret

- With probability at least $1-\delta$, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*}-V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3T})$$

Step 1: Design bonus to guarantee Optimism i.e. $V_h^*(s) \le \hat{V}_h^n(s)\ \forall h, s$

For any policy π, consider

$$\hat{Q}_h^n(s,a) - Q_h^{\pi}(s,a)$$
$$= b_h^n(s,a) + <\phi(s,a), \theta_h^n> - r_h(s,a) - \mathbb{E}_{s'\sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$
$$= b_h^n(s,a) + <\phi(s,a), w_h + \theta_h^{\hat{V}}> + <\phi(s,a), \Lambda_h^{n\,-1}\textstyle\sum_{i=1}^{N-1}\phi_h^i\eta_h^i> - r_h(s,a) - \mathbb{E}_{s'\sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$
$$= b_h^n(s,a) + <\phi(s,a), \theta_h^{\hat{V}}> + <\phi(s,a), \Lambda_h^{n\,-1}\textstyle\sum_{i=1}^{N-1}\phi_h^i\eta_h^i> - \mathbb{E}_{s'\sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$
$$= b_h^n(s,a) + \mathbb{E}_{s'\sim P(.|s,a)}[\hat{V}_{h+1}^{i}(s')] + <\phi(s,a), \Lambda_h^{n\,-1}\textstyle\sum_{i=1}^{N-1}\phi_h^i\eta_h^i> - \mathbb{E}_{s'\sim P(.|s,a)}[V_{h+1}^{\pi}(s')]$$
$$= b_h^n(s,a) + \mathbb{E}_{s'\sim P(.|s,a)}[\hat{V}_{h+1}^{i}(s')] - \mathbb{E}_{s'\sim P(.|s,a)}[V_{h+1}^{\pi}(s')] + e_h^n(s,a)$$
≥ 0. by induction for optimism and choice of bonus

$b - e \ge 0$ choose bonus

if $\hat{V}_{h+1} \ge V_{h+1} \Rightarrow \hat{Q}_h \ge Q_h \Rightarrow \hat{V}_h \ge V_h$

LSVI-UCB Regret

- With probability at least 1 – δ, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*}-V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3T})$$

Step 2: Characterize transition error

$$\text{w.p. atleast 1-}\delta,\ |e_h^n(s,a)| = |\phi^T(s,a)\Lambda_h^{n^{-1}}\textstyle\sum_{i=1}^{n-1}\phi_h^i\eta_h^i\,|$$

$$\leq \|\phi\|_{\Lambda_h^{n-1}}\left\|\textstyle\sum_{i=1}^{n-1}\phi_h^i\eta_h^i\right\|_{\Lambda_h^{n-1}}$$

$$\leq \|\phi\|_{\Lambda_h^{n-1}}\beta$$

where $\left\|\sum_{i=1}^{n-1}\phi_h^i\eta_h^i\right\|_{\Lambda_h^{n-1}} \leq \beta = \tilde{O}(dH)$ from LinUCB proof

LSVI-UCB Regret

- With probability at least 1 – δ, LSVI-UCB has regret

$$\text{Regret}, \sum_{n=1}^{N}[V^{*}-V^{\pi_n}] = \tilde{O}(H^2\sqrt{d^3N})$$

poly(H)

Step 3: Apply simulation lemma

$$\widehat{V}_0^n(s_0) - V_0^{\pi^n}(s_0) \le \sum_{h=0}^{H-1} \mathbb{E}_{s,a\sim d_h^{\pi^n}}\left[b_h^n(s,a) + \overbrace{(\widehat{P}_h^n(\cdot\,|s,a) - P_h(\cdot\,|s,a))\cdot \widehat{V}_{h+1}^n}^{e_h^n(s,a)}\right]$$

$$\sum_{n=1}^{N}[V^{*}-V^{\pi_n}] \le \sum_{n=1}^{N}[\widehat{V}^{*}-V^{\pi_n}] \le \sum_{n=1}^{N}\sum_{h=0}^{H-1}\mathbb{E}[b_h^n(s_h^n,a_h^n) + e_h^n(s_h^n,a_h^n)]$$

$2\beta\|\phi\|_{\Lambda^{-1}}$

optimism

Sim lemma

$$= 2\beta\sum_{h=0}^{H-1}\sqrt{N}\sqrt{\sum_{n=1}^{N}\left\|\phi_h^n\right\|^2_{\Lambda_h^{n-1}}} = \tilde{O}(\beta H\sqrt{dN}) = \tilde{O}(H^2 d\sqrt{dN})$$

$\beta = O(Hd)$

Cauchy-Schwarz

$\tilde{O}(d)$ using elliptical potential lemma from LinUCB

Fundamental property of Linear MDP

For any value function V, Bellman update T_h V is linear

where Bellman operator

$$T_h V(s) = r_h\big(s, \pi(s)\big) + \sum_{s'} P\big(s' \mid s, \pi(s)\big) V^{\pi}(s')$$

Proof: $r_h(s,a) = w_h^T \phi(s,a)$

$$\sum_{s'} P(s' \mid s,a) V^{\pi}(s') = \sum_{s'} \mu_h(s')^T \phi(s,a) V^{\pi}(s')$$

$$= \sum_{s'} \mu_h(s')^T V^{\pi}(s') \; . \phi(s,a)$$

θ_h^V

Implies Q* is linear! But stronger

LSVI-UCB: Least Square Value Iteration with UCB

Value iteration at episode n using $\{s_h^i, a_h^i, r_h^i, s_{h+1}^i\}_{h=1,i=1}^{H-1,n-1}$

$$\widehat{V}_H^n(s) = 0, \forall s$$

For h = H-1, H-2, …, 1

LS uses linear Bellman update property

$$\theta_h^n \leftarrow \operatorname*{argmin}_\theta \sum_{i=1}^{n-1} \left(\left\langle \phi(s_h^i, a_h^i), \theta \right\rangle - r_h^i - \widehat{V}_{h+1}^n(s_{h+1}^i) \right)^2 + \lambda \|\theta\|_2^2$$

$$\widehat{Q}_h^n(s,a) = \min \left\{ b_h^n(s,a) + \left\langle \phi(s,a), \theta_h^n \right\rangle, \quad H \right\}, \forall s, a$$

$$\widehat{V}_h^n(s) = \max_a \widehat{Q}_h^n(s,a), \qquad \pi_h^n(s) = \arg\max_a \widehat{Q}_h^n(s,a), \forall s$$

RL with continuous action spaces

- Approach 3:

  $\neq Q^*$ is linear

  Linear Bellman completeness: For any linear Q function, its Bellman update is also linear

  Given feature ϕ, take any linear function $w^\top \phi(s,a)$:

$$\forall h, \exists \theta \in \mathbb{R}^d, s.t., \theta^\top \phi(s,a) = r(s,a) + \mathbb{E}_{s' \sim P_h(s,a)} \max_{a'} w^\top \phi(s',a'), \forall s, a$$

Bellman update

stronger than assuming Q^* linear

but

weaker than r is linear & P is low rank

RL with continuous action spaces

General function approximation

- Use a parametric function, $Q(s, a; \Theta)$, to approximate $Q^*(s, a)$
  - E.g. $Q(s, a; \Theta)$ = Neural network

    Linear $\theta^T \phi(s, a)$ where $\phi(s, a)$ - features

- Learn the parameters , policy value iteration
- Training data $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1})$ gathered online by the agent/learning algorithm

RL with continuous action spaces

General function approximation

- Use a parametric function, $Q(s, a; \Theta)$, to approximate $Q^*(s, a)$
  - E.g. $Q(s, a; \Theta)$ = Neural network

    Linear $\theta^T \phi(s, a)$ where $\phi(s, a)$ - features

Q^* is realizable not enough!

Need **Bellman completeness**: For any Q function in $\mathcal{F}$, its Bellman update is also in $\mathcal{F}$