Fully adaptive linear bandit algorithm with optimal minimax regret

Kaige

February 26, 2020

Contents

1 Goal

In linear bandit, `SupLinRel` Auer and Ortner (2010), `SupLinUCB` Chu et al. (2011) and `Eliminator` Lattimore and Szepesvári (2018) achieve mini-max (worst case) regret upper bounded by $O(\sqrt{dT})$, while `LinUCB` Chu et al. (2011) and OFUL Abbasi-Yadkori et al. (2011) achieve $O(d\sqrt{T})$. `SupLinRel`, `SupLinUCB` and `Eliminator` are all based on Elimination of arms in phases which relies on pure exploration phase. This leads to high regret in practice.
The open problem is:
Could a fully adaptive algorithm, like `LinUCB`, achieve the optimal minimax regret $O(\sqrt{d\log(K)T})$?

$$R_T^{Eli} \leq \mathcal{O}(\sqrt{Td\log(\frac{k\log(T)}{\delta})}) \tag{1}$$

$$R_T^{LinUCB} \leq \mathcal{O}(d\sqrt{T}\log(T)) \tag{2}$$

PM: I think it is important to make appear the dependence on the number of arms even if it is a logarithmic dependence: if the set of arms $\mathcal{A} \in \mathbb{R}^d$ is finite of carnality K then the mini-max rate is $O(\sqrt{d\log(K)T})$ but in general (e.g. when $K \sim (1/\epsilon)^d$) the mini-max rate is $d\sqrt{T}$

2 Lemmas

Lemma 1. *Let arm i be the optimal arm and $\Delta_j = r_i - r_j \geq 0$ denote the sub-optimal gap of any arm $j \in [K]$. Suppose that for any $j \in [K]$, the index used in `LinUCB` is defined as $Index(j) = \hat{\theta}_t^T x_j + \beta||x_j||_{\mathbf{V}_t^{-1}}$. Let $t = T_j$ be the first time the following condition holds, `LinUCB` will not select arm j for any $t \geq T_j$ with high probability.*

$$\Delta_j > 2\beta||\mathbf{x}_j||_{\mathbf{V}_t^{-1}} \tag{3}$$

Lemma 2. *Let arm i be the optimal arm and $\Delta_j = r_i - r_j \geq 0$ denote the sub-optimal gap of an arm $j \in [K]$. Also, let $\hat{r}_{i,t}$ and $\hat{r}_{j,t}$ denote the estimate reward of i and j at time t. Suppose $|r_i - \hat{r}_{i,t-1}| \leq \beta||x_i||_{\mathbf{V}_{t-1}^{-1}}$. Let $P = \max_{i\in[K]} \hat{r}_{i,t-1} - \beta||x_i||_{\mathbf{V}_{t-1}^{-1}}$ denote the highest lower bound at time t. If the following holds, arm j is a sub-optimal arm with high probability.*

$$P \geq \hat{r}_{j,t-1} + \beta||\mathbf{x}_j||_{\mathbf{V}_{t-1}^{-1}} \tag{4}$$

Lemma 3. *If one arm is identified as a sub-optimal arm by Lemma 2. This arm remains to be sub-optimal due to the monotonic property of upper bound and lower bound. It means that the arm will not be selected by `LinUCB` anymore.*

3 Analysis of `LinUCB`

3.1 Questions

Does `LinUCB` keep selecting suboptimal arm when it is unnecessary?
[KG: NO].

1. Under which condition, an arm will not be selected anymore (eliminated implicitly)? Lemma 1.

2. Under which condition, an arm can be confirmed to be a sub-optimal arm? Lemma 2.

3. If an arm satisfies Lemma 2, would it be selected again? [KG: NO. Lemma 2 indicates the upper bound of arm i is lower than the lower bound of arm j which means the upper bound arm j is definitely larger than that of arm i. In this case, arm i is not selected. In addition, if the upper bound is a monotonic decreasing function and the lower bound is a monotonic increasing function w.r.t t, The upper bound of arm i will remain to be lower than the lower bound of arm j. It means `LinUCB` will not select arm i anymore.]

4. How to prove the monotonic property of upper bound and lower bound?

   Proof. The upper bound at time t is defined as

$$UP_{i,t} = \hat{r}_{i,t} + \beta||x_i||_{V_t^{-1}} = \hat{\theta}_t^T x_i + \beta||x_i||_{V_t^{-1}} \tag{5}$$

   In the same fashion, at time $t+1$,

$$UP_{i,t+1} = \hat{r}_{i,t+1} + \beta||x_i||_{V_{t+1}^{-1}} = \hat{\theta}_{t+1}^T x_i + \beta||x_i||_{V_{t+1}^{-1}} \tag{6}$$

   Therefore,

$$UP_{i,t} - UP_{i,t+1} = (\hat{\theta}_t - \hat{\theta}_{t+1})^T x_i + \beta(||x_i||_{V_t^{-1}} - ||x_i||_{V_{t+1}^{-1}}) \tag{7}$$

   [KG: We want to proof $UP_{i,t} - UP_{i,t+1} \geq 0$ for any $t \in [1, T]$.]
   According to Causchy-Shcraw inequality, we know that for any PSD matrix M,

$$|(\hat{\theta}_t - \hat{\theta}_{t+1})^T x_i| \leq ||\hat{\theta}_t - \hat{\theta}_{t+1}||_M ||x_i||_{M^{-1}} \tag{8}$$

   We also have $||\hat{\theta}_t - \hat{\theta}_{t+1}||_M \leq \beta$, so

$$-\beta||x_i||_{M^{-1}} \leq (\hat{\theta}_t - \hat{\theta}_{t+1})^T x_i \leq \beta||x_i||_{M^{-1}} \tag{9}$$

   Then

$$UP_{i,t} - UP_{i,t+1} \geq \beta||x_i||_{V_t^{-1}} - \beta||x_i||_{V_{t+1}^{-1}} - \beta||x_i||_{M^{-1}} \tag{10}$$

   Let $M = V_{t+1} - V_t$, [KG: We need to prove]

$$||x_i||_{V_t^{-1}} - ||x_i||_{V_{t+1}^{-1}} \geq ||x_i||_{(V_{t+1}-V_t)^{-1}} \tag{11}$$

   which is [KG: Looks trivial to prove]

$$||x_i||_{V_t^{-1}} \geq ||x_i||_{V_{t+1}^{-1}} + ||x_i||_{(V_{t+1}-V_t)^{-1}} \tag{12}$$

   The same applies to low bound. □

5. Why upper bounds are maintained to be similar in `LinUCB`? what are the benefits and drawbacks? [KG: Not clear]

6. In `LinUCB`, how long it takes before arm i satisfies Lemma 2?

3.2 Conclusion of `LinUCB`

1. `LinUCB` indeed eliminates arms since it will not select an arm anymore if the arm is identified as a sub-optimal arm according to Lemma 2 (the same as `Eliminator`).

2. However, comparing with `Eliminator`, `LinUCB` takes much longer time to identify a sub-optimal arm.

3.3 Open problem `LinUCB`

Since we already prove that `LinUCB` is indeed an eliminator-type algorithm, Can we upper bound its regret in the same way as in `Eliminator`, which might leads to tighter bound? Basically, the regret can be analysis in two steps:

1. Upper bound the time when an arm is eliminated.

2. After the elimination, the instantaneous regret is upper bounded by the regret of eliminated arm.

4 Analysis of `Eliminator`

1. **Uniform upper bound of estimate reward error.**
   Arm j is eliminated if

$$\max_{i\in[K]} \hat{\theta}_t^T x_i - \epsilon_\ell > \hat{\theta}_t^T x_j + \epsilon_\ell \tag{13}$$

   where $\epsilon_\ell \geq \sqrt{2\log\frac{1}{\delta}}||x_i||_{V_t^{-1}}$.
   Lemma 2 proves that if arm $j \in [K]$ satisfies the following condition, it can be eliminated.

$$\max_{i\in[K]} \hat{\theta}_t^T x_i - \beta||x_i||_{\mathbf{V}_t^{-1}} > \hat{\theta}_t^T x_j + \beta||x_j||_{\mathbf{V}_t^{-1}} \tag{14}$$

   Does Eq. 14 eliminate more arms than Eq. 13? [KG: Yes].

   Proof. Note that ϵ_ℓ is the upper bound of $\beta||x_i||_{V_\ell^{-1}}$. i.e., $\gamma||x_i||_{V_\ell^{-1}} \leq \epsilon_\ell$.
   Thus, it is clearly that

$$\hat{\theta}_t^T x_i - \beta||x_i||_{V_\ell^{-1}} \geq \hat{\theta}_t^T x_i - \epsilon_\ell \tag{15}$$

   and

$$\hat{\theta}_t^T x_i + \beta||x_i||_{V_\ell^{-1}} \leq \hat{\theta}_t^T x_i + \epsilon_\ell \tag{16}$$

   Therefore, it is possible that a suboptimal arm is not eliminated by Eq. 13 while eliminated by Eq. 14. □

2. **Not adaptive to the problem structure.**
   As the phase length is pre-defined as well as ϵ_ℓ. It is possible that no arms are eliminated during first phases which leads to a large amount of regret. However, it is also possible that

all sub-optimal arms are eliminated during one phase.
To avoid the waste of phase, only start a new phase when at least one arm can be eliminated by Eq. 14.

3. **Re-initializes the learning parameter at the beginning of each phase.**
   Make use of history information.

5 Successive Eliminator (SE)

5.1 Algorithm

SE keeps pure exploration and eliminating arms without restarting a new phase. Formally, at time t,

$$i_t = \arg\max_{i\in[K]} ||x_i||_{\mathbf{V}_t^{-1}} \tag{17}$$

and

$$\arg\max_{i\in[K]} \hat{r}_{i,t} - \beta||x_i||_{\mathbf{V}_t^{-1}} > \hat{r}_{j,t} + \beta||x_j||_{\mathbf{V}_t^{-1}} \tag{18}$$

5.2 Analysis

To analyze the regret, we introduce the notion of phase, a new phase starts after one arm is eliminated. $V_\ell = \sum_{t=T_{\ell-1}}^{T_\ell} x_t x_t^T$.

Theorem 1. *How to upper bound the regret of SE?*

Proof. SE assumes $|\theta^T x_i - \hat{\theta}_t^T x_i| \leq \beta||x_i||_{V_t^{-1}}$. During each phase, it is true that $|\theta^T x_i - \hat{\theta}_\ell^T x_i| \leq \beta||x_i||_{V_\ell^{-1}}$. This is also true that

$$|\theta^T x_i - \hat{\theta}_t^T x_i| \leq |\theta^T x_i - \hat{\theta}_\ell^T x_i|,\ \forall\ell \tag{19}$$

Furthermore,

$$|\theta^T x_i - \hat{\theta}_t^T x_i| \leq \arg\min_{\ell} |\theta^T x_i - \hat{\theta}_\ell^T x_i| \tag{20}$$

Note that $V_t = \sum_{\ell=1}^{L} V_\ell$ and $V_t^{-1} = (\sum_{\ell=1}^{L} V_\ell)^{-1}$.

$$||x_i||_{V^{-1}} = ||V^{-1/2}x_i||_2 \leq ||V^{-1/2}||_2||x_i||_2 \leq \sqrt{Tr(V^{-1/2})}||x_i||_2 \tag{21}$$

$$\sqrt{Tr(V_t^{-1/2})} = \sqrt{Tr((\sum_{\ell=1}^{L} V_\ell)^{-1/2})} \tag{22}$$

□

6 Linear Successive Eliminator (LSE)

6.1 Algorithm

1. The arm is selected by $x_t = \arg\max_{i\in[K]} ||x_i||_{V_\ell^{-1}}$ where $V_\ell = \sum_{\tau=T_{\ell-1}}^{t} x_\tau x_\tau^T$. Note that at the beginning of each phase ℓ, V_ℓ is initialized. i.e., $V_\ell = \alpha I$.
2. The gradient-decent update is $\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma(y_t - \hat{y}_t)x_t$.
3. The lower bound and upper bound are $\hat{\theta}_t^T x_i - \beta||x_i||_{V_t^{-1}}$ and $\hat{\theta}_t^T x_i + \beta||x_i||_{V_t^{-1}}$ where $V_t = \sum_{\tau=1}^{t} x_\tau x_\tau^T$ and $\beta = \sqrt{\alpha} + \sqrt{2\log\frac{1}{\delta} + d\log(\frac{d\alpha+T}{d\alpha})}$ which are the same V_t and β as in LinUCB.
4. Eliminate arm i if exist arm j such that $\hat{\theta}_t^T x_j - \beta||x_j||_{V_t^{-1}} > \hat{\theta}_t^T x_i + \beta||x_i||_{V_t^{-1}}$.

Remark 1. *Three differences from Eliminator:*

1. *The elimination condition is based on UCB of each arm instead of a uniform upper bound. i.e, $\epsilon_\ell > \beta||x_i||_{V_t^{-1}}, \forall i$. Formally, arm i is eliminated at time t if*
$$\hat{\theta}_t^T x_j - \beta||x_j||_{V_t^{-1}} > \hat{\theta}_t^T x_i + \beta||x_i||_{V_t^{-1}} \tag{23}$$
ϵ_ℓ might delay the time when one arm is eliminated.
2. *Historic information is leveraged via gradient-descent update.*
a), the estimation is remained by gradient descent update; b) the uncertainty is maintained as $\beta||x_i||_{V_t^{-1}}$.
3. *The phase length is not pre-defined. A new phase starts immediately once any arm is eliminated.*

6.2 Regret Analysis

6.3 Results

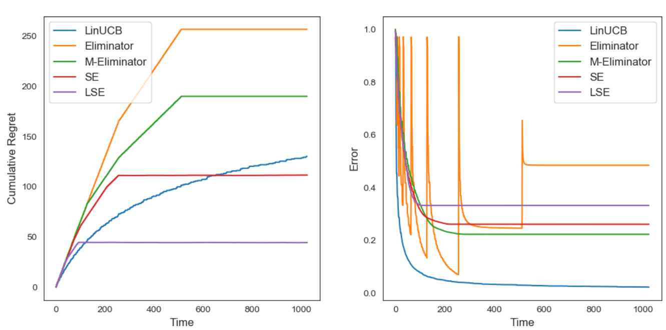

Figure 1: Algorithms: Regret and Error

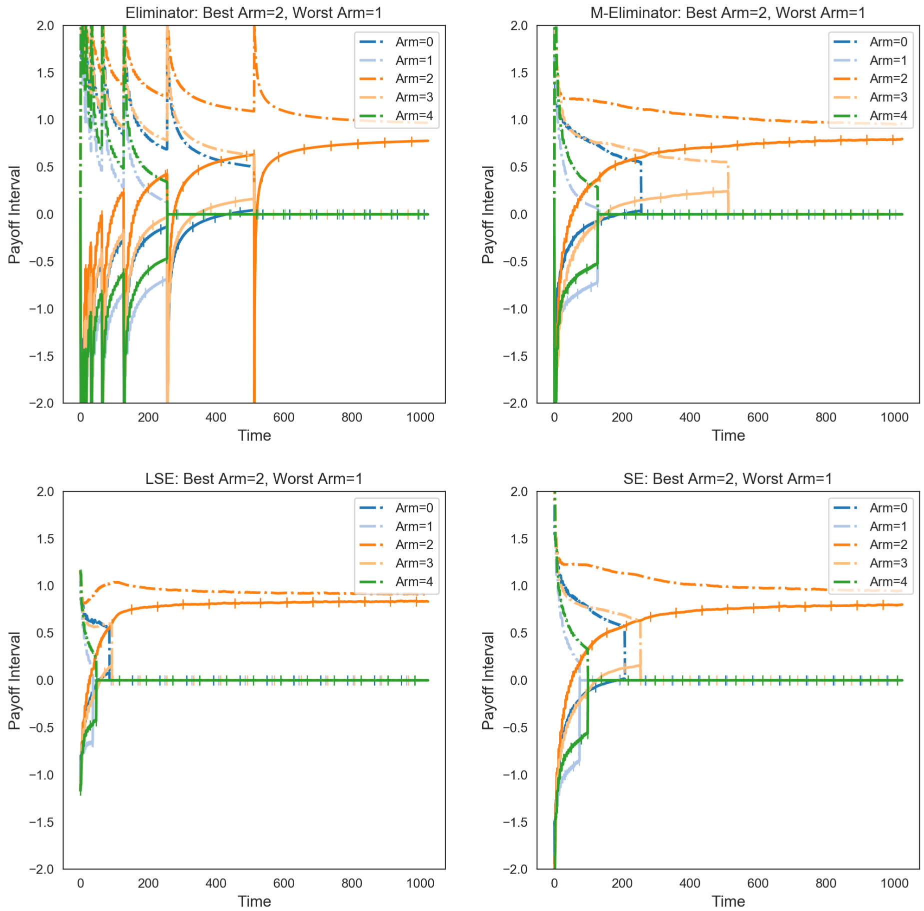

Figure 2: Algorithms: Upper Bound and Lower Bound

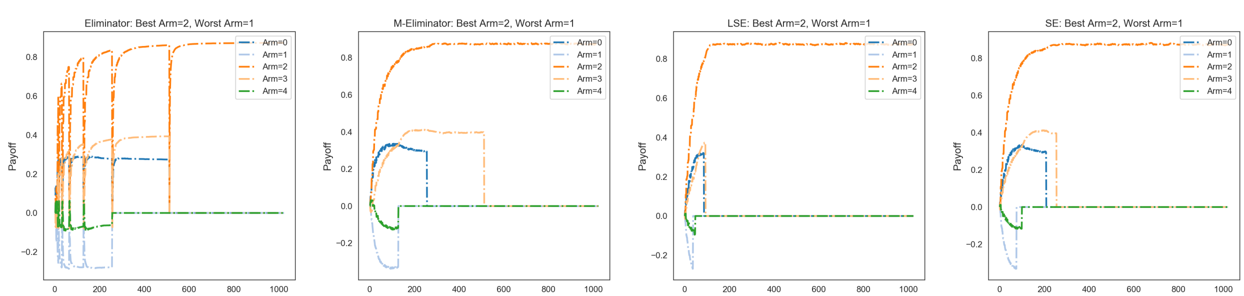

Figure 3: Algorithms: Estimated Payoffs

7 Appendix

7.1 LinUCB

Algorithm 1: LinUCB

Input : α, β, T, σ

Initialization : $\hat{\theta}_0 = \mathbf{0} \in \mathbb{R}^d$ and $\mathbf{V}_0 = \alpha \mathbf{I} \in \mathbb{R}^{d \times d}, \mathbf{B}_0 = \mathbf{0} \in \mathbf{R}^d$.

for $t \in [1, T]$ **do**

1. Select arm $i_t = \arg\max_{i \in [K]} \hat{\theta}_{t-1}^T x_i + \beta ||x_i||_{\mathbf{V}_{t-1}^{-1}}$.
2. Receive the reward y_t.
3. Update $\hat{\theta}_t$:
$$\hat{\theta}_t = \mathbf{V}_t^{-1} \mathbf{B}_t \tag{24}$$
Where $\mathbf{V}_t = \mathbf{V}_{t-1} + \mathbf{x}_{i_t} \mathbf{x}_{i_t}^T$; $\mathbf{B}_t = \mathbf{B}_{t-1} + \mathbf{x}_{i_t} y_t$.

end

7.2 `Eliminator`

Algorithm 2: `Eliminator`

Input : $\mathcal{A} \in \mathbb{R}^d$ and δ

1. Set $\ell = 1$ and let $\mathcal{A}_\ell = \mathcal{A}$.
2. Let $t_\ell = t$ be the current timestep and the find G-optimal design $\pi_\ell \in \mathcal{P}(\mathcal{A}_\ell)$ with $Supp(\pi_\ell) \leq \frac{d(d+1)}{2}$ that maximizes

$$\log detV(\pi_\ell) \; subject \; to \sum_{a \in \mathcal{A}_\ell} \pi_\ell(a) = 1 \tag{25}$$

3. Let $\epsilon_\ell = 2^{-\ell}$ and

$$T_\ell(a) = \lceil \frac{2d\pi_\ell(a)}{\epsilon_\ell^2} \log \frac{k\ell(\ell+1)}{\delta} \rceil \; and \; T_\ell = \sum_{a \in \mathcal{A}_\ell} T_\ell(a) \tag{26}$$

4. Choose each action $a \in \mathcal{A}_\ell$ exactly T_ℓ times.
5. Calculate empirical estimate:

$$\hat{\theta}_\ell = \mathbf{V}_\ell^{-1} \sum_{t=t_\ell}^{t_\ell + T_\ell} \mathbf{A}_t \mathbf{X}_t \; with \; \mathbf{V}_\ell = \sum_{a \in \mathcal{A}_\ell} T_\ell(a) a a^T \tag{27}$$

6. Eliminate low rewarding arms:

$$\mathcal{A}_{\ell+1} = \{a \in \mathcal{A}_\ell : \max_{b \in \mathcal{A}_\ell} \langle \hat{\theta}_\ell, b - a \rangle \leq 2\epsilon_\ell\} \tag{28}$$

7. $\ell \leftarrow \ell + 1$ and Go to step 1.

7.3 SpectralEliminator

Algorithm 3: SpectralEliminator

Input : $\mathcal{A} \in \mathbb{R}^d$ and δ

Initialization : $\ell = 1$, $T_\ell = 2^{\ell-1}$.

1. **for** $t \in [T_\ell, T_{\ell+1} - 1]$ **do**
   Select arm: $i_t = \arg\max_{i\in[K]} ||x_i||_{\mathbf{V}_t^{-1}}$.
   end
2. Update: $\hat{\theta}_\ell = \mathbf{V}_\ell^{-1}\mathbf{X}_\ell^T\mathbf{Y}_\ell$ where $\mathbf{V}_\ell = \sum_{\tau=T_{\ell-1}}^{\tau=T_\ell-1} x_\tau x_\tau^T$.
3. Find: $P = \max_{i\in[K]} \hat{\theta}_\ell^T x_i - \gamma||x_i||_{\mathbf{V}_\ell^{-1}}$.
4. Eliminate arm $j \in [K]$, if
$$P - \hat{\theta}_\ell^T x_j + \gamma||x_j||_{\mathbf{V}_\ell^{-1}} > 0 \tag{29}$$
5. Set $\mathbf{X}_{\ell+1} = \mathbf{0}, \mathbf{Y}_{\ell+1} = \mathbf{0}$ and $\mathbf{V}_{\ell+1} = \alpha\mathbf{I}$.

7.4 LSE

Algorithm 4: LSE

Input : $\mathcal{A} \in \mathbb{R}^d$, σ, δ, γ, β, λ

Initialization : $\hat{\theta}_0 = \mathbf{0}$ and $\mathbf{V}_0 = \alpha\mathbf{I}$, $\mathbf{B}_0 = \mathbf{0}$, $\mathbf{M}_0 = \alpha\mathbf{I}$,
$\mathcal{M}_l = \alpha\mathbf{I}$, $\mathcal{B}_l = \mathbf{0}$, $\tilde{\theta}_l = \mathbf{0}$, $\tilde{r}_{i,l} = 0$, $\tilde{\psi}_{i,l} = 0$ and $T_l = 0$ for $l \in [0, K]$,
$l = 1$.

for $t \in [1, T]$ **do**

1. Select the arm $i_t = \arg\max_{i \in [K]} ||x_i||_{\mathbf{V}_{t-1}^{-1}}$ and receive y_t.
2. Update $\mathbf{V}_t = \sum_{\tau=1}^{t} x_\tau x_\tau^T$.
3. Find $\mathbf{M}_t = \sum_{\tau=T_{l-1}}^{t} x_\tau x_\tau^T$ and $\mathbf{B}_t = \sum_{\tau=T_{l-1}}^{t} x_\tau y_\tau$.
4. Update $\hat{\theta}_t = \mathbf{M}_t^{-1}\mathbf{B}_t$, $\hat{r}_{i,t} = \hat{\theta}_t^T x_i$ and $\hat{\phi}_{i,t} = ||x_i||_{\mathbf{M}_t^{-1}}$
5. Find $P = \max_{i \in [K]} \left(\Psi_{i,l-1} + \lambda\big(\hat{r}_{i,t} - \beta\hat{\phi}_{i,t}\big) \right)$ where $\Psi_{i,l-1}$ and $\Phi_{i,l-1}$ defined in Eq. ??.
6. **if** *Any arm j is eliminated by* $P - \left(\Psi_{j,l-1} + \lambda\big(\hat{r}_{j,t} + \beta\phi_{j,t}\big) \right) > 0$ **then**
   (a) Record $T_l = t$, $\mathcal{M}_l = \mathbf{M}_t$, $\mathcal{B}_l = \mathbf{B}_t$.
   (b) Record $\tilde{\theta}_l = \mathcal{M}_l^{-1}\mathbf{B}_l$, $\tilde{r}_{i,l} = \tilde{\theta}_l^T x_i$ and $\tilde{\phi}_{i,l} = ||x_i||_{\mathcal{M}_l}$.
   (c) $l = l + 1$.

   end
7. **if** *No arm is eliminated* **then**
$$l = l \tag{30}$$
   end

end

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Auer, P. and Ortner, R. (2010). Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65.

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214.

Lattimore, T. and Szepesvári, C. (2018). Bandit algorithms. *preprint*.