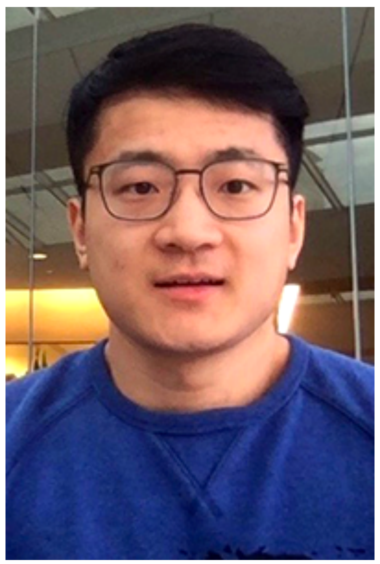
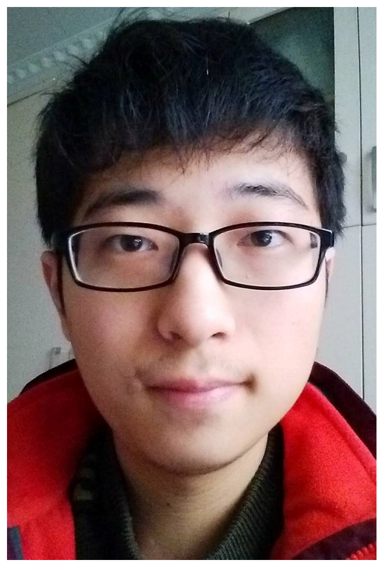
Pure Exploration in Kernel and Neural Bandits



Yinglun Zhu[1*], Dongruo Zhou[2*], Ruoxi Jiang[3*],
Quanquan Gu[2], Rebecca Willett[3] and Robert Nowak[1]

[1]University of Wisconsin-Madison
[2]University of California, Los Angeles
[3]University of Chicago

[*]Equal contribution

The Pure Exploration Problem

- Given action/arm set $\mathcal{X} \subseteq \mathbb{R}^D$.
- At each round, the learner
  - pulls arm $x \in \mathcal{X}$ based on past observations;
  - receives noisy feedback $r(x) = h(x) + \xi$.
- Goal: identify an ϵ-optimal arm $\widehat{x}$, i.e.,

$$h(\widehat{x}) \geq \max_{x} h(x) - \epsilon,$$

  with probability at least $1 - \delta$.
- Performance measure: sample complexity.

Applications

Drug discovery

Crowd-sourcing

Simulation-based planning

Overview of Results

Classical settings:

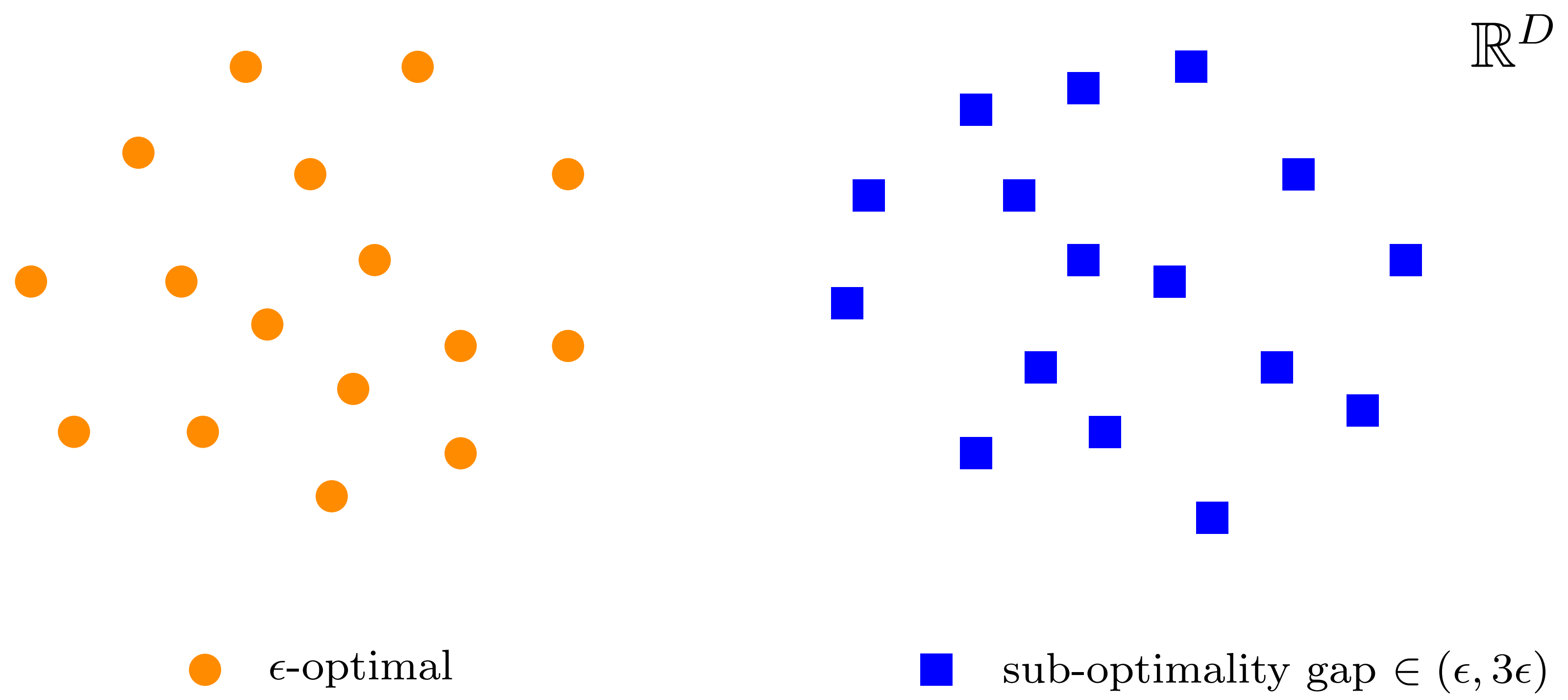
- Standard MAB: no relations between $h(x)$ and $h(x')$.
- Linear bandits: $h(x) = \langle \theta^\star, x \rangle$ with unknown $\theta^\star \in \mathbb{R}^D$.

Our results:

- Study the high-dimensional linear bandit setting.
- Kernel bandits: h belongs to the RKHS $\mathcal{H}$ induced by the kernel function $\mathcal{K}$.[1]
- Neural bandits: learn a general non-linear h with a neural network.

[1] Camilleri et al. 2021 also study pure exploration in kernel bandits.

A Motivating Example



Example: A high-dimensional linear bandit problem, where each circle/square represents an arm in $\mathbb{R}^D$.

Standard approach: sample complexity scales as $\Omega(D/\epsilon^2)$.

Our insights:

- embed arms into $\mathbb{R}^2$;
- carefully deal with the induced misspecification;
- identify an ϵ-optimal arm with $\widetilde{O}(1/\epsilon^2)$ samples.

Key Ideas

The embedding: feature mapping $\psi_d : \mathcal{X} \to \mathbb{R}^d$ such that there exists $\theta_d \in \mathbb{R}^d$ satisfying

$$\max_{x \in \mathcal{X}} |h(x) - \langle \psi_d(x), \theta_d \rangle| \le \widetilde{\gamma}(d).$$

The working dimension: select the smallest embedding dimension d_k so that the induced misspecification is well-controlled.

The Algorithm

Algorithm 1 Algorithmic Framework

1: Set $n = O(\log(1/\epsilon))$.
2: **for** $k = 1, 2, \ldots, n$ **do**
3: Set d_k be the smallest dimension so that the induced mis-specification $< O(2^{-k})$.
4: Eliminate arms (wrt ψ_{d_k}) with sub-optimality gaps $> O(2^{-k})$.
5: **end for**

Adaptive dimension selection: the embedding dimension d_k is allowed to change from round to round.

Theoretical Gaurantees

Theorem: Our algorithm identifies an ϵ-optimal arm with $\widetilde{O}(d_{\texttt{eff}}(\epsilon)/\epsilon^2)$ samples.

Kernel bandits:

$d_{\texttt{eff}}(\epsilon) = O(\epsilon^{-2/(2\beta-3)})$ with polynomial eigen-decay at rate characterized by the constant β;

$d_{\texttt{eff}}(\epsilon) = O(\log(1/\epsilon))$ with exponential eigen-decay.

Neural bandits:

$$d_{\texttt{eff}}(\epsilon) = \min_d \left\{ \sum_{i=d+1}^{|\mathcal{X}|} \lambda_i(\boldsymbol{H}) \leq \text{poly}(\epsilon) \right\},$$

where $\boldsymbol{H}$ is the Neural Tangent Kernel (NTK) matrix wrt $\mathcal{X}$.

The Kernel Case

Recall: $h \in \mathcal{H}$ where $\mathcal{H}$ is the RKHS induced by $\mathcal{K}$.

Mercer's Theorem and Corollary: Let $\{\phi_i\}_{i=1}^{\infty}$ and $\{\mu_i\}_{i=1}^{\infty}$ be the sequence of eigenfunctions and eigenvalues associated with kernel $\mathcal{K}$. Any $h \in \mathcal{H}$ can be written as $h = \sum_{i=1}^{\infty} \theta_i \phi_i$ for some $\{\theta_i\}_{i=1}^{\infty} \in \ell^2(\mathbb{N})$ such that $\sum_{i=1}^{\infty} \theta_i^2/\mu_i < \infty$.

Feature mapping and approximation error: One can construct

$$\boldsymbol{\psi}_d(\boldsymbol{x}) = \left[\sqrt{\mu_1}\phi_1(\boldsymbol{x}), \ldots, \sqrt{\mu_d}\phi_d(\boldsymbol{x})\right]^{\top} \in \mathbb{R}^d$$

so that $\widetilde{\gamma}(d) \leq C \sum_{j>d} \sqrt{\mu_j}$.

The Neural Case

Neural network approximation: Let $f(\boldsymbol{x};\boldsymbol{\theta})$ denote a randomly initialized neural network whose width m is large enough.
At each iteration:

- Train neural network wrt $\{(x_i, y_i)\}$ and get $\widehat{\boldsymbol{\theta}}$.
- Denote $\boldsymbol{g}(x;\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta})$, it can be shown that

$$h(\boldsymbol{x}) \approx \langle \boldsymbol{g}(x;\widehat{\boldsymbol{\theta}}), \boldsymbol{\theta}^\star \rangle.$$

Feature mapping and approximation error: Construct

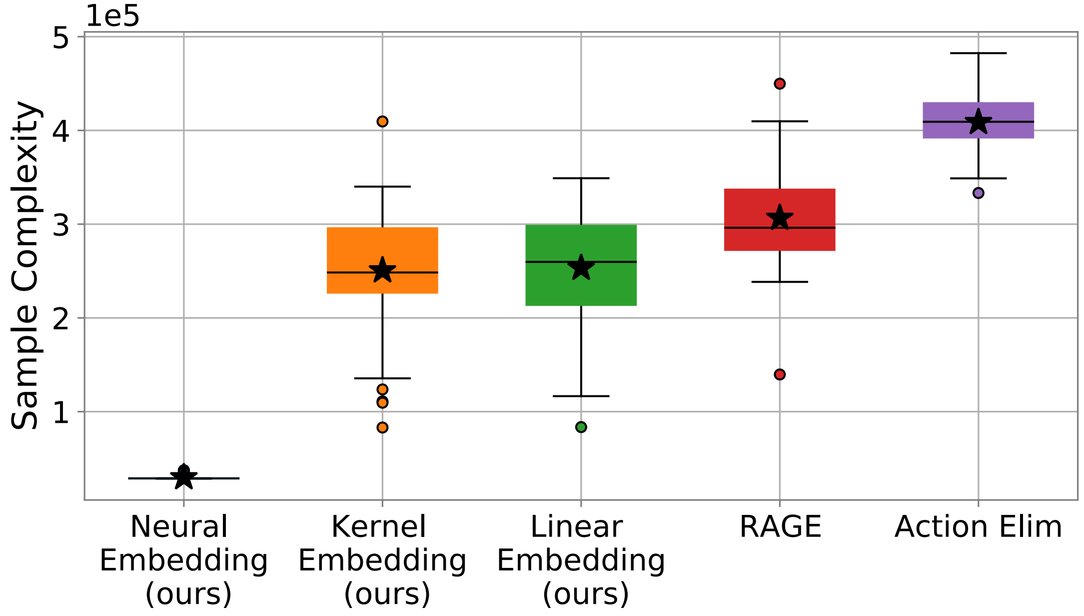
$$\boldsymbol{G} = [\boldsymbol{g}(\boldsymbol{x}_1;\widehat{\boldsymbol{\theta}})^\top; \ldots; \boldsymbol{g}(\boldsymbol{x}_{|\mathcal{X}|};\widehat{\boldsymbol{\theta}})^\top]/\sqrt{m}.$$

Let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} = \boldsymbol{G}$, we set the feature mapping as

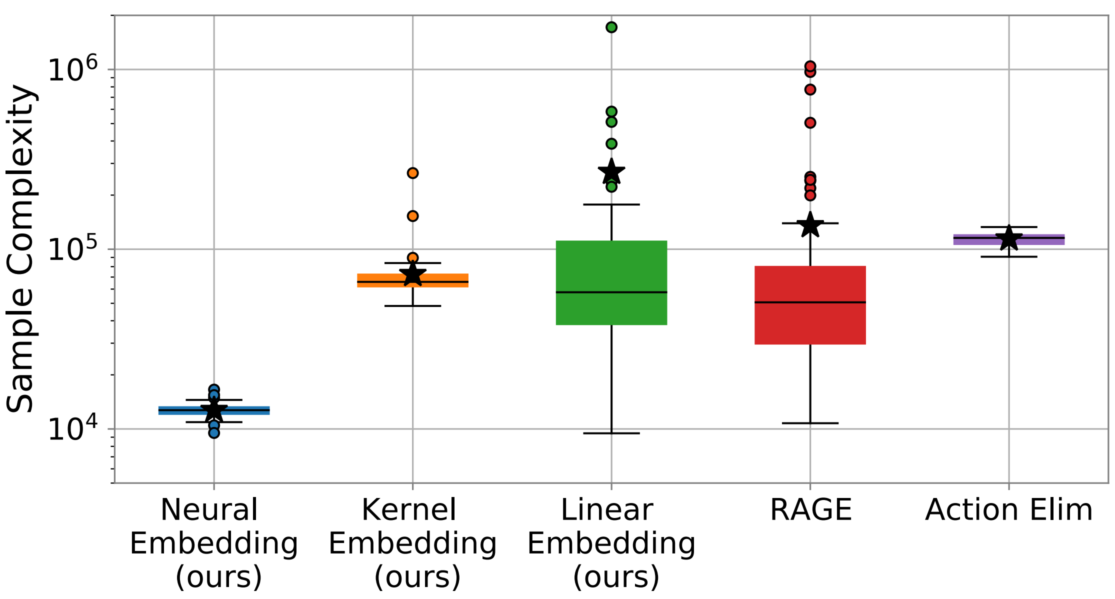
$$\psi_d(\boldsymbol{x}_i) = [\sigma_1 u_{i1}, \ldots, \sigma_d u_{id}]^\top;$$

the approximation error can be characterized by tail singular values.

Empirical Performance



(a) MNIST



(b) Yahoo

Empirical evaluations: The box is drawn from the first quartile to the third quartile; the mean sample complexity is marked as the black star.

Thank you!