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
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: $a_t \in \mathcal{A} \subset \mathbb{R}^d$, 

Unknown reward function $f$

Observe the (noisy) reward: $r_t = f(a) + \eta_t$, ($\eta_t$ is mean-zero sub-gaussian noise)

Goal: maximize reward and minimize regret:

$R(T) = \sum_{t=1}^{T} r^* - f(a_t)$. $r^* = \max_{a \in \mathcal{A}} f(a)$. 

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

\[
\mathcal{F}_{\text{EV}} = \left\{ f_\theta(a) = a^T M a, \quad M = \sum_{j=1}^{k} \lambda_j v_j v_j^T \right\}.
\]
Our focus:

Structured polynomial bandit

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

\[ \mathcal{F}_{LR} = \{ f_\theta(A) = \langle M, A \rangle = \text{vec}(M)^\top \text{vec}(A) \} . \]

Structured polynomial bandit
Our focus:

Structured polynomial bandit

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

Symmetric:

\[ F_{\text{SYM}} = \left\{ f_\theta(a) = \sum_{j=1}^{k} \lambda_j (v_j^\top a)^p \text{ for orthonormal } v_j \right\}; \]

Asymmetric:

\[ F_{\text{ASYM}} = \left\{ f_\theta(a) = \sum_{j=1}^{k} \lambda_j \prod_{q=1}^{p} (v_j(q)^\top a(q)), \right\} \text{ for orthonormal } v_j(q) \text{ for each } q. \]
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

\[ \mathcal{F}_{\text{NN}_1} = \left\{ f_{\theta}(a) = \sum_{i=1}^{k} \lambda_i \langle v_i, a \rangle^{p_i}, k \geq \max_{i} \{ p_i \} \right\}. \]

\[ \mathcal{F}_{\text{NN}_2} = \left\{ f_{\theta}(a) = q(Ua), U \in \mathbb{R}^{k \times d}, \deg q(\cdot) \leq p \right\}. \]
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: \( A = \{ a \in \mathbb{R}^d : \| a \|_2 \leq 1 \} \)
- Noisy reward: \( r_t = f_\theta(a_t) + \eta_t \).

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

- Optimal action \( a^* = \pm v_1 \).
## Prior Conjectures and Adapting Existing Work

- Jun et al. 2019 conjecture the regret for bandit eigenvector is at least $\Omega(\sqrt{d^3T})$.
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 $d^3 / \epsilon^2$ to attain $\epsilon$-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 $d^3 / \epsilon^2$ to attain $\epsilon$-optimal solution in the non-adaptive setting

- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\tilde{O}(\sqrt{d_E \log N \cdot T}) = \tilde{O}(\sqrt{d^3 kT})$, here covering number $\log N = O(dk)$, and eluder dimension $d_E = \Theta(d^2)$. (e.g. Russo and Van Roy 2013)
## 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 $d^3 / \epsilon^2$ to attain $\epsilon$-optimal solution in the non-adaptive setting

- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\tilde{O}(\sqrt{d_E \log N \cdot T}) = \tilde{O}(\sqrt{d^3 k T})$, here covering number $\log N = \tilde{O}(d k)$, and eluder dimension $d_E = \Theta(d^2)$. (e.g. Russo and Van Roy 2013)

- **Bandit PCA:** $\sqrt{d^3 T}$ regret in the adversarial bandit setting (Kotłowski and Neu 2019)
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 $d^3/\epsilon^2$ to attain $\epsilon$-optimal solution in the non-adaptive setting

- **Eluder dimension:** With EluderUCB algorithm, one can achieve regret of $\widetilde{O}(\sqrt{d_E \log N \cdot T}) = \widetilde{O}(\sqrt{d^3kT})$, here covering number $\log N =\widetilde{O}(dk)$, and eluder dimension $d_E = \Theta(d^2)$. (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: \( f(a_t) = (a_t^\top \theta^*)^2 \) (Bandit phase retrieval)

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

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

**Non-adaptive:**

\[
\sum_{s=1}^{d} d^2 = d^3
\]

\[
\text{snr} = \frac{1}{d^2}, \text{ need } d^2 \text{ samples}
\]

**Adaptive:**

\[
\sum_{s=1}^{d} \frac{d^2}{s^2} \approx d^2
\]

\[
\text{snr} = \frac{s^2}{d^2}, \text{ need } \frac{d^2}{s^2} \text{ samples}
\]
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: $\tilde{O}(d^2 \kappa^2 / \epsilon^2)$
- Total iterations: $\kappa \log(d/\epsilon)$
- PAC sample complexity: $\tilde{O}(\kappa^3 d^2 / \epsilon^2)$ 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: \( \mathcal{A} = \{ A \in \mathbb{R}^{d \times d} : \| M \|_F \leq 1 \} \)
- Noisy reward: \( r_t = f_\theta(a_t) + \eta_t \).

\[
f_\theta(A) = \langle M, A \rangle = \text{vec}(M)^\top \text{vec}(A), \quad \text{rank}(M) = k.
\]

- Optimal action \( A^* = M / \| M \|_F \).
Our algorithm: noisy subspace iteration

\( F_{LL} \)  Stochastic Low-rank Linear

Iterate \( X \) (maintain \( A \approx MXX^T \))

Update action:
\[
A^+ \leftarrow Y(X^+)^\top
\]
and normalize

Notice
\[
\mathbb{E}[r_{t,j}z_t] \propto Mx_j
\]

Observe:
\[
r_{t,j} = \langle M, x_jz_t^\top \rangle + \eta_t
\]

\[
Y : y_j = \sum_t r_{t,j}z_t
\]

Convergence:
\[
X \text{ converges to right eigenvector of } M,
A \text{ converges to } A^* = M/\|M\|_F
\]

\[
z_t \sim \mathcal{N}(0, \sigma^2 I)
\]

\[
Y \approx MX, A^+ \approx MXX^T
\]
Regret comparisons: quadratic reward

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{F}_{EV}$</th>
<th>$\mathcal{F}_{LR}$</th>
<th>Jun et al, 2019</th>
<th>NPM</th>
<th>Gap-free NPM</th>
<th>Subspace Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regret</td>
<td>$\sqrt{d^2 T}$</td>
<td>$\Omega(\sqrt{d^2 k^2 T})$</td>
<td>$\sqrt{d^3 k^2 \lambda_k^{-2} T}$</td>
<td>$\sqrt{\kappa^3 d^2 T}$</td>
<td>$d^{2/5} T^{4/5}$</td>
<td>$\min(k^{4/3} (dT)^{2/3}, k^{1/3} (\kappa dT)^{2/3})$</td>
</tr>
<tr>
<td>Regret</td>
<td>$\sqrt{d^2 T}$</td>
<td>$\sqrt{d^3 k^2 \lambda_k^{-2} T}$</td>
<td>$\sqrt{d^3 k^2 \lambda_k^{-2} T}$</td>
<td>$\sqrt{d^3 k^2 \lambda_k^{-2} T}$</td>
<td>$\min(\sqrt{d^2 k \lambda_k^{-2} T}, (dkT)^{2/3})$</td>
<td>$\min(\sqrt{d^2 k \lambda_k^{-2} T}, (dkT)^{2/3})$</td>
</tr>
</tbody>
</table>
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 $a \sim \text{Unif}(S^{d-1})$, the average signal strength is: $(a^\top a^*)^p \sim d^{-p/2}$.

Eluder-UCB incurs $\sqrt{dp+1}T$ regret, which is also what the incorrect heuristic predicts.
Problem III: Symmetric High-order Polynomial Bandit

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

\[
f_\theta(\mathbf{a}) = \sum_{j=1}^{k} \lambda_j (\mathbf{v}_j^\top \mathbf{a})^p, \text{ for orthonormal } \mathbf{v}_j,
\]

\[
1 \geq r^* = |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|
\]

- Equivalently \( f(\mathbf{a}) = T(\mathbf{a}^\otimes p) \), where \( T = \sum_{j=1}^{k} \lambda_j \mathbf{v}_j^\otimes p \)
- Optimal action \( \mathbf{a}^* = \mathbf{v}_1 \).
Algorithm: Zeroth order gradient-like ascent

\[ f(a) = T(a^\otimes p). \]

\[ a^+ \] performs multiple tensor product on \( a \) with order \( p, p - 2, \cdots \)
## Overall Regret Comparisons

<table>
<thead>
<tr>
<th>Regret</th>
<th>$\mathcal{F}_{\text{SYM}}$</th>
<th>$\mathcal{F}_{\text{ASYM}}$</th>
<th>$\mathcal{F}_{\text{EV}}$</th>
<th>$\mathcal{F}_{\text{LR}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinUCB/eluder</td>
<td>$\sqrt{dp+1}kT$</td>
<td>$\sqrt{dp+1}kT$</td>
<td>$\sqrt{d^3kT}$</td>
<td>$\sqrt{d^3kT}$</td>
</tr>
<tr>
<td>Our Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPM</td>
<td>N/A</td>
<td>N/A</td>
<td>$\sqrt{\kappa^3d^2T}$</td>
<td>$\sqrt{d^2k\lambda_k^{-2}T}$</td>
</tr>
<tr>
<td>Gap-free</td>
<td>$\sqrt{dpkT}$</td>
<td>$\sqrt{k^pdpT}$</td>
<td>$k^{4/3}(dT)^{2/3}$</td>
<td>$(dkT)^{2/3}$</td>
</tr>
<tr>
<td>Lower Bound</td>
<td>$\sqrt{dpT}$</td>
<td>$\sqrt{dpT}$</td>
<td>$\sqrt{d^2T}$</td>
<td>$\sqrt{d^2k^2T}$</td>
</tr>
</tbody>
</table>

\(^1\)from Lu et al. 2021
Two phases

Tighter Analysis

We can first learn $\alpha$ to constant accuracy via $k d^p / (r^*)^2$ actions and then can use fewer samples per iteration:

$$\tilde{O}\left(\frac{k d^p}{r^*} + \sqrt{k d^2 T}\right).$$

- The hardest part is the burn-in to get constant accuracy.
- Once in a region of local strong convexity, linear convergence ensures good regret.
Lower bound: Optimal dependence on $d$

Minimax regret lower bound

For all adaptive algorithms:

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

Optimality on burn-in phase

For all adaptive algorithms, we need at least $\Omega\left(\frac{dp}{(r^*)^2}\right)$ 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

- \( f(a) = \sum_{i=1}^{k} \lambda_i \langle v_i, a \rangle^{p_i}, \quad k \geq \max_i \{p_i\} : \)

\[
R(T) \lesssim \min\{T, dk\}
\]

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

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

However, we can construct action sets where any \( UCB \) algorithm

\[
R(T) \geq \min \left\{ T, \binom{d}{p} \right\}.
\]
$\mathcal{T}_h(Q_{h+1})(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim \mathbb{P}(s')|s, a} \left[ \max_{a'} Q_{h+1}(s', a') \right]$. 
\[ T_h(Q_{h+1})(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim P(\cdot|s, a)}[\max_{a'} Q_{h+1}(s', a')] \]

Settings:

- Assume \( F_{EV} = \{ f_M(s, a) = \phi(s, a)^\top M \phi(s, a), \text{rank}(M) \leq k \} \) is Bellman complete
- Observation: we query \( s_{h-1}, a_{h-1} \), we observe \( s_{h}' \sim P(\cdot|s_{h-1}, a_{h-1}) \) and reward \( r_{h-1}(s_{h-1}, a_{h-1}) \).
\[ T_h(Q_{h+1})(s, a) = r_h(s, a) + \mathbb{E}_{s' \sim \mathbb{P}(\cdot | s, a)}[\max_{a'} Q_{h+1}(s', a')]. \]

Settings:

- Assume \( \mathcal{F}_{EV} = \{ f_M(s, a) = \phi(s, a) \top M \phi(s, a), \text{rank}(M) \leq k \} \) 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 \( \hat{M}_h, h = H, H - 1, \cdots, 1 \) up to \( \epsilon/H \) error with \( \widetilde{O}(d^2k^2H^2/\epsilon^2) \) samples
- Overall we can learn \( \epsilon \)-optimal policy \( \pi \) with \( \widetilde{O}(d^2k^2H^3/\epsilon^2) \) samples
Extension to RL in simulator setting

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

Settings:

- Assume \( \mathcal{F}_{EV} = \{ f_M(s, a) = \phi(s, a)^\top M \phi(s, a), \text{ rank}(M) \leq k \} \) is Bellman complete

- Observation: we query \( s_{h-1}, a_{h-1} \), we observe \( s'_h \sim P(\cdot|s_{h-1}, a_{h-1}) \) and reward \( r_{h-1}(s_{h-1}, a_{h-1}) \).

Extend our findings from bandit:

- We can estimate \( \widehat{M}_h, h = H, H - 1, \cdots 1 \) up to \( \epsilon/H \) error with \( \widetilde{O}(d^2 k^2 H^2/\epsilon^2) \) samples

- Overall we can learn \( \epsilon \)-optimal policy \( \pi \) with \( \widetilde{O}(d^2 k^2 H^3/\epsilon^2) \) samples

In contrast, optimistic algorithm requires \( O(d^3 H^3/\epsilon^2) \) samples (or \( O(d^3 H^2/\epsilon^2) \) 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 ⇒ 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 ⇒ allow to play suboptimally sometimes
- Initial snr is already $1/d^p$ ⇒ with (super)linear convergence rate, can hope to get optimal dependence on $d$
- Initial phase is the hardest ⇒ 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

- Settle whether the condition number dependence is necessary
Future directions

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

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