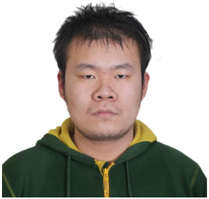
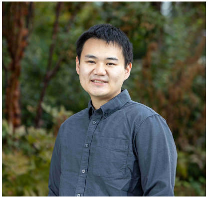
Variance-Aware Sparse Linear Bandits

(Published as a conference paper at ICLR 2023)

Yan Dai [1] Ruosong Wang [2] Simon S. Du [2]

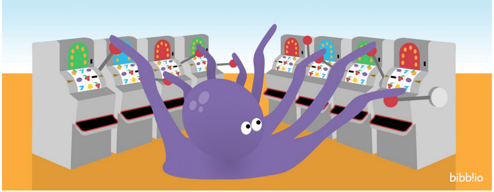




[1]IIIS, Tsinghua University [2]Paul G. Allen School, University of Washington

Table of Contents

Linear Bandit

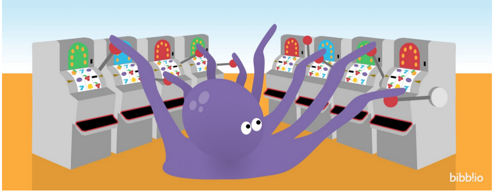
- A T-round game between an **agent** and the **environment**.[1]



[1] Figure from *Reinforcement Learning – Multi-Arm Bandit Implementation*, Jeremy Zhang.

Linear Bandit

- A T-round game between an **agent** and the **environment**.[1]

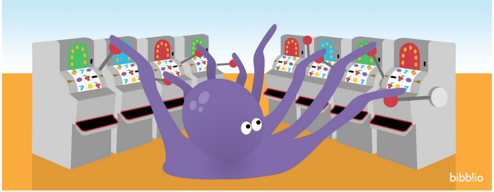


- For each round $t = 1, 2, \ldots, T$, the agent plays an **action** a_t from the unit sphere $\mathbb{S}^{d-1}$ (our assumption).

[1] Figure from *Reinforcement Learning – Multi-Arm Bandit Implementation*, Jeremy Zhang.

Linear Bandit

- A T-round game between an **agent** and the **environment**.[1]

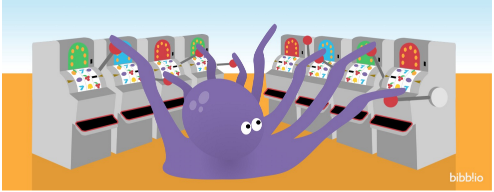


- For each round $t = 1, 2, \ldots, T$, the agent plays an **action** a_t from the unit sphere $\mathbb{S}^{d-1}$ (our assumption).
- For this round, she gains **reward** $r(a_t) = \langle a_t, \theta^* \rangle$ where $\theta^* \in \mathbb{S}^{d-1}$ is a *fixed but unknown* parameter.

[1] Figure from *Reinforcement Learning – Multi-Arm Bandit Implementation*, Jeremy Zhang.

Linear Bandit

- A T-round game between an **agent** and the **environment**.[1]



- For each round $t = 1, 2, \ldots, T$, the agent plays an **action** a_t from the unit sphere $\mathbb{S}^{d-1}$ (our assumption).
- For this round, she gains **reward** $r(a_t) = \langle a_t, \theta^* \rangle$ where $\theta^* \in \mathbb{S}^{d-1}$ is a *fixed but unknown* parameter.
- She cannot direcly access $r(a_t)$, but only observes noisy feedback $r(a_t) + \eta_t$ where η_t is a zero-mean *random* **noise**. Typically assume $\mathrm{Var}(\eta_t) \leq 1$ for all t.

[1] Figure from *Reinforcement Learning – Multi-Arm Bandit Implementation*, Jeremy Zhang.

Agent's Goal?

Maximize the (expected) total reward

$$\mathbb{E}\left[\sum_{t=1}^{T} r(a_t)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle a_t, \theta^* \rangle\right],$$

Agent's Goal?

Maximize the (expected) total reward

$$\mathbb{E}\left[\sum_{t=1}^{T} r(a_t)\right] = \mathbb{E}\left[\sum_{t=1}^{T}\langle a_t, \theta^*\rangle\right],$$

or equivalently, minimize the **regret**

$$\begin{aligned}\mathcal{R}_T &\triangleq \max_{a\in\mathbb{S}^{d-1}} \mathbb{E}\left[\sum_{t=1}^{T}\langle a - a_t, \theta^*\rangle\right].\\ &= \mathbb{E}\left[\sum_{t=1}^{T}\langle \theta^* - a_t, \theta^*\rangle\right].\end{aligned}$$

Sparse Linear Bandit

θ^* is guaranteed to have only a few non-zero coordinates, i.e., $s \triangleq \|\theta^*\|_0$ satisfies $s \ll d$. However, s is *unknown* to the agent.

Sparse Linear Bandit

θ^* is guaranteed to have only a few non-zero coordinates, i.e., $s \triangleq \|\theta^*\|_0$ satisfies $s \ll d$. However, s is *unknown* to the agent.

Known Results:

- **Upper Bound**: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
- **Lower Bound**: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009] even when sparsity factor $s = 1$ and the action set is $\mathbb{S}^{d-1}$.

Sparse Linear Bandit

θ^* is guaranteed to have only a few non-zero coordinates, i.e., $s \triangleq \|\theta^*\|_0$ satisfies $s \ll d$. However, s is *unknown* to the agent.

Known Results:

- **Upper Bound**: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
- **Lower Bound**: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009] even when sparsity factor $s = 1$ and the action set is $\mathbb{S}^{d-1}$.
- **Conclusion:** $\widetilde{\mathcal{O}}(\sqrt{sdT})$ is minimax optimal for SLB.

Variance-Aware Sparse Linear Bandit?

The noises $\{\eta_t\}_{t=1}^T$ have time-dependent variances. Formally, $\eta_t \sim \mathcal{N}(0, \sigma_t^2)$ where $\sigma_t \in [0, 1]$ varies with time (and is hidden).

Variance-Aware Sparse Linear Bandit?

The noises $\{\eta_t\}_{t=1}^T$ have time-dependent variances. Formally, $\eta_t \sim \mathcal{N}(0, \sigma_t^2)$ where $\sigma_t \in [0, 1]$ varies with time (and is hidden).

- **Worst Case** ($\sigma_t \equiv 1$): $\widetilde{\mathcal{O}}(\sqrt{sdT})$ is known to be optimal.

Variance-Aware Sparse Linear Bandit?

The noises $\{\eta_t\}_{t=1}^T$ have time-dependent variances. Formally, $\eta_t \sim \mathcal{N}(0, \sigma_t^2)$ where $\sigma_t \in [0, 1]$ varies with time (and is hidden).

- **Worst Case** ($\sigma_t \equiv 1$): $\widetilde{\mathcal{O}}(\sqrt{sdT})$ is known to be optimal.
- **Deterministic case** ($\sigma_t \equiv 0$): Divide-and-Conquer gets $\widetilde{\mathcal{O}}(s)$.

Variance-Aware Sparse Linear Bandit?

The noises $\{\eta_t\}_{t=1}^T$ have time-dependent variances. Formally, $\eta_t \sim \mathcal{N}(0, \sigma_t^2)$ where $\sigma_t \in [0,1]$ varies with time (and is hidden).

- **Worst Case** ($\sigma_t \equiv 1$): $\widetilde{\mathcal{O}}(\sqrt{sdT})$ is known to be optimal.
- **Deterministic case** ($\sigma_t \equiv 0$): Divide-and-Conquer gets $\widetilde{\mathcal{O}}(s)$.
- **In Between?**

Variance-Aware Sparse Linear Bandit?

The noises $\{\eta_t\}_{t=1}^T$ have time-dependent variances. Formally, $\eta_t \sim \mathcal{N}(0, \sigma_t^2)$ where $\sigma_t \in [0,1]$ varies with time (and is hidden).

- **Worst Case** ($\sigma_t \equiv 1$): $\widetilde{\mathcal{O}}(\sqrt{sdT})$ is known to be optimal.
- **Deterministic case** ($\sigma_t \equiv 0$): Divide-and-Conquer gets $\widetilde{\mathcal{O}}(s)$.
- **In Between?** This paper!
  Design an algorithm whose regret is **variance-aware**:

$$\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\text{poly}(s)\sqrt{d\sum_{t=1}^T \sigma_t^2} + \text{poly}(s)\right),$$

where $\sigma_t^2 = \text{Var}(\eta_t) \in [0,1]$ is the noise variance (σ_t's are all *unknown*) and $s = \|\theta^*\|_0$ is the sparsity (s is also *unknown*).

Related Work

1. **"Worst-Case" ($\sigma_t \equiv 1$) Sparse Linear Bandit:**
   - Upper Bound: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
   - Lower Bound: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009].

Related Work

1. **“Worst-Case” ($\sigma_t \equiv 1$) Sparse Linear Bandit:**
   - Upper Bound: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
   - Lower Bound: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009].
2. **“Worst-Case” ($\sigma_t \equiv 1$) Linear Bandits** (i.e., $s = d$):
   - Upper Bound: $\widetilde{\mathcal{O}}(d\sqrt{T})$ [Dani et al., 2008].
   - Lower Bound: $\Omega(d\sqrt{T})$ [Dani et al., 2008].

Related Work

1. **"Worst-Case" ($\sigma_t \equiv 1$) Sparse Linear Bandit:**
   - Upper Bound: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
   - Lower Bound: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009].
2. **"Worst-Case" ($\sigma_t \equiv 1$) Linear Bandits** (i.e., $s = d$):
   - Upper Bound: $\widetilde{\mathcal{O}}(d\sqrt{T})$ [Dani et al., 2008].
   - Lower Bound: $\Omega(d\sqrt{T})$ [Dani et al., 2008].
3. **"Variance-Aware" Linear Bandits:**
   - $\widetilde{\mathcal{O}}(d^{1.5}\sqrt{\sum \sigma_t^2} + d^2)$ [Kim et al., 2022].
   - $\widetilde{\mathcal{O}}(d\sqrt{\sum \sigma_t^2} + \sqrt{dT})$ [Zhou et al., 2021].
   - $\widetilde{\mathcal{O}}(d\sqrt{\sum \sigma_t^2} + d)$ [Zhao et al., 2023] (do not cover).

Related Work

1. **"Worst-Case" ($\sigma_t \equiv 1$) Sparse Linear Bandit:**
   - Upper Bound: $\widetilde{\mathcal{O}}(\sqrt{sdT})$ [Abbasi-Yadkori et al., 2012].
   - Lower Bound: $\Omega(\sqrt{dT})$ [Antos and Szepesvári, 2009].
2. **"Worst-Case" ($\sigma_t \equiv 1$) Linear Bandits** (i.e., $s = d$):
   - Upper Bound: $\widetilde{\mathcal{O}}(d\sqrt{T})$ [Dani et al., 2008].
   - Lower Bound: $\Omega(d\sqrt{T})$ [Dani et al., 2008].
3. **"Variance-Aware" Linear Bandits:**
   - $\widetilde{\mathcal{O}}(d^{1.5}\sqrt{\sum \sigma_t^2} + d^2)$ [Kim et al., 2022].
   - $\widetilde{\mathcal{O}}(d\sqrt{\sum \sigma_t^2} + \sqrt{dT})$ [Zhou et al., 2021].
   - $\widetilde{\mathcal{O}}(d\sqrt{\sum \sigma_t^2} + d)$ [Zhao et al., 2023] (do not cover).

This paper: convert *any* VA-LB Alg $\mathcal{A}$ to VA-SLB Alg $\mathcal{B}$ s.t.:

$$\text{if } \mathcal{A} \text{ ensures } \mathcal{R}_T^{\mathsf{LB}} = \widetilde{\mathcal{O}}\left(f(d)\sqrt{\sum \sigma_t^2} + g(d)\right) \text{ for some } f, g,$$

$$\text{then } \mathcal{B} \text{ ensures } \mathcal{R}_T^{\mathsf{SLB}} = \widetilde{\mathcal{O}}\left((sf(s) + s\sqrt{d})\sqrt{\sum \sigma_t^2} + sg(s)\right).$$

Classical "Explore-then-Commit" Idea

1. *Explore:* Find coordinates with "large enough" magnitudes.
2. *Commit:* Play "wisely" on these coordinates (ignore others).

Classical "Explore-then-Commit" Idea

1. *Explore:* Find coordinates with "large enough" magnitudes.
2. *Commit:* Play "wisely" on these coordinates (ignore others).

Example [Carpentier and Munos, 2012]:

1. *Explore:* Identify all i with $|\theta_i^*| = \Omega((Ts/d)^{-1/4})$ (call this threshold Δ).

Classical “Explore-then-Commit” Idea

1. *Explore:* Find coordinates with “large enough” magnitudes.
2. *Commit:* Play “wisely” on these coordinates (ignore others).

Example [Carpentier and Munos, 2012]:

1. *Explore:* Identify all i with $|\theta_i^*| = \Omega((Ts/d)^{-1/4})$ (call this threshold Δ).Takes $N = \widetilde{\mathcal{O}}(\Delta^{-2}d) = \widetilde{\mathcal{O}}(\sqrt{sdT})$ rounds to make the confidence radius $\sqrt{d/n}$ smaller than $\Delta/2$.

Classical "Explore-then-Commit" Idea

1. *Explore:* Find coordinates with "large enough" magnitudes.
2. *Commit:* Play "wisely" on these coordinates (ignore others).

Example [Carpentier and Munos, 2012]:

1. *Explore:* Identify all i with $|\theta_i^*| = \Omega((Ts/d)^{-1/4})$ (call this threshold Δ).Takes $N = \widetilde{\mathcal{O}}(\Delta^{-2}d) = \widetilde{\mathcal{O}}(\sqrt{sdT})$ rounds to make the confidence radius $\sqrt{d/n}$ smaller than $\Delta/2$.
2. *Commit:* For the remaining $T - N$ rounds, execute a linear bandit algorithm on these coordinates (i.e., only consider an $\mathcal{O}(s)$-dimensional subspace) and play 0 on the other ones.

Classical "Explore-then-Commit" Idea

1. *Explore:* Find coordinates with "large enough" magnitudes.
2. *Commit:* Play "wisely" on these coordinates (ignore others).

Example [Carpentier and Munos, 2012]:

1. *Explore:* Identify all i with $|\theta_i^*| = \Omega((Ts/d)^{-1/4})$ (call this threshold Δ).Takes $N = \widetilde{\mathcal{O}}(\Delta^{-2}d) = \widetilde{\mathcal{O}}(\sqrt{sdT})$ rounds to make the confidence radius $\sqrt{d/n}$ smaller than $\Delta/2$.
2. *Commit:* For the remaining $T - N$ rounds, execute a linear bandit algorithm on these coordinates (i.e., only consider an $\mathcal{O}(s)$-dimensional subspace) and play 0 on the other ones.

Regret Analysis: The regret $\mathcal{R}_T = \widetilde{\mathcal{O}}(\sqrt{sdT})$, as:

- *Exploration* causes no more than $N = \widetilde{\mathcal{O}}(\sqrt{sdT})$ regret.
- *Commitment* on s coordinates has regret $\widetilde{\mathcal{O}}(s\sqrt{T})$.
- Each *un-explored coordinate* i (which is "small") incurs regret $\le T\Delta^2 = \sqrt{dT/s}$; and there are no more than s such i's.

Generalize to Variance-Aware SLB?

Question 1: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "commit"?

Generalize to Variance-Aware SLB?

Question 1: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "commit"?

- **Answer:** Use variance-aware LB algorithms.

Generalize to Variance-Aware SLB?

Question 1: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "commit"?

- **Answer:** Use variance-aware LB algorithms.

Question 2: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "explore"?

Generalize to Variance-Aware SLB?

Question 1: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "commit"?

- **Answer:** Use variance-aware LB algorithms.

Question 2: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "explore"?

1. *Worst-Case*: Exploration thresold $\Delta \sim T^{-1/4}$.
2. *Deterministic-Case*: Exploration thresold $\Delta \sim 0$.

Generalize to Variance-Aware SLB?

Question 1: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "commit"?

- **Answer:** Use variance-aware LB algorithms.

Question 2: How to get $\sqrt{\sum \sigma_t^2}$-style regret in "explore"?

1. *Worst-Case*: Exploration thresold $\Delta \sim T^{-1/4}$.
2. *Deterministic-Case*: Exploration thresold $\Delta \sim 0$.

- **Answer:** Decide the "threshold" Δ *adaptively*.

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Our Idea: “Adaptive” Exploration Threshold

Algorithm “Explore-then-Commit” with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 3: How to do exploration?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 3: How to do exploration?

- Explore all coordinates? Then why halving?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 3: How to do exploration?

- Explore all coordinates? Then why halving?
- Ignore identified coordinates? Their regret?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 3: How to do exploration?

- Explore all coordinates? Then why halving?
- Ignore identified coordinates? Their regret?
- **Solution**: Put estimations on identified (large) coordinates. Use remaining mass $1 - \sum \hat{\theta}_i^2$ to explore remaining ones.

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 4: When to stop exploration?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 4: When to stop exploration?

- Confidence radius? (Chernoff / Bernstein ...)

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 4: When to stop exploration?

- Confidence radius? (Chernoff / Bernstein ...)
- $\frac{1}{n}\sqrt{d\sum_{k=1}^{n}\sigma_k^2}$ contains unknown σ_k's?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 4: When to stop exploration?

- Confidence radius? (Chernoff / Bernstein ...)
- $\frac{1}{n}\sqrt{d\sum_{k=1}^{n}\sigma_k^2}$ contains unknown σ_k's?
- Use "empirical" observations to replace σ_k^2?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 4: When to stop exploration?

Lemma: For common-mean, independent & symmetric $\{X_i\}_{i=1}^n$,

$$|\bar{X} - \mu| \le \frac{1}{n}\sqrt{2\sum_{i=1}^n (X_i - \bar{X})^2 \ln\frac{4}{\delta}} \quad \text{w.p. } 1-\delta,$$

where $n < \infty$ is stopping time, $\mu = \mathbb{E}[X_i]$, and $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$.

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

- Recall: we need $\hat{\theta}_i$ for all identified i?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

- Recall: we need $\hat{\theta}_i$ for all identified i?
- Recall: LB Alg can "learn" the parameter θ^*?

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

- Recall: we need $\hat{\theta}_i$ for all identified i?
- Recall: LB Alg can "learn" the parameter θ^*?
- **Answer:** Stop if a close estimation is learned.

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

"Regret-to-Sample-Complexity": if $\mathcal{A}$'s per-round regret $< \Delta^2$, i.e.,

$$\mathcal{R}_n^{\mathcal{A}} = \sum_{k=1}^{n} \langle \theta^* - a_k, \theta^* \rangle \le n\Delta^2, \text{ then } \hat{\theta} \triangleq \frac{1}{n}\sum_{k=1}^{n} a_k \text{ satisfies } \langle \theta^* - \hat{\theta}, \theta^* \rangle \le \Delta^2.$$

Our Idea: "Adaptive" Exploration Threshold

Algorithm "Explore-then-Commit" with Adaptive Threshold

1: **for** $\Delta = \frac{1}{2}, \ldots$ **do**
2: **Explore:** Identify all coordinates with magnitude $[\Delta, 2\Delta]$.
3: **Commit:** Deploy VA LB $\mathcal{A}$ on all identified coordinates.
4: **Continue:** Halve Δ and repeat.

Question 5: When to stop commit?

"Regret-to-Sample-Complexity": if $\mathcal{A}$'s per-round regret $< \Delta^2$, i.e.,

$$\mathcal{R}_n^{\mathcal{A}} = \sum_{k=1}^{n} \langle \theta^* - a_k, \theta^* \rangle \le n\Delta^2, \text{ then } \hat{\theta} \triangleq \frac{1}{n} \sum_{k=1}^{n} a_k \text{ satisfies } \langle \theta^* - \hat{\theta}, \theta^* \rangle \le \Delta^2.$$

So waiting until $\mathcal{R}_n^{\mathcal{A}} \le n\Delta^2$ gives "good" estimation $\hat{\theta}$.

Final Algorithm

Algorithm Final Algorithm (Using VA LB Algorithm $\mathcal{A}$)

1: **for** $\Delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (i.e., halve until T rounds) **do**

Final Algorithm

Algorithm Final Algorithm (Using VA LB Algorithm $\mathcal{A}$)

1: **for** $\Delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (i.e., halve until T rounds) **do**

2: For each round, put $\hat{\theta}_i$ on i for all identified i, and use remaining mass to explore like [Carpentier and Munos, 2012].

Final Algorithm

Algorithm Final Algorithm (Using VA LB Algorithm $\mathcal{A}$)

1: **for** $\Delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (i.e., halve until T rounds) **do**

2: For each round, put $\hat{\theta}_i$ on i for all identified i, and use remaining mass to explore like [Carpentier and Munos, 2012].

3: Terminate until 'explore" rounds n_Δ^b ensures

$$2\sqrt{2\sum_{k=1}^{n_\Delta^b}(r_{k,i} - \bar{r}_i)^2 \ln \frac{4}{\delta}} < n_\Delta^b \cdot \frac{\Delta}{4}, \quad \forall i \text{ unidentified},$$

where $r_{k,i}$ is the k-th estimate of θ_i^* and $\bar{r}_i$ is the average of all $r_{k,i}$'s. Then mark all i with $|\bar{r}_i| > \Delta$ as "identified".

Final Algorithm

Algorithm Final Algorithm (Using VA LB Algorithm $\mathcal{A}$)

1: **for** $\Delta = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$ (i.e., halve until T rounds) **do**

2: For each round, put $\hat{\theta}_i$ on i for all identified i, and use remaining mass to explore like [Carpentier and Munos, 2012].

3: Terminate until 'explore" rounds n_Δ^b ensures

$$2\sqrt{2\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2 \ln\frac{4}{\delta}} < n_\Delta^b \cdot \frac{\Delta}{4}, \quad \forall i \text{ unidentified},$$

where $r_{k,i}$ is the k-th estimate of θ_i^* and $\bar{r}_i$ is the average of all $r_{k,i}$'s. Then mark all i with $|\bar{r}_i| > \Delta$ as "identified".

4: Deploy $\mathcal{A}$ on all identified coordinates until "commit" rounds n_Δ^a ensures $\mathcal{R}_{n_\Delta^a}^{\mathcal{A}} < n_\Delta^a \cdot \Delta^2$. Calculate $\hat{\theta}_i$ for all identified i.

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

1. **"Explore" Regret:**
   1. Identified ones contribute regret $n_\Delta^b\langle\theta^*-\hat{\theta},\theta^*\rangle \le n_\Delta^b\cdot\Delta^2$.

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

1. **"Explore" Regret:**
   1. Identified ones contribute regret $n_\Delta^b\langle\theta^* - \hat{\theta}, \theta^*\rangle \le n_\Delta^b \cdot \Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^b \sum_i (\theta_i^*)^2 \le n_\Delta^b \cdot s\Delta^2$.

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

1. **"Explore" Regret:**
   1. Identified ones contribute regret $n_\Delta^b\langle\theta^*-\hat{\theta},\theta^*\rangle \le n_\Delta^b\cdot\Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^b\sum_i(\theta_i^*)^2 \le n_\Delta^b\cdot s\Delta^2$.
2. **"Commit" Regret:**
   1. Identified ones contribute regret $\mathcal{R}_{n_\Delta^a}^{\mathcal{A}} < n_\Delta^a\cdot\Delta^2$.

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

1. **"Explore" Regret:**
   1. Identified ones contribute regret $n_\Delta^b\langle\theta^*-\hat{\theta},\theta^*\rangle \le n_\Delta^b \cdot \Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^b \sum_i(\theta_i^*)^2 \le n_\Delta^b \cdot s\Delta^2$.
2. **"Commit" Regret:**
   1. Identified ones contribute regret $\mathcal{R}_{n_\Delta^a}^{\mathcal{A}} < n_\Delta^a \cdot \Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^a \sum_i(\theta_i^*)^2 \le n_\Delta^a \cdot s\Delta^2$.

Analysis Sketch

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}.$$

1. **"Explore" Regret:**
   1. Identified ones contribute regret $n_\Delta^b\langle\theta^*-\hat{\theta},\theta^*\rangle \le n_\Delta^b\cdot\Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^b\sum_i(\theta_i^*)^2 \le n_\Delta^b\cdot s\Delta^2$.
2. **"Commit" Regret:**
   1. Identified ones contribute regret $\mathcal{R}_{n_\Delta^a}^{\mathcal{A}} < n_\Delta^a\cdot\Delta^2$.
   2. Unidentified ones contribute regret $n_\Delta^a\sum_i(\theta_i^*)^2 \le n_\Delta^a\cdot s\Delta^2$.
3. **Conclusion:** Total Regret

$$\mathcal{R}_T = \mathcal{O}\left(\mathbb{E}\left[\sum_\Delta s\Delta^2(n_\Delta^b+n_\Delta^a)\right]\right).$$

Analysis Sketch (Cont'd)

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}},$$

Analysis Sketch (Cont'd)

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}},$$

and ...

$$\mathcal{R}_T = \mathcal{O}\left(\mathbb{E}\left[\sum_\Delta s\Delta^2(n_\Delta^b + n_\Delta^a)\right]\right),$$

Analysis Sketch (Cont'd)

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}},$$

and ...

$$\mathcal{R}_T = \mathcal{O}\left(\mathbb{E}\left[\sum_\Delta s\Delta^2(n_\Delta^b + n_\Delta^a)\right]\right),$$

so ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_\Delta \Delta^2\left(\frac{1}{\Delta}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2} + \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}\right)\right].$$

Analysis Sketch (Cont'd)

Recap: For each Δ, n_Δ^b and n_Δ^a are defined as (ignore constants)

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}, \quad n_\Delta^a \approx \Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}},$$

and ...

$$\mathcal{R}_T = \mathcal{O}\left(\mathbb{E}\left[\sum_\Delta s\Delta^2(n_\Delta^b+n_\Delta^a)\right]\right),$$

so ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_\Delta \Delta^2\left(\frac{1}{\Delta}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2}+\Delta^{-2}\mathcal{R}_{n_\Delta^a}^{\mathcal{A}}\right)\right].$$

We know ... $\mathcal{R}_n^{\mathcal{A}} = \widetilde{\mathcal{O}}\left(s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2}+s^2\right)$ [Kim et al., 2022],

and $\sum_{k=1}^{n_\Delta^b}(r_{k,i}-\bar{r}_i)^2 \approx \sum_{k=1}^{n_\Delta^b}\mathbb{E}[(r_{k,i}-\bar{r}_i)^2] = \sum_{k=1}^{n_\Delta^b}(1+\frac{d}{\Delta^2}\sigma_k^2)$.

Analysis Sketch (Cont'd)

So we have ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_{\Delta}\left(\sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2} + s^2\right)\right].$$

Analysis Sketch (Cont'd)

So we have ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_{\Delta}\left(\sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2} + s^2\right)\right].$$

Question 7: How to bound $\sum_\Delta \sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} \triangleq \sum_\Delta \sqrt{S_\Delta}$?

Analysis Sketch (Cont'd)

So we have ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_{\Delta}\left(\sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2 + s^2}\right)\right].$$

Question 7: How to bound $\sum_\Delta \sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} \triangleq \sum_\Delta \sqrt{S_\Delta}$?

- **Answer**: Recall $\sum_\Delta n_\Delta^b \le T$ and

$$n_\Delta^b \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}(r_{k,i} - \bar{r}_i)^2} \approx \Delta^{-1}\sqrt{\sum_{k=1}^{n_\Delta^b}\left(1 + \frac{d}{\Delta^2}\sigma_k^2\right)} = \Delta^{-2} S_\Delta.$$

In other words, we have $\sum_\Delta \Delta^{-2}\sqrt{S_\Delta} \le T$ (and $\Delta = 2^{-1}, 2^{-2}, \ldots$).

Analysis Sketch (Cont'd)

So we have ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_{\Delta}\left(\sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2} + s^2\right)\right].$$

Question 7: How to bound $\sum_\Delta \sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} \triangleq \sum_\Delta \sqrt{S_\Delta}$?

- **Answer (Cont'd)**: $\sum_\Delta \Delta^{-2}\sqrt{S_\Delta} \le T$ and $\Delta = 2^{-1}, 2^{-2}, \ldots$. Define a threshold $X = \sqrt{\sum_\Delta S_\Delta}/T$, then:
  - *For* $\Delta^2 \le X$: $\sum_{\Delta^2 \le X} \sqrt{S_\Delta} \le X \sum_{\Delta^2 \le X} \Delta^{-2}\sqrt{S_\Delta} \le TX$.
  - *For* $\Delta^2 \ge X$: $\sum_{\Delta^2 \ge X} \sqrt{S_\Delta} \le \widetilde{\mathcal{O}}(\sqrt{\sum_\Delta S_\Delta})$ $(\#\Delta \le \log_2 T)$.

So $\sum_\Delta \sqrt{S_\Delta} = \widetilde{\mathcal{O}}(\sqrt{\sum_\Delta S_\Delta}) = \widetilde{\mathcal{O}}(\sqrt{\sum_\Delta \sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)})$!

Analysis Sketch (Cont'd)

So we have ...

$$\mathcal{R}_T = \widetilde{\mathcal{O}}(s)\,\mathbb{E}\left[\sum_{\Delta}\left(\sqrt{\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{k=1}^{n_\Delta^a}\sigma_k^2 + s^2}\right)\right]$$

$$= \widetilde{\mathcal{O}}\left(s\,\mathbb{E}\left[\sqrt{\sum_{\Delta}\sum_{k=1}^{n_\Delta^b}(\Delta^2 + d\sigma_k^2)} + s^{1.5}\sqrt{\sum_{\Delta}\sum_{k=1}^{n_\Delta^a}\sigma_k^2 + \sum_{\Delta} s^2}\right]\right)$$

$$= \widetilde{\mathcal{O}}\left((s^{2.5} + s\sqrt{d})\sqrt{\sum_{t=1}^{T}\sigma_t^2 + s^3}\right). \qquad \square$$

Thank you for listening!

Questions are more than welcomed.

References

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. (2012).
Online-to-confidence-set conversions and application to sparse stochastic bandits.
In *Artificial Intelligence and Statistics*, pages 1–9. PMLR.

Antos, A. and Szepesvári, C. (2009).
Stochastic bandits with large action sets revisited.
Personal communication.

Carpentier, A. and Munos, R. (2012).
Bandit theory meets compressed sensing for high dimensional stochastic linear bandit.
In *Artificial Intelligence and Statistics*, pages 190–198. PMLR.

Dani, V., Hayes, T. P., and Kakade, S. M. (2008).
Stochastic linear optimization under bandit feedback.
In *21st Annual Conference on Learning Theory*, pages 355–366.

Kim, Y., Yang, I., and Jun, K.-S. (2022).
Improved regret analysis for variance-adaptive linear bandits and horizon-free linear mixture mdps.
In *Advances in Neural Information Processing Systems 35*.

Zhao, H., He, J., Zhou, D., Zhang, T., and Gu, Q. (2023).
Variance-dependent regret bounds for linear bandits and reinforcement learning: Adaptivity and computational efficiency.
arXiv preprint arXiv:2302.10371.

Zhou, D., Gu, Q., and Szepesvari, C. (2021).
Nearly minimax optimal reinforcement learning for linear mixture markov decision processes.
In *Conference on Learning Theory*, pages 4532–4576. PMLR.