Finding All ϵ-Good Arms in Stochastic Bandits: Fuzzy $\texttt{(ST)}^2$ Algorithm

David Ciprut
343898656
davidciprut@tauex.tau.ac.il

Shlomo Tannor
314389248
shlomotannor@mail.tau.ac.il

Abstract

The pure-exploration problem in stochastic multi-armed bandits aims to find one or more arms with the largest (or near largest) means. While most existing work focuses on finding the best arm, or top-k arms, recent work has explored a new yet intuitive problem of identifying all ϵ-good arms. In this project we present a variation of the recently introduced algorithm $\texttt{(ST)}^2$, called $\texttt{Fuzzy (ST)}^2$, which has better performance on edge cases. The simulation code can be found here.

1 Finding All ϵ-good Arms in Stochastic MAB Problem

Our project is based on a recent paper [1] in which the authors propose a new multi-armed bandit exploration problem where the objective is to return *all* arms that are ϵ-good with respect to the best arm. We assume the additive definition of ϵ-good arms in this paper. The formal definition of the problem is as follows.

Fix $\epsilon > 0$ and a failure probability $\delta > 0$. Let $\nu := \{\rho_1, ..., \rho_n\}$ be an instance of n distributions (or arms) with 1-sub-Gaussian distributions having unknown means

$$\mu_1 > ... > \mu_n \text{ where } \mu_1 = \max_{1 \le i \le n} \mu_i$$

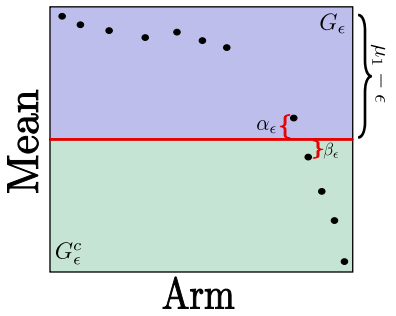
Additionally, we define the set

$$G_\epsilon(\nu) = \{i \,|\, \mu_i \ge \mu_1 - \epsilon\}$$

Then the goal is to return a set $\hat{G}_\epsilon$ such that

$$\mathbb{P}\left(\hat{G}_\epsilon = G_\epsilon\right) \ge 1 - \delta$$

using as few total samples as possible.
Throughout, we will make use of the following quantaties:



$$\alpha_\epsilon = \min_{i \in G_\epsilon} \mu_i - (\mu_1 - \epsilon) \qquad \beta_\epsilon = \min_{i \in G_\epsilon^c} (\mu_1 - \epsilon) - \mu_i \qquad \Delta_i = \mu_1 - \mu_i$$

Definition 1.1. (all-ϵ problem) An algorithm for the all-ϵ problem is δ-*PAC* if *(a)* the algorithm has a finite stopping time τ, *(b)* at time τ it recommends a set $\hat{G}$ such that with probability at least $1 - \delta$, $\hat{G} = G_\epsilon$.

Theorem 1.1. ***(Lower Bound)*** *Fix $\delta, \epsilon > 0$. Consider n arms, such that the i^{th} arm's mean reward is distributed according to $\mathcal{N}(\mu_i, 1)$. Any δ-PAC algorithm satisfies:*

$$\mathbb{E}[\tau] \geq 2 \sum_{i=1}^{n} \max \left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \right\} log \left(\frac{1}{2.4\delta} \right) \tag{1}$$

2 Previous Results

Mason et al. [1] propose two algorithms for solving the problem, $\texttt{(ST)}^2$ and FAREAST.

Given positive *slack* γ, $\texttt{(ST)}^2$ is meant to return a set containing all ϵ-good arms and none worse than $(\epsilon + \gamma)$ with probability $1 - \delta$. This is done by sampling the threshold, and splitting the threshold (using the LUCB algorithm [2]) at each iteration. $\texttt{(ST)}^2$ has the following theoretical guarantees.

Theorem 2.1. *Fix $\epsilon < 0$, $0 < \delta \leq \frac{1}{2}$, $\gamma \leq 16$ and an instance ν such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leq 8$ for all i. With probablity at least $1 - \delta$, there is a constant c_1 such that $\texttt{(ST)}^2$ returns a set $\hat{G}$ such that $G_\epsilon \subset \hat{G} \subset G_{(\epsilon+\gamma)}$ in at most the following number of samples:*

$$c_1 log \left(\frac{n}{\delta} \right) \sum_{i=1}^{n} \max \left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \beta_\epsilon - \mu_i)^2} \right\} \wedge \frac{1}{\gamma^2} \tag{2}$$

Given a positive slack γ, we are allowed to return an arm that is $(\epsilon + \gamma)$-good. Thus a confidence width less than $\Omega(\gamma)$ on any arm is not needed, resulting in the $\frac{1}{\gamma^2}$ term. In particular this prevents unbounded sample complexities if there is an arm at the threshold $\mu_1 - \epsilon$. For $\gamma = 0$, the first two terms inside the max are also present in the lower bound. When α_ϵ is within a constant factor of β_ϵ, the second and third term in the max have the same order, and the upper bound matches the lower bound up to a $log(n)$ factor. If $\beta_\epsilon \ll \alpha_\epsilon$, (2) has a different scaling than the lower bound. In such restrictive settings the upper bound above can be significantly larger than the lower bound. In order to tackle this gap the paper [1] has proposed the FAREAST algorithm which has the following theoretical guarantees.

Theorem 2.2. *Fix $\epsilon > 0$, $0 < \delta < \frac{1}{8}$, and an instance ν of n arms such that $\max(\Delta_i, |\epsilon - \Delta_i) \leq 8$ for all i. There exists an event E such that $\mathbb{P}(E) \geq 1 - \delta$ and on E, FAREAST terminates and returns G_ϵ. Letting T denote the number of samples taken, for a constant c_3:*

$$\mathbb{E}[\mathbb{1}_E] \leq \left[c_3 \sum_{i=1}^{n} \max \left\{ \frac{1}{(\mu_1 - \epsilon - \mu_i)^2}, \frac{1}{(\mu_1 + \alpha_\epsilon - \mu_i)^2} \right\} log \left(\frac{n}{\delta} \right) \right] + c_3 \sum_{i \in G_\epsilon^c} \frac{c'' n}{(\mu_1 - \epsilon - \mu_i)^2} \tag{3}$$

Additionally for $\gamma \leq 16$ FAREAST terminates on E and returns a set $\hat{G}$ such that $G_\epsilon \subset \hat{G} \subseteq G_{(\epsilon+\gamma)}$ in a number of samples no more than a constant times (2), the complexity of $\texttt{(ST)}^2$.

In the paper [1] the authors show the following results for both algorithms:

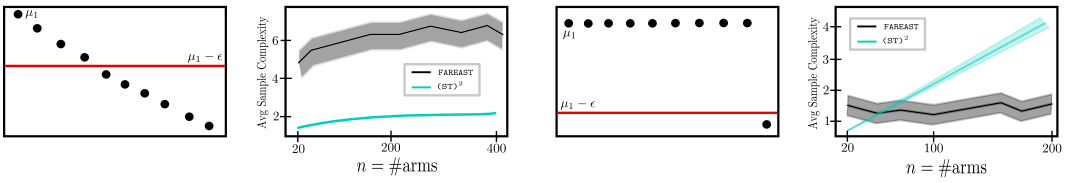


Figure 1: $\texttt{(ST)}^2$ is significantly better on the typical case than FAREAST, but can have much worse results on edge cases.

The $\texttt{(ST)}^2$ algorithm has good performance in typical cases, but has bad performance when $\beta_\epsilon \ll \alpha_\epsilon$. The FAREAST algorithm's upper bound matches the lower bound but has bad constant factors due to repeated use of median elimination [3], and its performance is not as good as $\texttt{(ST)}^2$ in typical cases.

3 $\texttt{(ST)}^2$ with Negative Slack

Our approach is to allow negative slack and thus avoid the worst case scenarios. We will provide an algorithm which returns $\hat{G}$ such that $G_{\epsilon-\gamma} \subset \hat{G} \subset G_{\epsilon+\gamma}$. In other words, we will return an approximation of G_ϵ up to a γ factor.

3.1 Fuzzy $\texttt{(ST)}^2$ Algorithm

We propose the following algorithm:

Algorithm 1: Fuzzy $\texttt{(ST)}^2$ Algorithm

Input: $\epsilon > 0$, $\delta > 0$, $\gamma_1 > 0$, $\epsilon \geq \gamma_2 > 0$, instance ν

Initialize: Pull each arm once, initialize $T_i \leftarrow 1$, update $\overline{\mu}_i$, `update_thresh = False`

Empirically good arms $\hat{G} = \{i \ : \ \overline{\mu}_i \geq \max_j \overline{\mu}_j - \epsilon\}$

Define $U_t = \max_j \overline{\mu}_j(T_j) + C_{\frac{\delta}{n}}(T_j) - \epsilon - \gamma_1$ and $L_t = \max_j \overline{\mu}_j(T_j) - C_{\frac{\delta}{n}}(T_j) - \epsilon$

Define $d = 0$

Known arms: $\mathcal{K} = \{i \ : \ \overline{\mu}_i(T_i) + C_{\frac{\delta}{n}} < L_t \text{ or } \overline{\mu}_i(T_i) - C_{\frac{\delta}{n}}(T_i) > U_t\}$

while $\mathcal{K} \neq [n]$ **do**

  pull arm $i_1(t) = \arg\min_{i \in \hat{G} \setminus K} \overline{\mu}_i(T_i) - C_{\frac{\delta}{n}}(T_i)$, update $T_{i_1}, \overline{\mu}_{i_1}$

  pull arm $i_2(t) = \arg\max_{i \in \hat{G}^c \setminus K} \overline{\mu}_i(T_i) + C_{\frac{\delta}{n}}(T_i)$, update $T_{i_2}, \overline{\mu}_{i_2}$

  pull arm $i^\star(t) = \arg\max_i \overline{\mu}_i(T_i) + C_{\frac{\delta}{n}}(T_i)$, update $T_{i^\star}, \overline{\mu}_{i^\star}$

  $d \leftarrow \overline{\mu}_{i_1} - \overline{\mu}_{i_2}$

  if $C_{\frac{\delta}{n}}(T_{i_1}) + C_{\frac{\delta}{n}}(T_{i_2}) + C_{\frac{\delta}{n}}(T_{i^\star}) \leq \min\left\{\gamma_2, \frac{d}{4}\right\} \ \wedge$ `update_thresh == False`

    $\epsilon \leftarrow \overline{\mu}_{i^\star} - \left(\frac{\overline{\mu}_{i_1} + \overline{\mu}_{i_2}}{2}\right)$

    `update_thresh` $\leftarrow$ `True`

  update bounds L_t, U_t, sets $\hat{G}, \mathcal{K}$

return the set of good arms $\{i \ : \ \overline{\mu}_i(T_i) - C_{\frac{\delta}{n}} > U_t\}$

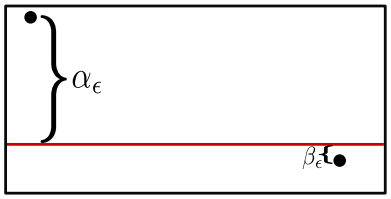
In the algorithm $C_{\frac{\delta}{n}}$ is the confidence radius which satisfies:

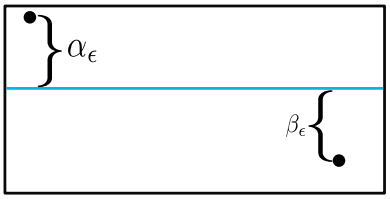
$$\mathbb{P}\left(\bigcup_{t=1}^{\infty} |\overline{\mu}_i(t) - \mu_i| > C_{\frac{\delta}{n}}(t)\right) \leq \frac{\delta}{n} \tag{4}$$

For this to work, we take:

$$C_\delta(t) = \sqrt{\frac{4log\left(\frac{log_2(2t)}{\delta}\right)}{t}} \tag{5}$$

The idea of our algorithm is as follows. If we are in the typical case, then we would like to run $\texttt{(ST)}^2$ as is. If we are in the edge case, then we would like to change ϵ's value so that $\alpha_{\epsilon_{\text{new}}} \approx \beta_{\epsilon_{\text{new}}}$. To illustrate the idea, let's assume that we are in the edge case as in the leftmost figure below. We change ϵ's value once the criterion on line 11 is satisfied. When the condition is satisfied we simply replace the threshold in the middle of the estimates of the two closest arms (one from above, the other from below) as in the rightmost figure below.





Since we do not know what α_ϵ and β_ϵ are, we will work with their estimators. According to the law of large numbers, after enough iterations, we get:

$$\alpha_\epsilon \approx \overline{\mu}_{i_1}(T_{i_1}) - (\overline{\mu}_{i^\star}(T_{i^\star}) - \epsilon)$$

$$\beta_\epsilon \approx (\overline{\mu}_{i^\star}(T_{i^\star}) - \epsilon) - \overline{\mu}_{i_2}(T_{i_2})$$

$$\mu_1 \approx \overline{\mu}_{i^\star}$$

At that stage we change our ϵ so that the following is satisfied:

$$\overline{\mu}_{i_1}(T_{i_1}) - (\overline{\mu}_{i^\star}(T_{i^\star}) - \epsilon_{new}) = (\overline{\mu}_{i^\star}(T_{i^\star}) - \epsilon_{new}) - \overline{\mu}_{i_2}(T_{i_2})$$

Notice that this "equality" is satisfied if and only if:

$$2\epsilon_{new} = 2\overline{\mu}_{i^\star}(T_{i^\star}) - \left(\overline{\mu}_{i_1}(T_{i_1}) + \overline{\mu}_{i_2}(T_{i_2})\right) \iff \epsilon_{new} = \overline{\mu}_{i^\star}(T_{i^\star}) - \left(\frac{\overline{\mu}_{i_1}(T_{i_1}) + \overline{\mu}_{i_2}(T_{i_2})}{2}\right)$$

And so we get $\alpha_{\epsilon_{\text{new}}} \approx \beta_{\epsilon_{\text{new}}}$, as desired.

3.1.1 Upper Bound of the Fuzzy $\texttt{(ST)}^2$ Algorithm

We claim the following theorem:

Theorem 3.1. *Fix $\epsilon < 0$, $0 < \delta \leq \frac{1}{2}$, $\gamma_1 \leq 16$, $\gamma_2 \leq \epsilon$ and an instance ν such that $\max(\Delta_i, |\epsilon - \Delta_i|) \leq 8$ for all i. With probablity at least $1-\delta$, there is a constant c_1 such that Fuzzy $\texttt{(ST)}^2$ algorithm returns a set $\hat{G}$ such that $G_{(\epsilon-\gamma_2)} \subset \hat{G} \subset G_{(\epsilon+\max\{\gamma_1,\gamma_2\})}$ in at most the following number of samples:*

$$c_1 log\left(\frac{n}{\delta}\right)\sum_{i=1}^{n}\max\left\{\frac{1}{(\mu_1-\epsilon-\mu_i)^2}, \frac{1}{(\mu_1+\min\{\gamma_2, \max\{\alpha_\epsilon, \beta_\epsilon\}\}-\mu_i)^2}\right\} \wedge \frac{1}{\gamma_1^2} \tag{6}$$

To prove this theorem at first we must bound the total number of samples. In our case we divide the algorithm into two stages, before the ϵ update, and after.

After the ϵ update, provided that $\alpha_{\epsilon_{\text{new}}} \approx \beta_{\epsilon_{\text{new}}}$ we get a similar expression to the upper bound of $\texttt{(ST)}^2$ complexity, but we can now substitute $\min(\alpha_\epsilon, \beta_\epsilon)$ for their average, or their max (up to a constant).

The number of steps it takes until we can do the ϵ update can be bounded similarly. We simply want to make sure that we have good enough estimates of the threshold, α_ϵ, and β_ϵ.

For our needs, an estimate of magnitude $min(\gamma_2, \frac{d}{4})$ for the sum of the bounds will suffice, where $d = \overline{\mu}_{i_1} - \overline{\mu}_{i_2}$. The required sample complexity can be derived with similar methods to those used in the original paper, substituting $\min(\frac{\gamma_2}{3}, \frac{\alpha_\epsilon + \beta_\epsilon}{12})$ for ω.

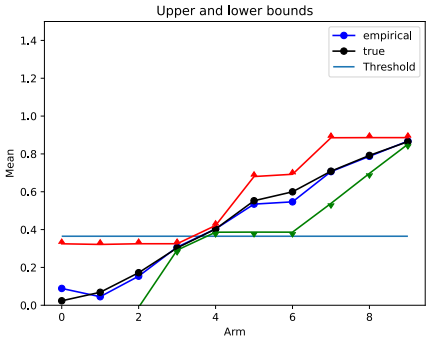
Thus, the total bound on sample complexity can be derived.

Finally, since it is guaranteed that there are no arms in between i_1 and i_2 at the time of the threshold update, and because the bounds are smaller than γ, we get $G_{\epsilon-\gamma_2} \subset \hat{G} \subset G_{\epsilon+\gamma_2}$ as required.

3.2 Simulations

We have run simulations of the Fuzzy $\texttt{(ST)}^2$ algorithm and compared it to $\texttt{(ST)}^2$ and `FAREAST` algorithm and got the following results.

3.2.1 Comparison in the Typical Case



In the typical case the Fuzzy $\texttt{(ST)}^2$ did not enter the if block on line 11, and thus was identical to $\texttt{(ST)}^2$ algorithm and we got the figure on the left. In this case the algorithm ran for $363,064$ iterations. Since this is the typical case, as the paper [1] has shown, it has much better results than `FAREAST` algorithm.

3.2.2 Comparison on the Edge Case

In the edge case the Fuzzy $\texttt{(ST)}^2$ algorithm did enter the if block and we got the following results:

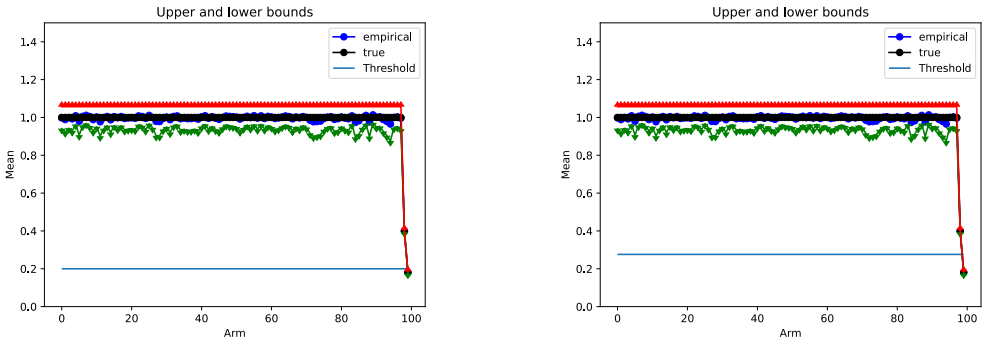


Figure: Before threshold change on the left, after threshold change on the right

As it can be seen from the figures above, the threshold is in between the two bottom arms, and even though the threshold changed (our ϵ changed) the algorithm still returned the same set of arms. Our algorithm ran for $4,195,390$ iterations.

For the $\texttt{(ST)}^2$ and FAREAST algorithms we got the following figures:

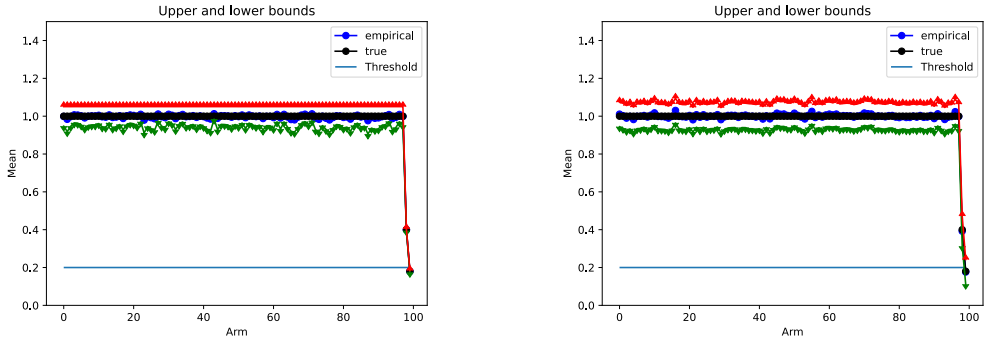


Figure: $\texttt{(ST)}^2$ on the left, FAREAST on the right

In this case the $\texttt{(ST)}^2$ algorithm ran for $5,348,905$ iterations and the FAREAST algorithm ran for $4,440,847$ iterations. Thus our algorithm has the benefit of both algorithms.

3.2.3 Symmetric Edge Case ($\alpha_\epsilon \ll \beta_\epsilon$)

Note that our algorithm can achieve better results than the lower bound in a symmetric edge case, since we can eliminate the term that depends on α_ϵ.

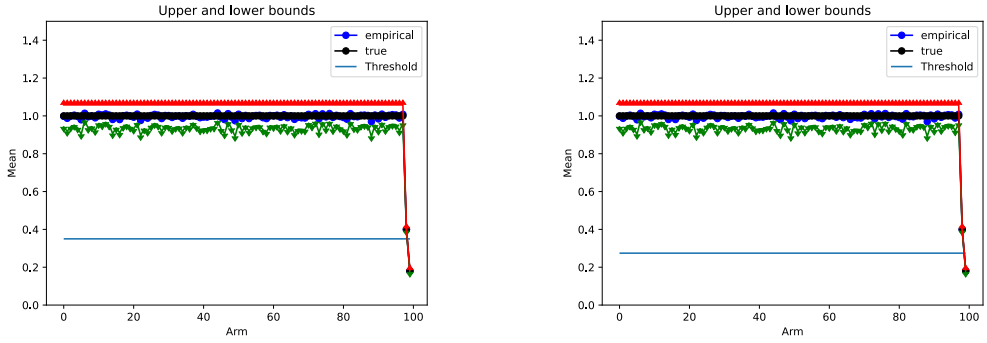


Figure: visualization for the ϵ update where $\alpha_\epsilon \ll \beta_\epsilon$

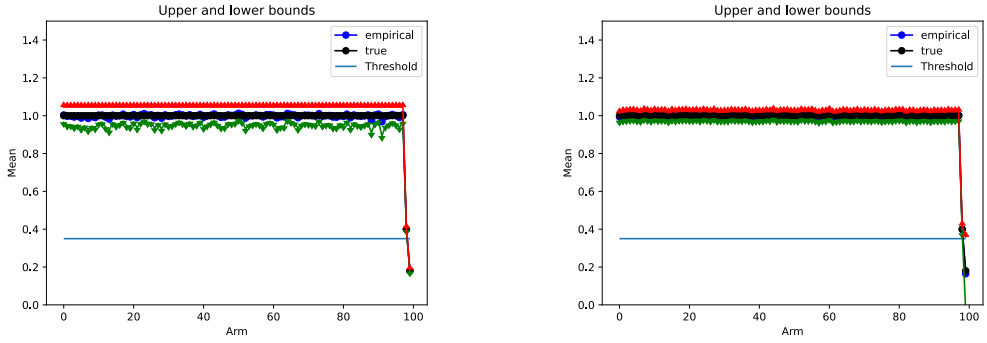
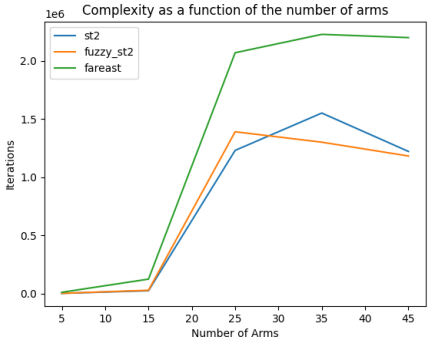
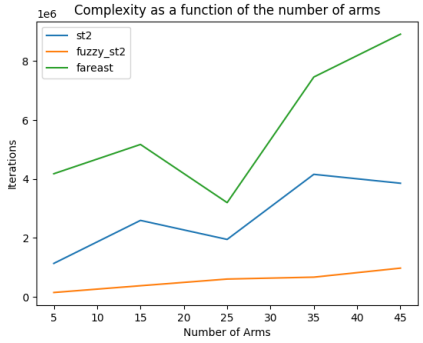


Figure: $\texttt{(ST)}^2$ on the left, FAREAST on the right

In this case FAREAST required 57,262,366 arm pulls, $\texttt{(ST)}^2$ required 6,518,383 pulls, and our algorithm required merely 3,801,085 pulls.



We plotted the graph of number of the iterations until stopping time of each algorithm on the typical case as a function of number of arms and received the plot on the left. As it can be seen from the graph, in the typical case the fuzzy $\texttt{(ST)}^2$ algorithm and $\texttt{(ST)}^2$ algorithm have similar sample complexity, and, as can be expected, the `FAREAST` algorithm has worse sample complexity.



Lastly, we have plotted the graph of the number of the iteration until stopping time of each algorithm on the edge case as a function of number of arms and received the graph on the left. As it can be seen from the plot, Fuzzy $\texttt{(ST)}^2$ sample complexity is much lower than $\texttt{(ST)}^2$ and `FAREAST` sample complexity.

3.3 Conclusions

To conclude, we have proposed an approximation algorithm for finding all ϵ-good arms that has better sample complexity than the algorithms proposed in the paper [1]. Thus, we enable users to control the trade-off between accuracy and sample complexity while ensuring near-optimal run time.

References

[1] Blake Mason et al. "Finding All \epsilon-Good Arms in Stochastic Bandits". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 20707–20718. URL: https://proceedings.neurips.cc/paper/2020/file/edf0320adc8658b25ca26be5351b6c4a-Paper.pdf.

[2] Shivaram Kalyanakrishnan et al. "PAC Subset Selection in Stochastic Multi-Armed Bandits". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. ICML'12. Edinburgh, Scotland: Omnipress, 2012, pp. 227–234. ISBN: 9781450312851.

[3] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. "Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems". In: *Journal of Machine Learning Research* 7.39 (2006), pp. 1079–1105. URL: http://jmlr.org/papers/v7/evendar06a.html.