Optimal Gradient-based Algorithms for Non-concave Bandit Optimization

Jason D. Lee
Princeton University

Joint work with

Qi Lei, Baihe Huang, Kaixuan Huang, Sham M Kakade, Jason D Lee, Runzhe Wang, and Jiaqi Yang

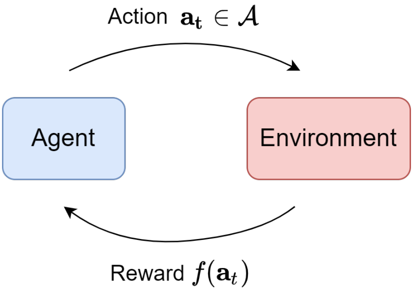
Slides by Qi Lei

Qi Lei is on the academic job market for 2021-2022. Baihe Huang will be applying to PhD programs.

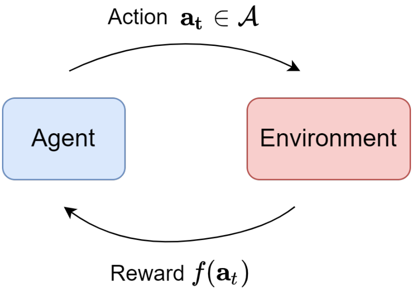
https://arxiv.org/abs/2107.04518

Bandit Problem

An agent interacts with the environment, only receives a scalar reward, and aims to maximize the reward.



Bandit Problem



- Each round, play action from action set: $\boldsymbol{a}_t \in \mathcal{A} \subset \mathbb{R}^d$,
- Unknown reward function f
- Observe the (noisy) reward: $r_t = f(\boldsymbol{a}) + \eta_t$, ($\eta_t$ is mean-zero sub-gaussian noise)
- Goal: maximize reward and minimize regret:
  $R(T) = \sum_{t=1}^{T} r^* - f(\boldsymbol{a}_t)$. $r^* = \max_{\boldsymbol{a}\in\mathcal{A}} f(\boldsymbol{a})$.

Applications

1. Ad placement
2. Recommendation services
3. Network routing
4. Dynamic pricing
5. Resource allocation
6. Necessary step to RL
7. ...

Our focus: beyond linearity and concavity

Motivation

- Linear bandit is well-studied, but doesn't have sufficient representation power
- Existing analysis on nonlinear setting is potentially sub-optimal

Our goal:

- What is the optimal regret for non-concave bandit problems, including structured polynomials (low-rank etc.)?
- Can we design algorithms with optimal dimension dependency?

Our focus:

Structured polynomial bandit

- The stochastic bandit eigenvector case

$$\mathcal{F}_{\mathsf{EV}} = \left\{ \ f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \boldsymbol{a}^T \boldsymbol{M} \boldsymbol{a}, \boldsymbol{M} = \sum_{j=1}^{k} \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\top \ \right\}.$$

Our focus:

Structured polynomial bandit

- The stochastic bandit eigenvector case
- The stochastic low-rank linear reward case

$$\mathcal{F}_{\mathsf{LR}} = \left\{ \; f_{\boldsymbol{\theta}}(\boldsymbol{A}) = \langle \boldsymbol{M}, \boldsymbol{A} \rangle = \mathrm{vec}(\boldsymbol{M})^{\top}\mathrm{vec}(\boldsymbol{A}) \; \right\}.$$

Our focus:

Structured polynomial bandit

- The stochastic bandit eigenvector case
- The stochastic low-rank linear reward case
- The stochastic homogeneous polynomial reward case

  Symmetric:

$$\mathcal{F}_{\mathsf{SYM}} = \left\{ \; f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \textstyle\sum_{j=1}^{k} \lambda_j (\boldsymbol{v}_j^\top \boldsymbol{a})^p \text{ for orthonormal } \boldsymbol{v}_j \; \right\};$$

  Asymmetric:

$$\mathcal{F}_{\mathsf{ASYM}} = \left\{ \begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \sum_{j=1}^{k} \lambda_j \prod_{q=1}^{p} (\boldsymbol{v}_j(q)^\top \boldsymbol{a}(q)), \\ \text{for orthonormal } \boldsymbol{v}_j(q) \text{ for each } q \end{array} \right\}.$$

Our focus:

Structured polynomial bandit

- The stochastic bandit eigenvector case
- The stochastic low-rank linear reward case
- The stochastic homogeneous polynomial reward case
- The noiseless two-layer neural network case

$$\mathcal{F}_{\mathsf{NN}_1} = \left\{ f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \sum_{i=1}^{k} \lambda_i \langle \boldsymbol{v}_i, \boldsymbol{a} \rangle^{p_i}, k \geq \max_i \{p_i\} \right\}.$$

$$\mathcal{F}_{\mathsf{NN}_2} = \left\{ f_{\boldsymbol{\theta}}(\boldsymbol{a}) = q(\boldsymbol{U}\boldsymbol{a}), \boldsymbol{U} \in \mathbb{R}^{k \times d}, \deg q(\cdot) \leq p \right\}.$$

Outline

1. Bandit problem
   - Our focus: beyond linearity and concavity
2. Stochastic Quadratic Reward
   - Stochastic Bandit Eigenvector Problem
   - Stochastic Low-rank linear reward
3. Stochastic high-order homogeneous polynomials
   - Symmetric setting
   - Lower bound
4. Noiseless two-layer neural network
5. Conclusion and Future direction

Problem I: the Stochastic Bandit Eigenvector Problem

- Action set: $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \|\boldsymbol{a}\|_2 \leq 1\}$
- Noisy reward: $r_t = f_{\boldsymbol{\theta}}(\boldsymbol{a}_t) + \eta_t$.

$$f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \boldsymbol{a}^T \boldsymbol{M} \boldsymbol{a}, \boldsymbol{M} = \sum_{j=1}^k \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\top \text{ for orthonormal } \boldsymbol{v}_j, \\ \boldsymbol{M} \in \mathbb{R}^{d \times d}, 1 \geq \lambda_1 \geq |\lambda_2| \geq \cdots \geq |\lambda_k| \quad .$$

- Optimal action $\boldsymbol{a}^* = \pm \boldsymbol{v}_1$.

Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3T})$

Some related work

Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3 T})$
- **Phase retrieval ($k = 1$ case):** lower bound of d^3/ϵ^2 to attain ϵ-optimal solution in the non-adaptive setting (Candes et al. 2015) (Cai et al. 2016)

Some related work

Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3 T})$
- **Phase retrieval ($k = 1$ case):** lower bound of d^3/ϵ^2 to attain ϵ-optimal solution in the non-adaptive setting
- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\widetilde{O}(\sqrt{d_E \log \mathcal{N} \cdot T}) = \widetilde{O}(\sqrt{d^3 k T})$, here covering number $\log \mathcal{N} = \widetilde{O}(dk)$, and eluder dimension $d_E = \widetilde{\Theta}(d^2)$. (e.g. Russo and Van Roy 2013)

Some related work

Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3T})$
- **Phase retrieval ($k = 1$ case):** lower bound of d^3/ϵ^2 to attain ϵ-optimal solution in the non-adaptive setting
- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\widetilde{O}(\sqrt{d_E \log \mathcal{N} \cdot T}) = \widetilde{O}(\sqrt{d^3kT})$, here covering number $\log \mathcal{N} = \widetilde{O}(dk)$, and eluder dimension $d_E = \widetilde{\Theta}(d^2)$. (e.g. Russo and Van Roy 2013)
- **Bandit PCA:** $\sqrt{d^3T}$ regret in the adversarial bandit setting (Kotłowski and Neu 2019)

Some related work

Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3T})$
- **Phase retrieval ($k=1$ case):** lower bound of d^3/ϵ^2 to attain ϵ-optimal solution in the non-adaptive setting
- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\widetilde{O}(\sqrt{d_E \log \mathcal{N} \cdot T}) = \widetilde{O}(\sqrt{d^3kT})$, here covering number $\log \mathcal{N} = \widetilde{O}(dk)$, and eluder dimension $d_E = \widetilde{\Theta}(d^2)$. (e.g. Russo and Van Roy 2013)
- **Bandit PCA:** $\sqrt{d^3T}$ regret in the adversarial bandit setting (Kotłowski and Neu 2019)
- **Summary:** $\sqrt{d^3T}$ is attainable and conjectured to be optimal.

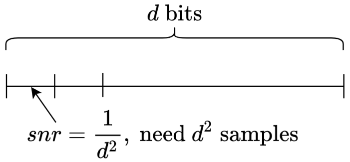
Why the Conjecture?

Intuition of Jun et al. 2019]

Let's first look at the simplest case: $f(\boldsymbol{a}_t) = (\boldsymbol{a}_t^\top \boldsymbol{\theta}^*)^2$ (Bandit phase retrieval)

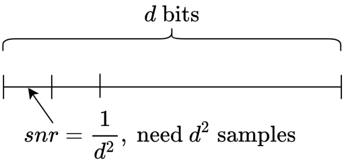
- A random action $\boldsymbol{a} \sim \text{Unif}(\mathbb{S}^{d-1})$ has $f(\boldsymbol{a}) \asymp 1/d$
- Noise has standard deviation $\Omega(1)$
- SNR is $O(1/d^2)$
- $\boldsymbol{\theta}^*$ requires d bits to encode

Conclusion: if we were to play non-adaptively, this would require $O(d^3)$ queries and result in regret $\sqrt{d^3 T}$.

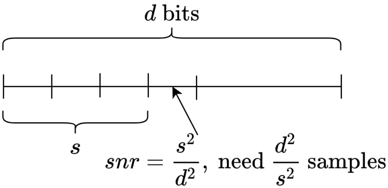


$$\text{Non-adaptive: } \sum_{s=1}^{d} d^2 = d^3$$

Beating d^3

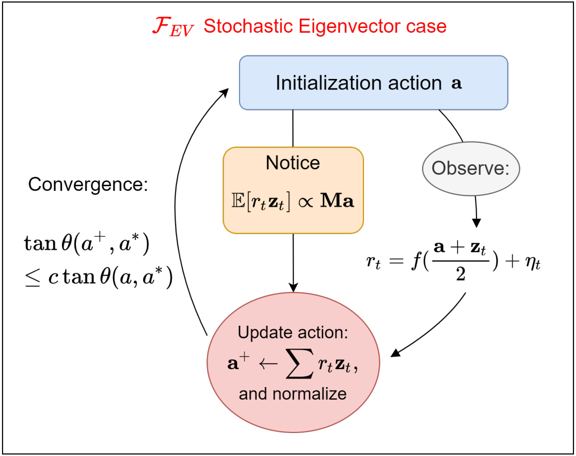


Non-adaptive: $\sum_{s=1}^{d} d^2 = d^3$



Adaptive: $\sum_{s=1}^{d} \frac{d^2}{s^2} \approx d^2$

Our method: noisy power method



$$z_t \sim \mathcal{N}(0, \sigma^2 I).$$

Recall $f(a) = a^\top M a$.

Our method: PAC bound and regret bound

Define $\kappa := \frac{\lambda_1}{\lambda_1 - |\lambda_2|}$.

- Samples per iteration: $\widetilde{O}(d^2\kappa^2/\epsilon^2)$
- Total iterations: $\kappa \log(d/\epsilon)$
- PAC sample complexity: $\widetilde{O}(\kappa^3 d^2/\epsilon^2)$ to make sure $\tan\theta(a, a^*) \leq \epsilon$
- PAC to regret: $\sqrt{\kappa^3 d^2 T}$.

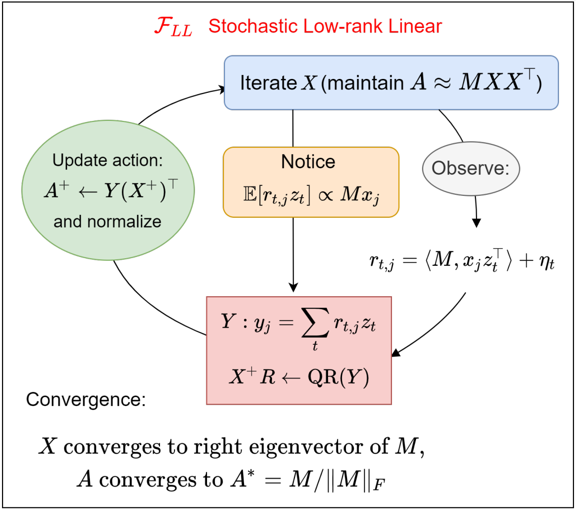
Concurrent work of Lattimore and Hao also show $\sqrt{d^2 T}$ regret in the rank 1 case.

Problem II: Stochastic Low-rank linear reward

- Action set: $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{d\times d} : \|\boldsymbol{M}\|_F \leq 1\}$
- Noisy reward: $r_t = f_{\boldsymbol{\theta}}(\boldsymbol{a}_t) + \eta_t$.

$$\begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{A}) = \langle \boldsymbol{M}, \boldsymbol{A} \rangle = \mathrm{vec}(\boldsymbol{M})^\top \mathrm{vec}(\boldsymbol{A}), \\ \mathsf{rank}(\boldsymbol{M}) = k \end{array} .$$

- Optimal action $\boldsymbol{A}^* = \boldsymbol{M}/\|\boldsymbol{M}\|_F$.



$$\boldsymbol{z}_t \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

$$Y \approx MX, A^+ \approx MXX^\top$$

Regret comparisons: quadratic reward

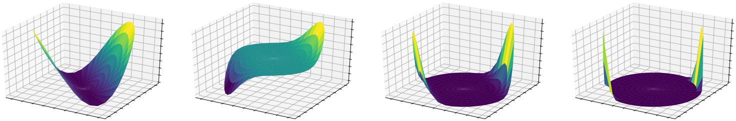
$\mathcal{F}_{\mathsf{EV}}$	LB ($k=1$)	Jun et al, 2019	NPM	Gap-free NPM	Subspace Iteration
Regret	$\sqrt{d^2T}$	$\sqrt{d^3k\lambda_k^{-2}T}$	$\sqrt{\kappa^3d^2T}$	$d^{2/5}T^{4/5}$	$\min(k^{4/3}(dT)^{2/3}, k^{1/3}(\widetilde{\kappa}dT)^{2/3})$

$\mathcal{F}_{\mathsf{LR}}$	LB (Lu et al, 2021)	UB (Lu et al, 2021)	Subspace Iteration
Regret	$\Omega(\sqrt{d^2k^2T})$	$\sqrt{d^3kT}^*$ or $\sqrt{d^3k\lambda_k^{-2}T}$	$\min(\sqrt{d^2k\lambda_k^{-2}T}, (dkT)^{2/3})$

Outline

1. Bandit problem
   - Our focus: beyond linearity and concavity
2. Stochastic Quadratic Reward
   - Stochastic Bandit Eigenvector Problem
   - Stochastic Low-rank linear reward
3. Stochastic high-order homogeneous polynomials
   - Symmetric setting
   - Lower bound
4. Noiseless two-layer neural network
5. Conclusion and Future direction

Higher-order problems



$p = 2$ $p = 5$ $p = 10$ $p = 50$

Signal strength becomes weaker for larger p

Random action $\boldsymbol{a} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$, the average signal strength is: $(\boldsymbol{a}^\top \boldsymbol{a}^*)^p \sim d^{-p/2}$.

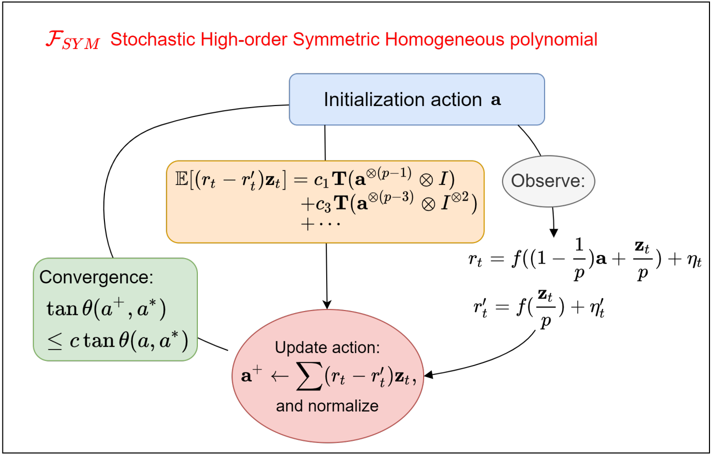
Eluder-UCB incurs $\sqrt{d^{p+1}T}$ regret, which is also what the incorrect heuristic predicts

Problem III: Symmetric High-order Polynomial Bandit

- Action set: $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \|\boldsymbol{a}\|_2 \leq 1\}$
- Noisy reward: $r_t = f_{\boldsymbol{\theta}}(\boldsymbol{a}_t) + \eta_t$.

$$\begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \sum_{j=1}^{k} \lambda_j (\boldsymbol{v}_j^\top \boldsymbol{a})^p, \text{ for orthonormal } \boldsymbol{v}_j, \\ 1 \geq r^* = |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k| \end{array} .$$

- Equivalently $f(\boldsymbol{a}) = \boldsymbol{T}(\boldsymbol{a}^{\otimes p})$, where $\boldsymbol{T} = \sum_{j=1}^{k} \lambda_j \boldsymbol{v}_j^{\otimes p}$
- Optimal action $\boldsymbol{a}^* = \boldsymbol{v}_1$.



$$f(\boldsymbol{a}) = \boldsymbol{T}(\boldsymbol{a}^{\otimes p}).$$

$\boldsymbol{a}^+$ performs multiple tensor product on $\boldsymbol{a}$ with order $p, p-2, \cdots$

Overall Regret Comparisons

Regret			$\mathcal{F}_{\text{SYM}}$	$\mathcal{F}_{\text{ASYM}}$	$\mathcal{F}_{\text{EV}}$	$\mathcal{F}_{\text{LR}}$
LinUCB/eluder			$\sqrt{d^{p+1}kT}$	$\sqrt{d^{p+1}kT}$	$\sqrt{d^3kT}$	$\sqrt{d^3kT}$
Our Results	NPM	Gap	N/A	N/A	$\sqrt{\kappa^3 d^2 T}$	$\sqrt{d^2 k \lambda_k^{-2} T}$
		Gap-free	$\sqrt{d^p kT}$	$\sqrt{k^p d^p T}$	$k^{4/3}(dT)^{2/3}$	$(dkT)^{2/3}$
	Lower Bound		$\sqrt{d^p T}$	$\sqrt{d^p T}$	$\sqrt{d^2 T}$	$\sqrt{d^2 k^2 T}$ [1]

[1] from Lu et al. 2021

Two phases

Tighter Analysis

We can first learn $\boldsymbol{a}$ to constant accuracy via $kd^p/(r^*)^2$ actions and then can use fewer samples per iteration:

$$\widetilde{O}(\frac{kd^p}{r^*} + \sqrt{kd^2T}).$$

- The hardest part is the burn-in to get constant accuracy.
- Once in a region of local strong convexity, linear convergence ensures good regret.

Minimax regret lower bound

For all adaptive algorithms:

- Symmetric action set: $R(T) \geq \Omega(\sqrt{d^p T}/p^p)$

Lower bound: Optimal dependence on d

Minimax regret lower bound

For all adaptive algorithms:

- Symmetric action set: $R(T) \geq \Omega(\sqrt{d^p T}/p^p)$
- Asymmetric action set: $R(T) \geq \Omega(\sqrt{d^p T})$

Lower bound: Optimal dependence on d

Minimax regret lower bound

For all adaptive algorithms:

- Symmetric action set: $R(T) \geq \Omega(\sqrt{d^p T}/p^p)$
- Asymmetric action set: $R(T) \geq \Omega(\sqrt{d^p T})$

Optimality on burn-in phase

For all adaptive algorithms, we need at least $\Omega(\frac{d^p}{(r^*)^2})$ actions to get reward at least constant of the optimal reward r^*.

Outline

1. Bandit problem
   - Our focus: beyond linearity and concavity
2. Stochastic Quadratic Reward
   - Stochastic Bandit Eigenvector Problem
   - Stochastic Low-rank linear reward
3. Stochastic high-order homogeneous polynomials
   - Symmetric setting
   - Lower bound
4. Noiseless two-layer neural network
5. Conclusion and Future direction

Problem V: Noiseless two-layer neural network reward

Upper bound via solving polynomial equations

- $f(\boldsymbol{a}) = \sum_{i=1}^{k} \lambda_i \langle \boldsymbol{v}_i, \boldsymbol{a} \rangle^{p_i}$, $k \geq \max_i \{p_i\}$:

$$R(T) \lesssim \min\{T, dk\}$$

- $f(\boldsymbol{a}) = q(\boldsymbol{U}\boldsymbol{a}), \boldsymbol{U} \in \mathbb{R}^{k \times d}, \deg q(\cdot) \leq p$:

$$R(T) \lesssim \min\{T, dk + (k+1)^p\}.$$

However, we can construct action sets where any UCB algorithm

$$R(T) \geq \min\left\{T, \binom{d}{p}\right\}.$$

Extension to RL in simulator setting

$$\mathcal{T}_h(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[\max_{a'} Q_{h+1}(s',a')].$$

Extension to RL in simulator setting

$$\mathcal{T}_h(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[\max_{a'} Q_{h+1}(s',a')].$$

Settings:

- Assume $\mathcal{F}_{EV} = \{f_{\boldsymbol{M}}(s,a) = \phi(s,a)^\top \boldsymbol{M} \phi(s,a),$ rank$(M) \leq k\}$ is Bellman complete
- Observation: we query s_{h-1}, a_{h-1}, we observe $s'_h \sim \mathbb{P}(\cdot|s_{h-1}, a_{h-1})$ and reward $r_{h-1}(s_{h-1}, a_{h-1})$.

Extension to RL in simulator setting

$$\mathcal{T}_h(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[\max_{a'} Q_{h+1}(s',a')].$$

Settings:

- Assume $\mathcal{F}_{EV} = \{f_{\boldsymbol{M}}(s,a) = \phi(s,a)^\top \boldsymbol{M} \phi(s,a),$ $\text{rank}(M) \le k\}$ is Bellman complete
- Observation: we query s_{h-1}, a_{h-1}, we observe $s'_h \sim \mathbb{P}(\cdot|s_{h-1}, a_{h-1})$ and reward $r_{h-1}(s_{h-1}, a_{h-1})$.

Extend our findings from bandit:

- We can estimate $\widehat{\boldsymbol{M}}_h, h = H, H-1, \cdots 1$ up to ϵ/H error with $\widetilde{O}(d^2k^2H^2/\epsilon^2)$ samples
- Overall we can learn ϵ-optimal policy π with $\widetilde{O}(d^2k^2H^3/\epsilon^2)$ samples

Extension to RL in simulator setting

$$\mathcal{T}_h(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[\max_{a'} Q_{h+1}(s',a')].$$

Settings:

- Assume $\mathcal{F}_{EV} = \{f_{\boldsymbol{M}}(s,a) = \phi(s,a)^\top \boldsymbol{M}\phi(s,a),$ $\text{rank}(M) \leq k\}$ is Bellman complete
- Observation: we query s_{h-1}, a_{h-1}, we observe $s'_h \sim \mathbb{P}(\cdot|s_{h-1}, a_{h-1})$ and reward $r_{h-1}(s_{h-1}, a_{h-1})$.

Extend our findings from bandit:

- We can estimate $\widehat{\boldsymbol{M}}_h, h = H, H-1, \cdots 1$ up to ϵ/H error with $\widetilde{O}(d^2k^2H^2/\epsilon^2)$ samples
- Overall we can learn ϵ-optimal policy π with $\widetilde{O}(d^2k^2H^3/\epsilon^2)$ samples

In contrast, optimistic algorithm requires $O(d^3H^3/\epsilon^2)$ samples (or $O(d^3H^2/\epsilon^2)$ trajectories) (Zanette et al. 2020, Jin et al. 2021)

Outline

1. Bandit problem
   - Our focus: beyond linearity and concavity
2. Stochastic Quadratic Reward
   - Stochastic Bandit Eigenvector Problem
   - Stochastic Low-rank linear reward
3. Stochastic high-order homogeneous polynomials
   - Symmetric setting
   - Lower bound
4. Noiseless two-layer neural network
5. Conclusion and Future direction

Conclusions

We find optimal regret for different types of reward function classes:

- the stochastic bandit eigenvector case
- the stochastic low-rank linear reward case
- the stochastic homogeneous polynomial reward case
- the noiseless neural network with polynomial activation

Take-away messages

- Optimistic algorithms have suboptimal regret $\Rightarrow$ allow to play suboptimally sometimes

Conclusions

We find optimal regret for different types of reward function classes:

- the stochastic bandit eigenvector case
- the stochastic low-rank linear reward case
- the stochastic homogeneous polynomial reward case
- the noiseless neural network with polynomial activation

Take-away messages

- Optimistic algorithms have suboptimal regret $\Rightarrow$ allow to play suboptimally sometimes
- Initial snr is already $1/d^p$ $\Rightarrow$ with (super)linear convergence rate, can hope to get optimal dependence on d

Conclusions

We find optimal regret for different types of reward function classes:

- the stochastic bandit eigenvector case
- the stochastic low-rank linear reward case
- the stochastic homogeneous polynomial reward case
- the noiseless neural network with polynomial activation

Take-away messages

- Optimistic algorithms have suboptimal regret $\Rightarrow$ allow to play suboptimally sometimes
- Initial snr is already $1/d^p$ $\Rightarrow$ with (super)linear convergence rate, can hope to get optimal dependence on d
- Initial phase is the hardest $\Rightarrow$ play adaptively and consider burn-in algorithms

Conclusions

We find optimal regret for different types of reward function classes:

- the stochastic bandit eigenvector case
- the stochastic low-rank linear reward case
- the stochastic homogeneous polynomial reward case
- the noiseless neural network with polynomial activation

Take-away messages

- Optimistic algorithms have suboptimal regret $\Rightarrow$ allow to play suboptimally sometimes
- Initial snr is already $1/d^p$ $\Rightarrow$ with (super)linear convergence rate, can hope to get optimal dependence on d
- Initial phase is the hardest $\Rightarrow$ play adaptively and consider burn-in algorithms
- Strongly convex action set $\Rightarrow$ Still have $\sqrt{T}$ PAC to regret conversion with explore-then-commit

Future directions

Future directions

- Settle whether the condition number dependence is necessary

Future directions

Future directions

- Settle whether the condition number dependence is necessary
- Non-orthogonal high-order polynomials?

Future directions

Future directions

- Settle whether the condition number dependence is necessary
- Non-orthogonal high-order polynomials?
- Discrete action set?

Future directions

Future directions

- Settle whether the condition number dependence is necessary
- Non-orthogonal high-order polynomials?
- Discrete action set?
- Extension multi-task representation learning for bandits or MDPs

Thank you!