INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Best Arm Identification in Linear Bandits

Richa Kushwaha 20172056

Goutham Bhatta 20172063

Mentors-

Dr. Naresh Manwani

Sahil Chelaramani

Problem Setting

Problem	**Multi Arm Bandit Setting**	**Linear Bandit Setting**
Environment	K independent arms, with unknown distribution.	Stochastic linear arms $r(x) = x^T\theta^*$
Predict	μ, $\boldsymbol{\sigma}$ (given $\{r_1 \ldots r_t\}$)	θ^* (given x_i, $\{r_1 \ldots r_t\}$)
Objective	Find best arm, while maximizing cumulative reward. <u>(Trade off between exploration and exploitation)</u>	Find the best arm with fixed confidence while minimizing sample complexity. <u>(Pure exploration)</u>

Are rewards deterministic ??

Reward => $r(x) = x^T\theta^* + \varepsilon$,

where ε is a zero-mean i.i.d. noise bounded in $[-\sigma; \sigma]$.

Also, $X \subseteq R^d$ be the set arms, $|X| = K$ and $\theta^* \in R^d$

Terminology

Value gap : The difference between the rewards of two arms.

$$\Delta(x, x') = (x - x')^T\theta^*$$

Direction : The difference between two arms.

$$Y = \{x - x'\} \quad \forall\, x, x' \in X$$

Modelling the problem

$\hat{x}(n)$ => estimated best arm after n steps

Regret : $R = (x^* - \hat{x}(n))^\top \theta^*$.

PAC setting: $P(R \geq \epsilon) \leq \delta$ where $\epsilon, \delta \in (0, 1)$

Design an allocation strategy such that it returns arm $\hat{x}(n)$ following PAC condition, while minimizing the needed number of steps.

OLS estimate of θ

We know that, $r(x) = x^T\theta$. Let $\mathbf{x_n}$ represent sequence of **n** pulls. It is given as,

$\mathbf{x}_n = (x_1, \ldots, x_n)$ and $(r_1, \ldots, r_n)$

$$
\begin{aligned}
\text{At,}\quad & t_1 \to x_1 x^T\theta^* = x_1 r_1 \\
& t_2 \to x_2 x^T\theta^* = x_2 r_2 \\
& \quad\cdot \qquad \cdot \qquad \cdot \\
& \quad\cdot \qquad \cdot \qquad \cdot \\
& t_n \to x_n x^T\theta^* = x_n r_n
\end{aligned}
$$

summation from t_1 to t_n we get the following equation.

$$\hat{\theta}_n = A_{\mathbf{x}_n}^{-1} b_{\mathbf{x}_n}$$

where $A_{\mathbf{x}_n} = \sum_{t=1}^{n} x_t x_t^\top$ and $b_{\mathbf{x}_n} = \sum_{t=1}^{n} x_t r_t$

Bounds on prediction error of the OLS estimate :

Case 1) **Fixed sequence (varying confidence):**

P (orignal_reward - predicted_reward <= k) >= 1 - ***δ***

$$\mathbb{P}\left(\forall n \in \mathbb{N}, \forall x \in \mathcal{X}, \left|x^\top \theta^* - x^\top \hat{\theta}_n\right| \leq c||x||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log(c' n^2 K/\delta)}\right) \geq 1 - \delta.$$

(obtained using azuma's inequality)

Case 2) **Adaptive sequence (fixed confidence):**

P (orignal_reward - predicted_reward <= √d k) = 1 - ***δ***

$$\left|x^\top \theta^* - x^\top \hat{\theta}_n^\eta\right| \leq ||x||_{(\widetilde{A}_{\mathbf{x}_n}^\eta)^{-1}} \left(\sigma \sqrt{d \log\left(\frac{1 + nL^2/\eta}{\delta}\right)} + \eta^{1/2}||\theta^*||\right).$$

Soft allocation strategy:

- Considers the proportions of pulls of arm x.
- Replace A_x by Λ_λ where $\Lambda_\lambda = \lambda(x)\, xx^\top$ and $\lambda(x) = T_n(x)/n$, $T_n(x)$ = no.of times arm x is pulled in sequence $\mathbf{x}_n$

Cone of Arm :

- $C(x) = \{ \theta \in R^d , x \in \pi(\theta) \}$
  - set of all parameters θ which admit *x* is an optimal arm.
- Since Oracle knows x*, which means it also knows *C(x*)*.

Confidence Set :

Given static allocation, $\mathbf{x_n}$

- $S^*(x_n) \subseteq R^d$,
  - $s.t.\ \theta^* \in S^*(x_n)$
  - O.L.S. estimate of $\hat{\theta}_n \in S^*(x_n)$ with high probability $P(\hat{\theta}_n \in S^*(x_n)) \geq 1- \delta.$

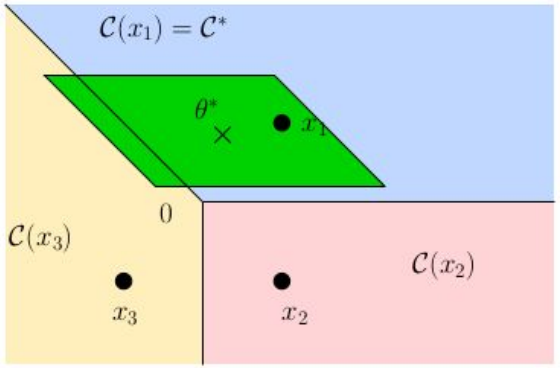
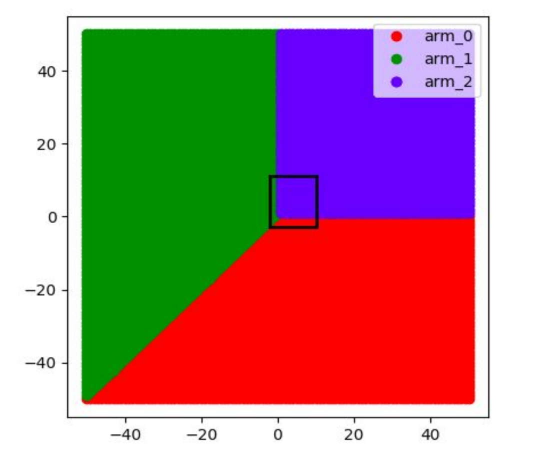
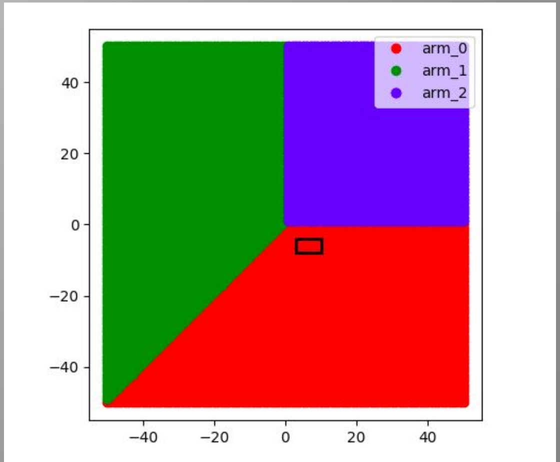


Figure 1: The cones corresponding to three arms (dots) in $\mathbb{R}^2$. Since $\theta^* \in \mathcal{C}(x_1)$, then $x^* = x_1$. The confidence set $\mathcal{S}^*(\mathbf{x}_n)$ (in green) is aligned with directions $x_1 - x_2$ and $x_1 - x_3$. Given the uncertainty in $\mathcal{S}^*(\mathbf{x}_n)$, both x_1 and x_3 may be optimal.

Oracle Stopping Condition :

- Stopping condition => If $S^*(x_n)$ is contained in $C(x^*)$.
- Two Scenarios :
  - $S^*(x_n)$ overlaps cones of different arms $x \in X$
    - Ambiguity to identify arm $\pi(\hat{\theta}_n)$.
  - $S^*(x_n)$ lies in one cone
    - Optimal arm is returned.





Modelling Confidence Set [Oracle Allocation Strategy] :

- Objective: To converge $S^*(x_n)$ *into* $C(x^*)$ in minimum no.of step
  - The condition $S^*(x_n) \subseteq C(x^*)$ is equivalent to

$$\forall x \in \mathcal{X}, \forall \theta \in \mathcal{S}^*(\mathbf{x}_n), (x^* - x)^\top \theta \geq 0 \quad \Leftrightarrow \quad \forall y \in \mathcal{Y}^*, \forall \theta \in \mathcal{S}^*(\mathbf{x}_n), y^\top (\theta^* - \theta) \leq \Delta(y)$$

  - Replacing y (directions) in place of x (arms) *Prop. 1* , we obtain

$$c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_n (K^2/\delta)} \leq \Delta(y)$$

  - Using the above two equations, we can define an optimal static allocation as

$$\mathbf{x}_n^* = \arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{c||y||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_n (K^2/\delta)}}{\Delta(y)} = \arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}^*} \frac{||y||_{A_{\mathbf{x}_n}^{-1}}}{\Delta(y)}$$

4. Oracle to Empirical stopping condition

- Oracle algorithm is not feasible, since x* and theta* are unknown.
- Given arms X, C(x) can be computed for each arm.
- S^(x) (Empirical confidence set) can be constucted from samples.
- Hence, new stopping condition becomes $\widehat{\mathcal{S}}(\mathbf{x}_n) \subseteq \mathcal{C}(x)$

$$\exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, \forall \theta \in \widehat{\mathcal{S}}(\mathbf{x}_n), (x - x')^\top \theta \geq 0$$
$$\Leftrightarrow \quad \exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, \forall \theta \in \widehat{\mathcal{S}}(\mathbf{x}_n), (x - x')^\top (\hat{\theta}_n - \theta) \leq \widehat{\Delta}_n(x, x'). \tag{9}$$

This suggests that the empirical confidence set can be defined as

$$\widehat{\mathcal{S}}(\mathbf{x}_n) = \left\{ \theta \in \mathbb{R}^d, \forall y \in \mathcal{Y}, y^\top (\hat{\theta}_n - \theta) \leq c \|y\|_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_n(K^2/\delta)} \right\}. \tag{10}$$

Unlike $\mathcal{S}^*(\mathbf{x}_n)$, $\widehat{\mathcal{S}}(\mathbf{x}_n)$ is centered in $\hat{\theta}_n$ and it considers all directions $y \in \mathcal{Y}$. As a result, the stopping condition in Eq. 9 could be reformulated as

$$\exists x \in \mathcal{X}, \forall x' \in \mathcal{X}, c\|x - x'\|_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_n(K^2/\delta)} \leq \widehat{\Delta}_n(x, x'). \tag{11}$$

4.1 Static allocation strategies

Here, We propose two allocations strategies that achieve the stopping condition as fast as possible.

1. **G Allocation Strategy.** (Name borrowed from optimal design)

   It follows from the observation that,

   for any pair $(x, x') \in \mathcal{X}^2$ we have that $||x - x'||_{A_{\mathbf{x}_n}^{-1}} \leq 2 \max_{x'' \in \mathcal{X}} ||x''||_{A_{\mathbf{x}_n}^{-1}}$.

   We try to minimize this upper bound. Leading to the following eqn.

$$\mathbf{x}_n^G = \arg\min_{\mathbf{x}_n} \max_{x \in \mathcal{X}} ||x||_{A_{\mathbf{x}_n}^{-1}}.$$

2. **XY Allocation Strategy.**

$$\mathbf{x}_n^{\mathcal{X}\mathcal{Y}} = \arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}} ||y||_{A_{\mathbf{x}_n}^{-1}}.$$

   Follows from the observation that, arms should be pulled with the objective of increasing the accuracy over directions rather than arms

4.1.1 Static allocation algorithms

The above problems are NP-hard discrete optimization problems. Hence we use an incremental approach to get an approximate solution.

Input: decision space $\mathcal{X} \in \mathbb{R}^d$, confidence $\delta > 0$
Set: $t = 0$; $Y = \{y = (x - x'); x \neq x' \in \mathcal{X}\}$;
while Eq. 11 is not true **do**
  if G-allocation **then**
    $x_t = \arg\min_{x \in X} \max_{x' \in X} x'^\top (A + xx^\top)^{-1} x'$
  else if $\mathcal{X}\mathcal{Y}$-allocation **then**
    $x_t = \arg\min_{x \in X} \max_{y \in Y} y^\top (A + xx^\top)^{-1} y$
  end if
  Update $\hat{\theta}_t = A_t^{-1} b_t$, $t = t + 1$
end while
Return arm $\Pi(\hat{\theta}_t)$

Figure 2: Static allocation algorithms

5. Adaptive algorithms

- Upper bounds for sample complexity of both G, XY allocation algorithms scale linearly with ‘d’ . (From theorem 1, 2)
- Even adaptive algorithms suffer from ‘sqrt(d)’ dimensionality problem. As seen in proposition 2.
- Hence we propose a phased algorithm where we combine both static and adaptive algorithms, whose sample complexity bound does not depend upon ‘d’.

Sub Optimal Condition:

$$\exists x' \in \mathcal{X} \ s.t. \ c||x' - x||_{A_{\mathbf{x}_n}^{-1}} \sqrt{\log_n(K^2/\delta)} < \widehat{\Delta}_n(x', x),$$

5. XY - Adaptive algorithm

We Introduce few terms,

X_hat_j => Set of potentially optimal arms in phase j

Hence, new stopping condition => | X_hat_j | = 1

Algorithm:

1. In each phase we implement XY iterative algo.
2. The phase length is determined by the uncertainty present in estimating the active directions between successive phases.
3. Once a phase ends then we compute theta_hat using OLS method.
4. We then use the sub-optimal condition to remove the arms from X_hat_j.
5. And loop over the above steps until we meet stopping condition.

Input: decision space $\mathcal{X} \in \mathbb{R}^d$; parameter α; confidence δ
Set $j=1$; $\widehat{\mathcal{X}}_j = \mathcal{X}$; $\widehat{\mathcal{Y}}_1 = \mathcal{Y}$; $\rho_0 = 1$; $n_0 = d(d+1)+1$
while $|\widehat{\mathcal{X}}_j| > 1$ **do**
  $\rho^j = \rho^{j-1}$
  $t = 1$; $A_0 = I$
  while $\rho^j / t \geq \alpha \rho^{j-1}(\mathbf{x}^{j-1}_{n_{j-1}})/n_{j-1}$ **do**
    Select arm $x_t = \arg\min_{x \in X} \max_{y \in Y} y^\top (A + xx^\top)^{-1} y$
    Update $A_t = A_{t-1} + x_t x_t^\top$, $t = t + 1$
    $\rho^j = \max_{y \in \widehat{\mathcal{Y}}_j} y^\top A_t^{-1} y$
  end while
  Compute $b = \sum_{s=1}^{t} x_s r_s$; $\hat{\theta}_j = A_t^{-1} b$
  $\widehat{\mathcal{X}}_{j+1} = \mathcal{X}$
  for $x \in \mathcal{X}$ **do**
    if $\exists x' : ||x - x'||_{A_t^{-1}} \sqrt{\log_n(K^2/\delta)} \leq \widehat{\Delta}_j(x', x)$ **then**
      $\widehat{\mathcal{X}}_{j+1} = \widehat{\mathcal{X}}_{j+1} - \{x\}$
    end if
  end for
  $\widehat{\mathcal{Y}}_{j+1} = \{y = (x - x'); x, x' \in \widehat{\mathcal{X}}_{j+1}\}$
end while
Return $\Pi(\hat{\theta}_j)$

Figure 3: $\mathcal{X}\mathcal{Y}$-Adaptive allocation algorithm

Published results:

Algorithm	No.of samples required
XY-adaptive	O(k)
G, XY-static	O(d)

Our scope Implementation:

- G allocation strategy
- XY allocation strategy
- XY-adaptive allocation strategy

Plot the performance w.r.t dimensionality as shown in the paper and compare them.

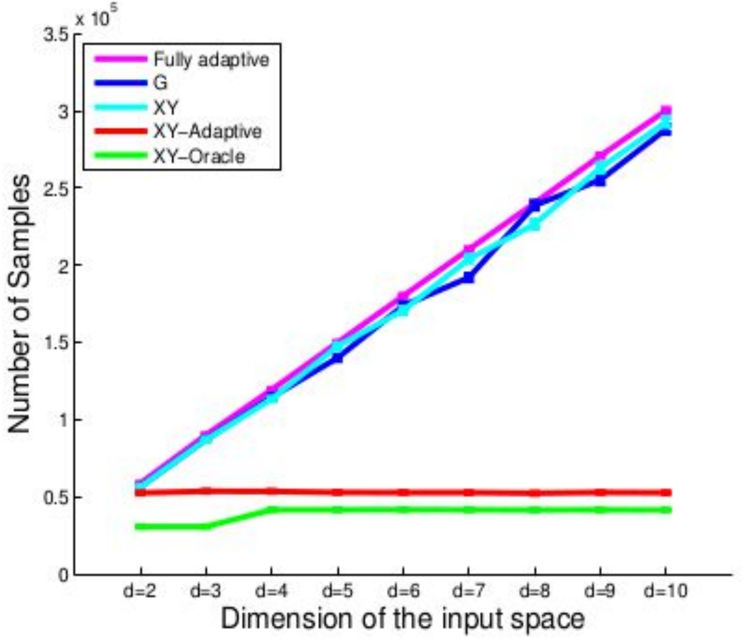


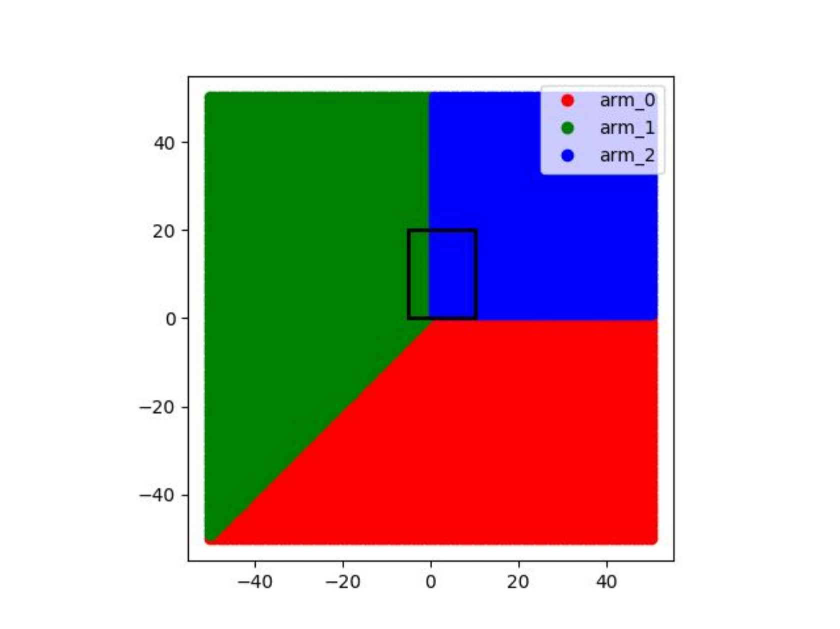
Figure 4: The sampling budget needed to identify the best arm, when the dimension grows from $\mathbb{R}^2$ to $\mathbb{R}^{10}$.

Implementation:

```
****************Phase No. : 2**********************
Current phase length : 27
Theta hat[Predicted Theta] :
[0.46426205]
[0.46782945]
No. of dominant arms : 3
The dominant arms :
0       1       2


****************Phase No. : 3**********************
Current phase length : 42
Theta hat[Predicted Theta] :
[0.4809838]
[0.47436704]
No. of dominant arms : 3
The dominant arms :
0       1       2


****************Phase No. : 4**********************
Current phase length : 56
Theta hat[Predicted Theta] :
[0.48038249]
[0.48229667]
No. of dominant arms : 1
The dominant arms :
2

Original Theta :
[0.5]
[0.5]
```

XY-adaptive algo with d = 2, K = 3,d = 0.05

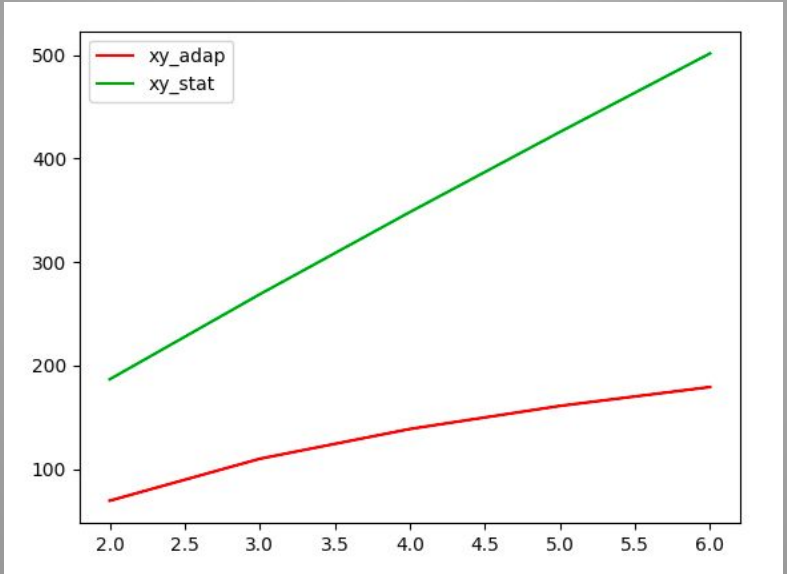
Visualization of confidence set after each phase in theta space.



On a sample run of XY-adaptive algorithm

Observation:

No.of samples vs dimensionality



Reference :

- Research paper on Best Arm Identification in Linear Bandit
  https://arxiv.org/abs/1409.6110

THANK-YOU