

# Thresholding Bandits with Augmented UCB

Subhojyoti Mukherjee  
K.P. Naveen  
Nandan Sudarsanam  
Balaraman Ravindran

IIT Madras

August 13, 2017

# Overview

- 1 Problem Definition
- 2 Contribution
- 3 Previous Works
- 4 AugUCB
- 5 Theoretical Analysis
- 6 Experiments
- 7 Conclusion

# Problem Definition of TBP

- **Primary aim:** Identify all the arms whose mean of the reward distribution ( $r_i$ ) is above a particular threshold  $\tau$  given as input.

# Problem Definition of TBP

- **Primary aim:** Identify all the arms whose mean of the reward distribution ( $r_i$ ) is above a particular threshold  $\tau$  given as input.
- **Condition:** This has to be achieved within  $T$  timesteps of exploration and this is termed as a fixed-budget problem.

# Problem Definition of TBP

- We define the set  $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$ .

# Problem Definition of TBP

- We define the set  $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$ .
- $S_\tau^c$  denote the complement of  $S_\tau$ , i.e.,  $S_\tau^c = \{i \in \mathcal{A} : r_i < \tau\}$ .

# Problem Definition of TBP

- We define the set  $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$ .
- $S_\tau^c$  denote the complement of  $S_\tau$ , i.e.,  $S_\tau^c = \{i \in \mathcal{A} : r_i < \tau\}$ .
- Let  $\hat{S}_\tau$  denote the recommendation of a learning algorithm after  $T$  time units of exploration, while  $\hat{S}_\tau^c$  denotes its complement.

# Problem Definition of TBP

- We define the set  $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$ .
- $S_\tau^c$  denote the complement of  $S_\tau$ , i.e.,  $S_\tau^c = \{i \in \mathcal{A} : r_i < \tau\}$ .
- Let  $\hat{S}_\tau$  denote the recommendation of a learning algorithm after  $T$  time units of exploration, while  $\hat{S}_\tau^c$  denotes its complement.
- The goal of the learning agent is to minimize the expected loss:

$$\mathbb{E}[\mathcal{L}(T)] = \mathbb{P}\left( \underbrace{\{S_\tau \cap \hat{S}_\tau^c \neq \emptyset\}}_{\text{Rejected good arms}} \cup \underbrace{\{\hat{S}_\tau \cap S_\tau^c \neq \emptyset\}}_{\text{Accepted bad arms}} \right)$$



# Challenges in the TBP Settings

- Closer the true mean of reward distribution of the arms' to the threshold  $\Rightarrow$  harder the problem.

# Challenges in the TBP Settings

- Closer the true mean of reward distribution of the arms' to the threshold  $\Rightarrow$  harder the problem.
- Lesser the budget  $\Rightarrow$  harder the problem.

# Challenges in the TBP Settings

- Closer the true mean of reward distribution of the arms' to the threshold  $\Rightarrow$  harder the problem.
- Lesser the budget  $\Rightarrow$  harder the problem.
- Higher the variance of reward distribution of the arms'  $\Rightarrow$  harder the problem.

- We propose the Augmented UCB (AugUCB) [Mukherjee et al. (2017)] algorithm for the fixed-budget TBP setting.

# Contributions

- We propose the Augmented UCB (AugUCB) [Mukherjee et al. (2017)] algorithm for the fixed-budget TBP setting.
- AugUCB takes into account the empirical variances of the arms along with mean estimates.

# Contributions

- We propose the Augmented UCB (AugUCB) [Mukherjee et al. (2017)] algorithm for the fixed-budget TBP setting.
- AugUCB takes into account the empirical variances of the arms along with mean estimates.
- It is the first variance-based arm elimination algorithm for the considered TBP settings.

# Contributions

- We propose the Augmented UCB (AugUCB) [Mukherjee et al. (2017)] algorithm for the fixed-budget TBP setting.
- AugUCB takes into account the empirical variances of the arms along with mean estimates.
- It is the first variance-based arm elimination algorithm for the considered TBP settings.
- It addresses an open problem discussed in Auer and Ortner (2010) of designing an algorithm that can eliminate arms based on variance estimates.

- We propose the Augmented UCB (AugUCB) [Mukherjee et al. (2017)] algorithm for the fixed-budget TBP setting.
- AugUCB takes into account the empirical variances of the arms along with mean estimates.
- It is the first variance-based arm elimination algorithm for the considered TBP settings.
- It addresses an open problem discussed in Auer and Ortner (2010) of designing an algorithm that can eliminate arms based on variance estimates.
- We also define a new problem complexity which uses empirical variance estimates along with arm's mean for giving the theoretical bound.



- The Anytime Parameter Free (APT) [Locatelli et al. (2016)] algorithm *was proposed for TBP setting* in ICML 2016.

# APT Approach

- The Anytime Parameter Free (APT) [Locatelli et al. (2016)] algorithm *was proposed for TBP setting* in ICML 2016.
- This algorithm uses only mean estimation to find the  $S_\tau$ .

- The Anytime Parameter Free (APT) [Locatelli et al. (2016)] algorithm *was proposed for TBP setting* in ICML 2016.
- This algorithm uses only mean estimation to find the  $S_\tau$ .
- Theoretically they proved this algorithm to be almost optimal when only mean estimation is used as a metric of comparison.

- The Anytime Parameter Free (APT) [Locatelli et al. (2016)] algorithm *was proposed for TBP setting* in ICML 2016.
- This algorithm uses only mean estimation to find the  $S_\tau$ .
- Theoretically they proved this algorithm to be almost optimal when only mean estimation is used as a metric of comparison.
- Empirically it outperformed other state-of-the-art algorithms which were modified to perform in the TBP setting.

---

## Algorithm 1 APT

---

**Input:** Time horizon  $T$ , threshold  $\tau$ , tolerance factor  $\epsilon \geq 0$

Pull each arm once

**for**  $t = K + 1, \dots, T$  **do**

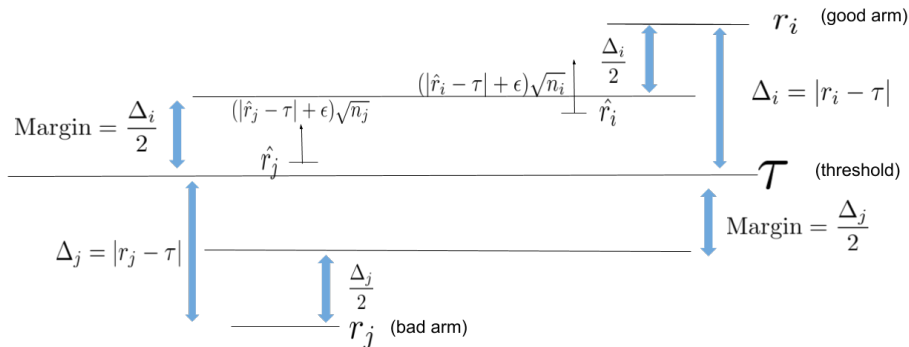
    Pull arm  $j \in \arg \min_{i \in A} \left\{ (|\hat{r}_i - \tau| + \epsilon) \sqrt{n_i} \right\}$  and observe the reward for arm  $j$ .

**end for**

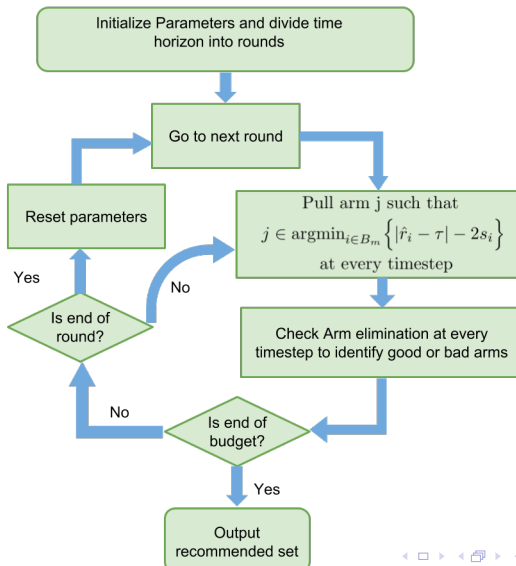
**Output:**  $\hat{S}_\tau = \{i : \hat{r}_i \geq \tau\}$ .

---

# Intuition of APT

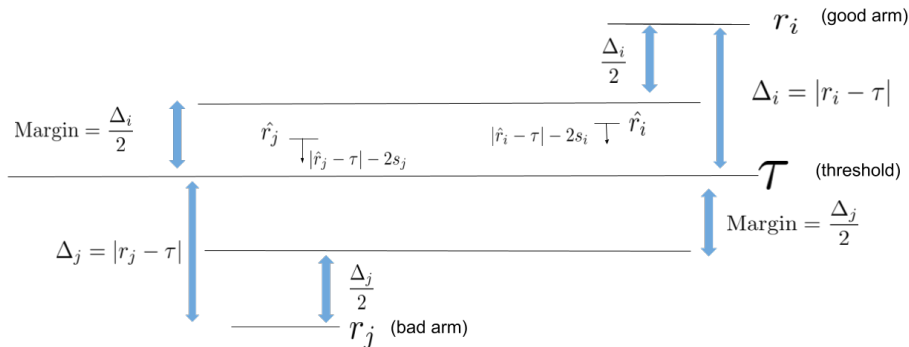


# AugUCB algorithm



# AugUCB algorithm (Intuition, Arm pulling)

- Like UCB-Imp, AugUCB also divides the time budget  $T$  into rounds.
- At every timestep we pull arm  $j$  s.t.  $j \in \arg \min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$  (like APT).

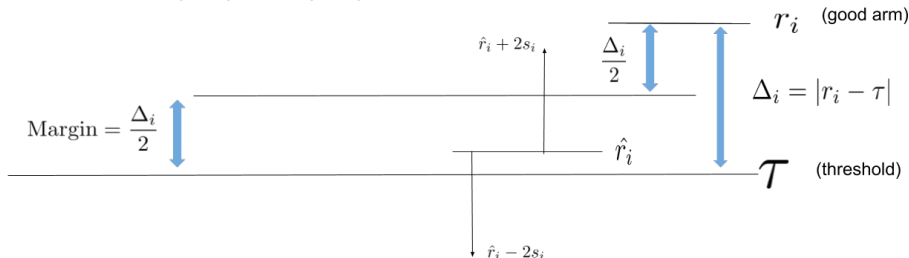




# AugUCB algorithm (Intuition, Arm Elimination)

- We eliminate an arm when we are sure that  $\hat{r}_i$  is close to  $r_i$  with high probability and hence identify it as good or bad arm.
- It's risky to eliminate an arm when  $\hat{r}_i$  is inside *Margin*.
- Confidence interval  $s_i$  will make sure arm  $i$  is not eliminated while inside Margin with a high probability.

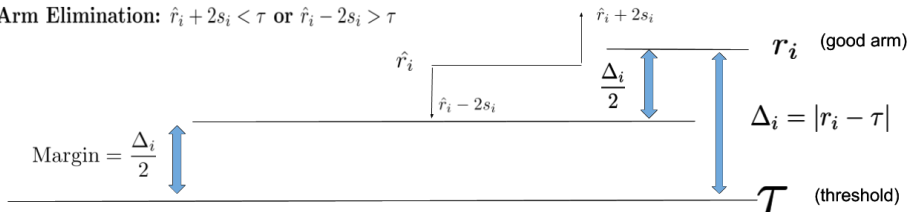
Arm Elimination:  $\hat{r}_i + 2s_i < \tau$  or  $\hat{r}_i - 2s_i > \tau$



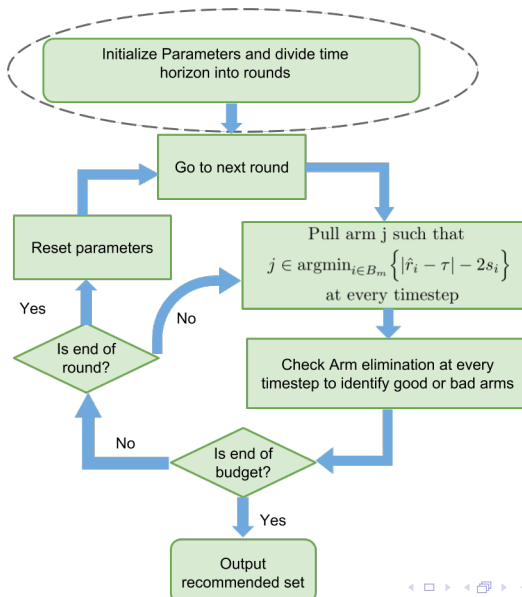
# AugUCB algorithm (Intuition, Arm Elimination)

- Now we see that  $\hat{r}_i$  has moved close to its true estimate  $r_i$ .
- We eliminate  $i$  and re-allocate the remaining budget to pull arms close to the threshold

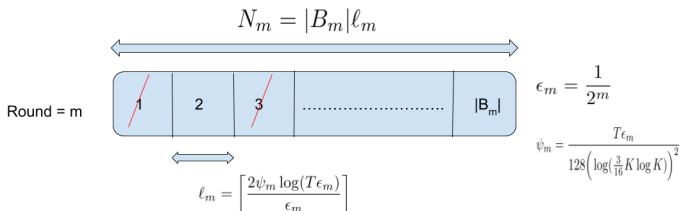
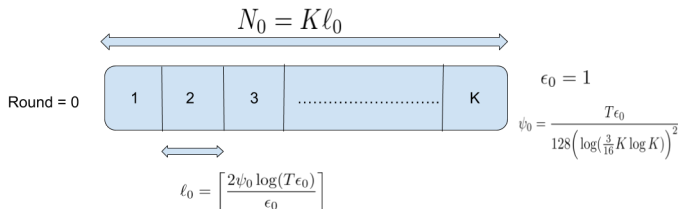
Arm Elimination:  $\hat{r}_i + 2s_i < \tau$  or  $\hat{r}_i - 2s_i > \tau$



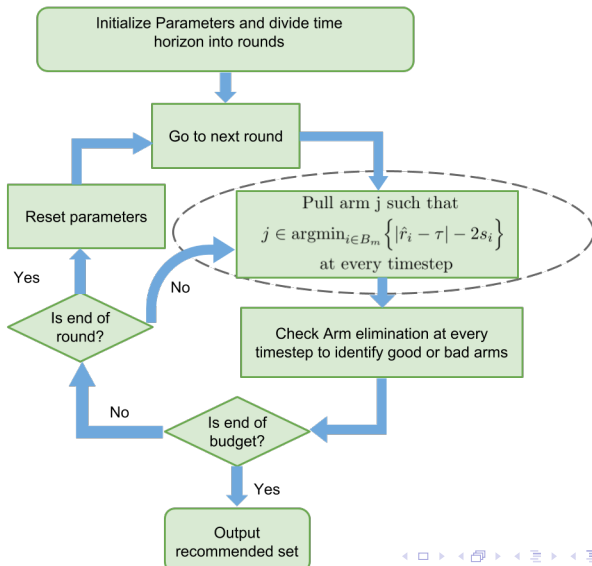
# AugUCB parameter initialization



# Parameter initialization



# AugUCB arm pull



- We pull the arm that minimizes  $j \in \arg \min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$

- We pull the arm that minimizes  $j \in \arg \min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$
- We define the confidence interval  $s_i = \sqrt{\frac{\rho \psi_m(\hat{v}_i + 1) \log(T \epsilon_m)}{4n_i}}$ .

# Arm pull

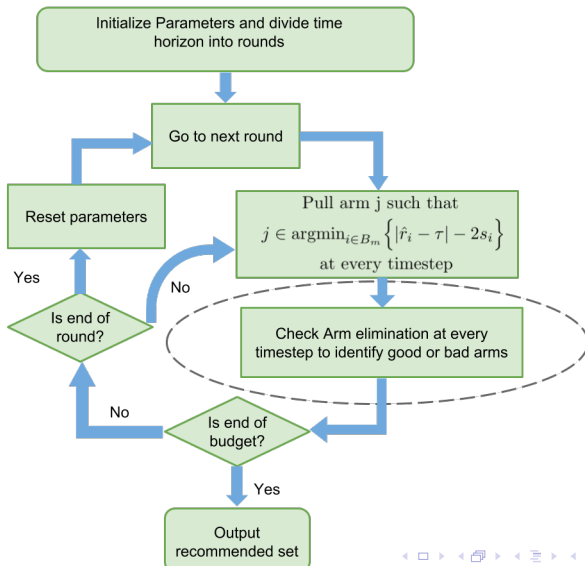
- We pull the arm that minimizes  $j \in \arg \min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$
- We define the confidence interval  $s_i = \sqrt{\frac{\rho \psi_m(\hat{v}_i + 1) \log(T \epsilon_m)}{4n_i}}$ .
- $s_i$  decreases with more  $n_i$  and  $\psi_m$  and  $\rho$  ensures that it decreases at a correct rate.



# Arm pull

- We pull the arm that minimizes  $j \in \arg \min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$
- We define the confidence interval  $s_i = \sqrt{\frac{\rho \psi_m(\hat{v}_i + 1) \log(T \epsilon_m)}{4n_i}}$ .
- $s_i$  decreases with more  $n_i$  and  $\psi_m$  and  $\rho$  ensures that it decreases at a correct rate.
- Note that  $\hat{v}_i$  estimated variance in  $s_i$  makes the algorithm pull the arm which shows more variance.

# AugUCB arm elimination



# Arm elimination

- Arm elimination condition is checked at every timestep.

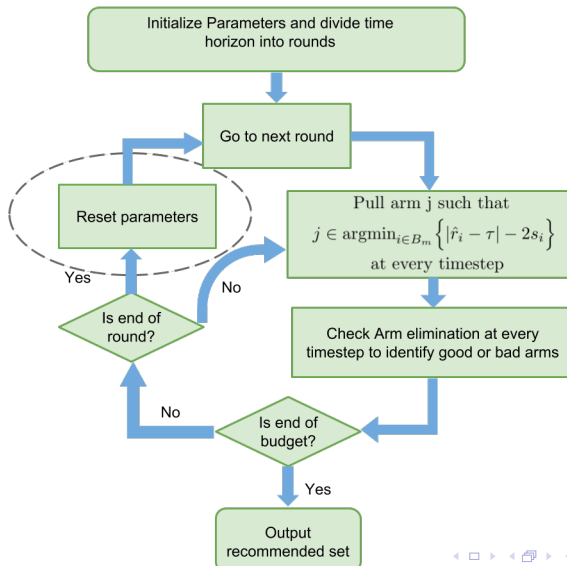
# Arm elimination

- Arm elimination condition is checked at every timestep.
- It identifies the arm whose estimates lies close to their true mean and thus help in identifying the good or bad arms.

# Arm elimination

- Arm elimination condition is checked at every timestep.
- It identifies the arm whose estimates lies close to their true mean and thus help in identifying the good or bad arms.
- It eliminates the arms which have been identified as good or bad arms (with a high probability) and re-allocates the remaining budget for surviving arms.

# AugUCB parameter reset



- Increase the allocated pulls  $\ell_m$  for each surviving arms.

# AugUCB parameter reset

- Increase the allocated pulls  $\ell_m$  for each surviving arms.
- Proportionally reduce the exploration factor  $\psi_m$  for next round.



# AugUCB parameter reset

- Increase the allocated pulls  $\ell_m$  for each surviving arms.
- Proportionally reduce the exploration factor  $\psi_m$  for next round.
- Recalculate the length of next round on the number of surviving arms.

# Problem Complexity

- We have defined  $\Delta_i = |r_i - \tau|$ .

# Problem Complexity

- We have defined  $\Delta_i = |r_i - \tau|$ .
- We define  $H_1 = \sum_{i=1}^K \frac{1}{\Delta_i^2}$  and  $H_2 = \min_{i \in \mathcal{A}} \frac{i}{\Delta_{(i)}^2}$  where  $\Delta_{(i)}$  is an increasing ordering of  $\Delta_i$ .

# Problem Complexity

- We have defined  $\Delta_i = |r_i - \tau|$ .
- We define  $H_1 = \sum_{i=1}^K \frac{1}{\Delta_i^2}$  and  $H_2 = \min_{i \in \mathcal{A}} \frac{i}{\Delta_{(i)}^2}$  where  $\Delta_{(i)}$  is an increasing ordering of  $\Delta_i$ .
- From Audibert and Bubeck (2010) the relationship between  $H_1$  and  $H_2$  can be derived as,

$$H_2 \leq H_1 \leq \log(2K)H_2$$

# Problem Complexity

- For a variance aware algorithm we define  $H_{\sigma,1}$  ( as in Gabillon et al. (2011)) that incorporates reward variances into its expression as:

$$H_{\sigma,1} = \sum_{i=1}^K \frac{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}{\Delta_i^2}.$$

# Problem Complexity

- For a variance aware algorithm we define  $H_{\sigma,1}$  ( as in Gabillon et al. (2011)) that incorporates reward variances into its expression as:

$$H_{\sigma,1} = \sum_{i=1}^K \frac{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}{\Delta_i^2}.$$

- Finally, analogous to  $H_2$ , we introduce  $H_{\sigma,2}$ , such that  $H_{\sigma,2} = \max_{i \in \mathcal{A}} \frac{i}{\tilde{\Delta}_{(i)}^2}$ , where  $\tilde{\Delta}_i^2 = \frac{\Delta_i^2}{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}$ ,  $(\tilde{\Delta}_{(i)})$  is an increasing ordering of  $(\tilde{\Delta}_i)$ .

# Problem Complexity

- For a variance aware algorithm we define  $H_{\sigma,1}$  ( as in Gabillon et al. (2011)) that incorporates reward variances into its expression as:

$$H_{\sigma,1} = \sum_{i=1}^K \frac{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}{\Delta_i^2}.$$

- Finally, analogous to  $H_2$ , we introduce  $H_{\sigma,2}$ , such that  $H_{\sigma,2} = \max_{i \in \mathcal{A}} \frac{i}{\tilde{\Delta}_{(i)}^2}$ , where  $\tilde{\Delta}_i^2 = \frac{\Delta_i^2}{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}$ ,  $(\tilde{\Delta}_{(i)})$  is an increasing ordering of  $(\tilde{\Delta}_i)$ .
- From Audibert and Bubeck (2010), we can show that

$$H_{\sigma,2} \leq H_{\sigma,1} \leq \log(2K)H_{\sigma,2}.$$

# Problem Complexity

- For a variance aware algorithm we define  $H_{\sigma,1}$  ( as in Gabillon et al. (2011)) that incorporates reward variances into its expression as:

$$H_{\sigma,1} = \sum_{i=1}^K \frac{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}{\Delta_i^2}.$$

- Finally, analogous to  $H_2$ , we introduce  $H_{\sigma,2}$ , such that  $H_{\sigma,2} = \max_{i \in \mathcal{A}} \frac{i}{\tilde{\Delta}_{(i)}^2}$ , where  $\tilde{\Delta}_i^2 = \frac{\Delta_i^2}{\sigma_i + \sqrt{\sigma_i^2 + (16/3)\Delta_i}}$ ,  $(\tilde{\Delta}_{(i)})$  is an increasing ordering of  $(\tilde{\Delta}_i)$ .
- From Audibert and Bubeck (2010), we can show that

$$H_{\sigma,2} \leq H_{\sigma,1} \leq \log(2K)H_{\sigma,2}.$$

- Note that  $H_1$ ,  $H_2$  and  $H_{\sigma,1}$ ,  $H_{\sigma,2}$  are not directly comparable to each other except in a special case when variances and gaps  $(\Delta_i)$  are very low we can say that  $H_{\sigma,1} < H_1$ .



# Expected Loss of AugUCB

## Theorem

For  $K \geq 4$  and  $\rho = 1/3$ , the expected loss of the AugUCB algorithm is given by,

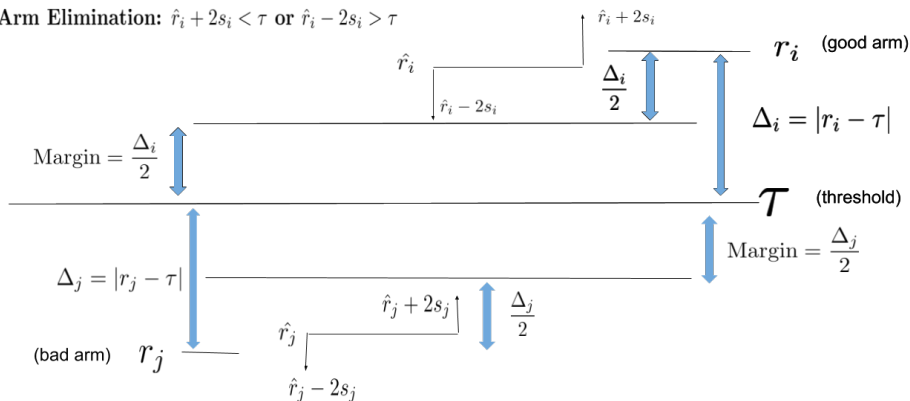
$$\mathbb{E}[\mathcal{L}(T)] \leq 2KT \exp \left( - \frac{T}{4096 \log(K \log K) H_{\sigma,2}} \right).$$

Table: AugUCB vs. State of the art

Algorithm	Upper Bound on Expected Loss	Oracle
AugUCB	$\exp \left( - \frac{T}{4096 \log(K \log K) H_{\sigma,2}} + \log(2KT) \right)$	No
UCBEV	$\exp \left( - \frac{1}{512} \frac{T-2K}{H_{\sigma,1}} + \log(6KT) \right)$	Yes
APT	$\exp \left( - \frac{T}{64H_1} + 2 \log((\log(T) + 1)K) \right)$	No
UCBE	$\exp \left( - \frac{T-K}{18H_1} - 2 \log(\log(T)K) \right)$	Yes

# Sketch of the proof

Arm Elimination:  $\hat{r}_i + 2s_i < \tau$  or  $\hat{r}_i - 2s_i > \tau$



# Finally, experiment!!!

- We compare with APT, AugUCB, UCBE, UCBEV, UA.

# Finally, experiment!!!

- We compare with APT, AugUCB, UCBE, UCBEV, UA.
- Note that UCBE and UCBEV require access to  $H_1$  and  $H_{\sigma,1}$  as input and hence not implementable in real life.
- By access we mean that an oracle supplies them the  $H_1$  or  $H_{\sigma,1}$ . They do not have access to individual means and variances.

# Finally, experiment!!!

- We compare with APT, AugUCB, UCBE, UCBEV, UA.
- Note that UCBE and UCBEV require access to  $H_1$  and  $H_{\sigma,1}$  as input and hence not implementable in real life.
- By access we mean that an oracle supplies them the  $H_1$  or  $H_{\sigma,1}$ . They do not have access to individual means and variances.
- APT, AugUCB, UA do not require access to  $H_1$  or  $H_{\sigma,1}$ .

# Experimental Setup

- This setup involves Gaussian reward distributions with  $K = 100$ ,  $T = 10000$  and  $\tau = 0.5$  with the reward means set in two groups.

# Experimental Setup

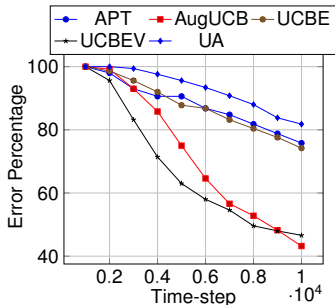
- This setup involves Gaussian reward distributions with  $K = 100$ ,  $T = 10000$  and  $\tau = 0.5$  with the reward means set in two groups.
- The first 10 arms partitioned into two groups; the respective means are  $r_{1:5} = 0.45$ ,  $r_{6:10} = 0.55$ .

# Experimental Setup

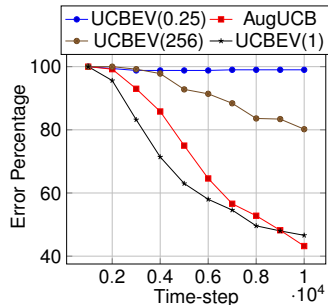
- This setup involves Gaussian reward distributions with  $K = 100$ ,  $T = 10000$  and  $\tau = 0.5$  with the reward means set in two groups.
- The first 10 arms partitioned into two groups; the respective means are  $r_{1:5} = 0.45$ ,  $r_{6:10} = 0.55$ .
- The means of arms  $i = 11 : 100$  are chosen same as  $r_{11:100} = 0.4$ .
- Variances are set as  $\sigma_{1:5}^2 = 0.3$  and  $\sigma_{6:10}^2 = 0.8$ ;  $\sigma_{11:100}^2$  are independently and uniformly chosen in the interval  $[0.2, 0.3]$ .



# Experimental Results

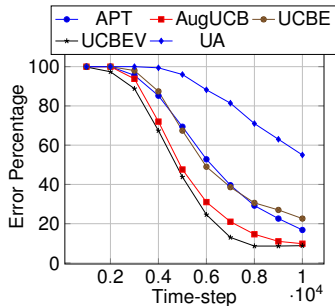


(a) Expt-1: Two Group Setting (Advance)

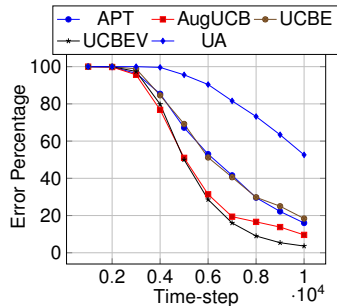


(b) Expt-2: Two Group Setting (Advance)

# Experimental Results



(c) Expt-1: Arithmetic Progression (Gaussian)



(d) Expt-2: Geometric Progression (Gaussian)

# Conclusion

- We proposed the AugUCB algorithm for the fixed budget TBP which uses variance estimation and arm elimination to give an improved theoretical and experimental guarantees than APT.

# Conclusion

- We proposed the AugUCB algorithm for the fixed budget TBP which uses variance estimation and arm elimination to give an improved theoretical and experimental guarantees than APT.
- Further studies are required to establish a lower bound on the expected loss of AugUCB.

# Conclusion

- We proposed the AugUCB algorithm for the fixed budget TBP which uses variance estimation and arm elimination to give an improved theoretical and experimental guarantees than APT.
- Further studies are required to establish a lower bound on the expected loss of AugUCB.
- A more detailed analysis of the non-uniform arm selection and parameter selection is also required.

# Thank You