Challenges in adaptive algorithms:

a discussion through best arm identification in linear bandits

Aarshvi Gajjar

29 April 2025

Best Arm Identification - The General Setting

- The learner and the environment interact sequentially over some number of rounds. The number of rounds is not fixed in advance.
- In each round $t = 1, 2, \ldots$ the learner chooses an action A_t from a fixed finite set, which is fed to the environment.
- The environment then samples a reward R_t from some distribution which depends on A_t.
- The goal of the learner is to identify, with high probability, an action close enough to the optimal action, in as few rounds of interaction as possible.

Best Arm Identification - The Linear Bandit Setting

- The actions (also called arms) are denoted with the set $[K]$.
- Each arm $i \in [K]$ has an associated *known* feature vector $x_i \in \mathbb{R}^d$, for some $d \geq 1$.
- The rewards are modeled as

$$R_t = x_{A_t}^\top \theta^* + \eta_t,$$

  where $\theta^* \in \mathbb{R}^d$ is an *unknown* parameter and η_t is noise.

- Denote

$$\mathcal{F}_t = \sigma(A_1, R_1, ..., A_{t-2}, R_{t-2}, A_{t-1})$$

  the σ-algebra summarizing the information available just before the reward R_t is observed.

Best Arm Identification - The Linear Bandit Setting

- We assume that the noise is conditionally R-sub-Gaussian, i.e.,

$$\mathbb{E}[\exp(\lambda \eta_t) \,|\, \mathcal{F}_t] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right).$$

- Note that this implies $\mathbb{E}[\eta_t \,|\, \mathcal{F}_t] = 0$ and thus $\mathbb{E}[R_t \,|\, \mathcal{F}_t] = x_{A_t}^\top \theta^*$.
- Denote $a^* = \arg\max_{i \in [K]} x_i^\top \theta^*$.
- **Goal**: Design an algorithm that given $\varepsilon, \delta \in (0, 1)$ outputs an arm $\hat{a}$ such that

$$\mathbb{P}\{x_{a^*}^\top \theta^* - x_{\hat{a}}^\top \theta^* \geq \varepsilon\} \leq \delta$$

  in as few rounds of interaction as possible.

Comparison with Multi Armed Bandits

- Linear bandits setting generalizes the multi-armed bandits setting.
- To see this, let $x_i = e_i$ be the i^{th} standard basis vector in $\mathbb{R}^d$.

Static vs Adaptive Algorithms

- In static algorithms the learner chooses the action sequence $A_1, A_2, \ldots$ *before* observing the rewards, and then estimates the best action in an "offline" manner.
- In adaptive algorithms the learner chooses each action A_t as a function of the past observations. This creates probabilistic dependencies in the stochastic process $A_1, R_1, A_2, R_2, \ldots$, making the analysis challenging, as we will se.

A General Strategy: Constructing Confidence Sets

- At time t we have observed the history $H_t = (A_1, R_1, ..., A_{t-1}, R_{t-1})$.
- Using H_t we estimate two quantities:
  - an estimator $\hat{\theta}_t$ of the unknown parameter θ^*, and
  - a confidence ellipsoid C_t such that it contains θ^* with high probability.
- We want to choose actions in a manner that shrinks the confidence ellipsoids as quickly as possible.

Interlude: Least Squares Regression

- Suppose you observe i.i.d. data $(x_1, y_1), ..., (x_n, y_n) \in \left(\mathbb{R}^d, \mathbb{R}\right)$.
- You assume that there exists $w^* \in \mathbb{R}^d$ such that $Xw^* \approx y$, where X is the $n \times d$ matrix containing x_i in its i^{th} row and y is the column vector containing y_i's.
- In ℓ^2-regularized regression (also called ridge regression) we find

$$w_{\text{RR}} = \arg\min_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \lambda \|w\|_2^2.$$

- It can be shown that

$$w_{\text{RR}} = \left(X^\top X + \lambda I\right)^{-1} X^\top y.$$

- Note that $X^\top X = \sum_{i=1}^n x_i x_i^\top$.

A Static Algorithm: Estimating $\hat{\theta}_n$

- A similar strategy as above can be applied to linear bandits.
- If the actions are chosen in a static manner, then the data $\left(x_{A_1}, R_1\right), \ldots, \left(x_{A_n}, R_n\right)$ becomes i.i.d.
- Define

$$\Sigma_n^\lambda = \sum_{i=1}^n x_{A_i} x_{A_i}^\top + \lambda I, \quad b_n = \sum_{i=1}^n x_{A_i} R_i.$$

- Our estimate $\hat{\theta}_n$ of θ^* then becomes

$$\hat{\theta}_n = \left(\Sigma_n^\lambda\right)^{-1} b_n.$$

A Static Algorithm: Estimating Confidence Ellipsoids

- To get a confidence ellipsoid we use Azuma-Hoeffding's inequality.
- Using union bound over all possible $x \in \{x_1, ..., x_K\}$, it suffices to upper bound

$$\left| x^\top \left(\hat{\theta}_n - \theta^* \right) \right|.$$

- This can be written as

$$\left| x^\top \left(\Sigma_n^\lambda \right)^{-1} \left(\sum_{i=1}^{n} x_{A_i} \eta_i \right) \right|$$

A Static Algorithm: Estimating Confidence Ellipsoids

- If we now define

$$Z_t = x^\top \left(\Sigma_n^\lambda\right)^{-1} \left(\sum_{i=1}^{t} x_{A_i} \eta_i \right),$$

  then the sequence $(Z_t)_{t \geq 1}$ becomes a martingale with bounded differences to which Azuma-Hoeffding's inequality can be applied.

- To show that $(Z_t)_{t \geq 1}$ is a martingale, write

$$Z_t = Z_{t-1} + x^\top \left(\Sigma_n^\lambda\right)^{-1} x_{A_t} \eta_t$$

  and use the property of noise that $\mathbb{E}[\eta_t \,|\, \mathcal{F}_t] = 0$.

Showing the Martingale property

Definition 0.1: A stochastic process $(Z_n)_{n\in\mathbb{N}}$ is a Martingale with respect to $(\mathcal{F}_n : n \in \mathbb{N})$ if:

1. $\mathbb{E}[|Z_n|] < \infty$.
2. Z_n is adapted to $\mathcal{F}_n$.
3. $\mathbb{E}\big[Z_{n+1} \mid \mathcal{F}_n\big] = Z_n$ for each $n \in \mathbb{N}$.

- $\mathbb{E}[Z_t \mid \mathcal{F}_t] = Z_{t-1} + x^\top \big(\Sigma_n^\lambda\big)^{-1} \mathbb{E}\Big[x_{A_t}\eta_t \mid \mathcal{F}_t\Big] = Z_{t-1}.$

A Static Algorithm: Sample Complexity

- [Soare et. al. (2014)] prove a uniform high probability bound on the estimation error.

Theorem 0.1: With probability at least $1-\delta$, for all $t \geq 1$ and for all $x \in [K]$, $\left|x^\top \left(\hat{\theta}_t - \theta^*\right)\right| \leq 2\sigma \, \|x\|_{A_t^{-1}} \sqrt{2\log\left(\frac{6n^2K}{\delta\pi^2}\right)}$.

- Using this, they were able to show a sample complexity of $\tilde{O}\left(\frac{d\log\left(\frac{K^2}{\delta}\right)}{\Delta_{\min}^2}\right)$, hiding many factors, $\Delta_{\min} = \max_{x,x'} |x - x^*|$.

Adaptivity breaks the Martingale

- In adaptive strategies, the chosen arm x_{A_t} depends on the past rewards $R_1, R_2, ..., R_{t-1}$.
- Therefore, x_{A_t} is **not** conditionally independent of the past noise $(\eta_1, ..., \eta_{t-1})$.
- Recall showing $(Z_t)_{t \geq 0}$ is a Martingale, relied **crucially** x_{A_t} and η_t being independent conditioned on the past.

$$\mathbb{E}[Z_t \mid \mathcal{F}_t] = Z_{t-1} + x^\top \left(\Sigma_n^\lambda\right)^{-1} \mathbb{E}\left[x_{A_t} \eta_t \mid \mathcal{F}_t\right] = Z_{t-1}$$

- In adaptive processes, this is no longer true since

$$\mathbb{E}\left[x_{A_t} \eta_t \mid \mathcal{F}_t\right] \neq x_{A_t} \mathbb{E}[\eta_t \mid \mathcal{F}_t].$$

Self Normalized Concentration for adaptive strategies

Theorem 0.2: [Abbasi et. al, 2011] In the Linear Bandit with conditionally -R subGaussian noise, if the ℓ_2 norm of the parameter θ is less than S and the arm selection only depends on the previous observations, then the following statement holds with probability at least $1-\delta$,

$$\left|x^\top\left(\widehat{\theta_n}^\lambda - \theta\right)\right| \leq \|x\|_{(\Sigma_n^\lambda)^{-1}} C_n,$$

where C_n is defined as $C_n = R\sqrt{2\log\left(\frac{\det(\Sigma_n^\lambda)^{\frac{1}{2}}\cdot\det(\lambda I)^{-\frac{1}{2}}}{\delta}\right)} + \sqrt{\lambda}S.$

- To minimize samples, we should pull the arm that most shrinks the confidence ellipsoid at each step.
- Static Rule: (Soare et.al, 2014):
  - Their strategy makes a sequence of selection, $\mathbf{x}_n$ to be

$$\arg\min_{\mathbf{x}_n} \max_{y \in \mathcal{Y}} \|y\|_{(\Sigma_n^\lambda)^{-1}},$$

  where $\mathcal{Y} = \{x - x' \mid x, x' \in \{x_1, ..., x_K\}\}$.
  - minimizes all the worst case directions equally.
- Adaptive rule: (Xu et. al, 2017):

$$\mathbf{x}_n^*(i_t, j_t) := \arg\min_{\mathbf{x}_n} \|y(i_t, j_t)\|_{(\Sigma_n^\lambda)^{-1}},$$

where $y(i_t, j_t) = x_{i_t} - x_{j_t}$.
- Only focus on directions between the true best arm and it's competitors.

- [Xu et. al, 2017] propose a strategy inspired from this, called LinGapE, which at each step t, first chooses two arms:
  - The arm with the largest estimated reward i_t.
  - The most ambiguous arm j_t.
  - Then, it pulls the most informative arm to estimate the gap $\left(x_{i_t} - x_{j_t}\right)^\top \theta$).
- Such an arm can be pulled greedily, where

$$a_{t+1} = \arg\min_{a \in [K]} \left(y(i_t, j_t)^\top (A_t + x_a x_a^\top)^{-1} y(i_t, j_t) \right).$$

- They propose another strategy to select the most informative arm.
- Define an "ideal" sampling proportion. Let $p^*(y(i_t, j_t))$ be the ratio of the arm i appearing in the sequence $\mathbf{x}_n^*(i_t, j_t)$ when $n \to \infty$.
- Let $T_{a(t)}$ be the number of times an arm a has been pulled until the t-th round.
- At $t+1$ pick an arm according to:

$$a_{t+1} = \arg \min_{a \in [K]: p_a^*(y(i_t, j_t)) > 0} \frac{T_{a(t)}}{p_a^*(y(i_t, j_t))}$$

Algorithm 1: LinGapE

Input: accuracy ε, confidence level δ, noise level R, norm S of unknown parameter θ, regularization parameter λ

Output: the arm $\hat{a}^*$ which satisfies stopping condition (1)

Set $A_0 \leftarrow \lambda I$, $b_0 \leftarrow \mathbf{0}$, $t \leftarrow 0$;
`// Initialize by pulling each arm once`
for $i \in [K]$ **do**
    $t \leftarrow t + 1$;
    Observe $r_t \leftarrow x_i^\top \theta + \varepsilon_t$, and update A_t and b_t;
Loop
    `// Select which gap to examine`
    $(i_t, j_t, B(t)) \leftarrow$ Select-direction(t);
    if $B(t) \leq \varepsilon$ **then**
        return i_t as the best arm $\hat{a}^*$;
    `// Pull the arm based on the gap`
    Pull the arm a_{t+1} based on (9) or (12) ;
    $t \leftarrow t + 1$;
    Observe $r_t \leftarrow x_{a_t}^\top \theta + \varepsilon_t$, and update A_t and b_t;

Algorithm 2: Select-direction

Procedure `Select-direction(`t`):`

$\hat{\theta}_t^{\lambda} \leftarrow A_t^{-1} b_t$;

$i_t \leftarrow \arg\max_{i \in [K]} (x_i^{\top} \hat{\theta}_t^{\lambda})$;

$j_t \leftarrow \arg\max_{j \in [K]} (\hat{\Delta}_t(j, i_t) + \beta_t(j, i_t))$;

$B(t) \leftarrow \max_{j \in [K]} (\hat{\Delta}_t(j, i_t) + \beta_t(j, i_t))$;

return $(i_t, j_t, B(t))$;

Sample Complexity Scaling:

- Static Allocation: Treats every arm as if the gap were the smallest gap, $\Delta_{\min} = \min_{i \neq a^*} (x_{a^*} - x_i)^\top \theta$.
  - Samples needed: $N \approx O\Big(K \frac{d}{\Delta_{\min}^2} \log\big(\frac{K}{\delta}\big)\Big)$, for a large enough λ.
- LinGAPE: Uses actual gaps.
  - $N \approx O\Big(Hd \log\Big(\frac{K^2}{\delta}\Big)\Big)$, where $H \approx \sum_{i,j \in [K]} \frac{1}{\max\Big(\varepsilon, \frac{\varepsilon + \Delta_i}{3}, \frac{\varepsilon + \Delta_j}{3}\Big)}$, where

$$\Delta_i := \begin{cases} (x_{a^*} - x_i)^\top \theta \text{ if } (i \neq a^*) \\ \arg\min_{j \in [K]} \big(x_{a^*} - x_j\big)^\top \theta \text{ if } (i = a^*) \end{cases}$$

References

- A fully adaptive algorithm for pure exploration in linear bandits [Xu. et. al, 2018].
- Best-arm identification in linear bandits. [Soare et. al, 2014].
- Improved Algorithms for Linear Stochastic Bandits [Abbasi-Yadkori et. al, 2011].