Thresholding Bandits with Augmented UCB

Subhojyoti Mukherjee
CS15S300
Guide: Dr. Balaraman Ravindran
Co-Guide: Dr. Nandan Sudarsanam

IIT Madras

July 15, 2017

Overview

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.
- A finite set of actions or arms belonging to set A such that $|A| = K$.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.
- A finite set of actions or arms belonging to set A such that $|A| = K$.
- The rewards for each of the arms are identical and independent random variables drawn from distribution specific to the arm.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.
- A finite set of actions or arms belonging to set A such that $|A| = K$.
- The rewards for each of the arms are identical and independent random variables drawn from distribution specific to the arm.
- The learner does not know the mean $r_i, \forall i \in A$ of the distribution or the variance σ_i^2.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The distributions for each of the arms are fixed throughout the time horizon denoted by T.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The distributions for each of the arms are fixed throughout the time horizon denoted by T.
- The estimated reward $\hat{r}_i = \frac{1}{n_i}\sum_{z=1}^{n_i} X_{i,z}$.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The distributions for each of the arms are fixed throughout the time horizon denoted by T.
- The estimated reward $\hat{r}_i = \frac{1}{n_i}\sum_{z=1}^{n_i} X_{i,z}$.
- The more we pull arm i the closer $\hat{r}_i$ gets to r_i.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The distributions for each of the arms are fixed throughout the time horizon denoted by T.
- The estimated reward $\hat{r}_i = \frac{1}{n_i}\sum_{z=1}^{n_i} X_{i,z}$.
- The more we pull arm i the closer $\hat{r}_i$ gets to r_i.
- Due to the uncertainty in $\hat{r}_i$ we have carefully conduct exploration.

Problem Definition of TBP

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

Problem Definition of TBP

- Primary aim: Identify the arms whose mean of the reward distribution (r_i) is above a particular threshold τ 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

- Primary aim: Identify the arms whose mean of the reward distribution (r_i) is above a particular threshold τ given as input.
- Condition: This has to be achieved within T timesteps of exploration and this is termed as a fixed-budget problem.
- At the end of the given T timesteps the learner must recommend a set of arms which (according to it) are the arms having reward mean above τ.

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_τ^c denote the complement of S_τ, 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_τ^c denote the complement of S_τ, 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_τ^c denote the complement of S_τ, 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:

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

Challenges in the TBP Settings

- Closer arms' mean to the threshold $\Rightarrow$ harder is to discriminate.

Challenges in the TBP Settings

- Closer arms' mean to the threshold $\Rightarrow$ harder is to discriminate.
- Lesser the budget $\Rightarrow$ harder the problem.

Challenges in the TBP Settings

- Closer arms' mean to the threshold $\Rightarrow$ harder is to discriminate.
- Lesser the budget $\Rightarrow$ harder the problem.
- Higher variance of the arms' $\Rightarrow$ harder the problem.

Applications

- Selecting the best channels (out of several existing channels) for mobile communications in a very short duration whose qualities are above an acceptable threshold (see Audibert and Bubeck (2010)).

Applications

- Selecting the best channels (out of several existing channels) for mobile communications in a very short duration whose qualities are above an acceptable threshold (see Audibert and Bubeck (2010)).
- Selecting a small set of best workers (out of a very large pool of workers) whose productivity is above a threshold.

Applications

- Selecting the best channels (out of several existing channels) for mobile communications in a very short duration whose qualities are above an acceptable threshold (see Audibert and Bubeck (2010)).
- Selecting a small set of best workers (out of a very large pool of workers) whose productivity is above a threshold.
- In anomaly detection and classification (see Locatelli et al. (2016)).

Contributions

- 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.

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 also define a new problem complexity which uses empirical variance estimates along with arm's mean for giving the theoretical bound.

The Upper Confidence Bound (UCB) Approach

- Since there is an initial uncertainty in the estimated mean ($\hat{r}_i$) introduce a confidence interval term c_i.

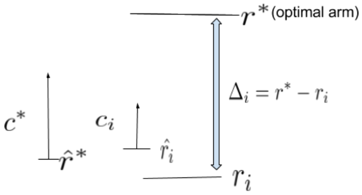
The Upper Confidence Bound (UCB) Approach

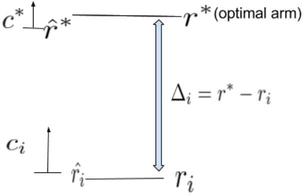
- Since there is an initial uncertainty in the estimated mean ($\hat{r}_i$) introduce a confidence interval term c_i.
- c_i represents the uncertainty about $\hat{r}_i$.
- c_i ensures that the arm i is properly explored and is gradually reduced with time as one pulls the arm i more.

The Upper Confidence Bound (UCB) Approach

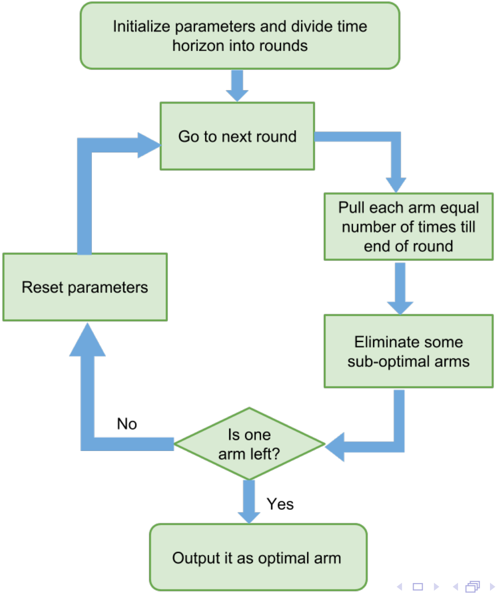
- Since there is an initial uncertainty in the estimated mean ($\hat{r}_i$) introduce a confidence interval term c_i.
- c_i represents the uncertainty about $\hat{r}_i$.
- c_i ensures that the arm i is properly explored and is gradually reduced with time as one pulls the arm i more.
- At every timestep pull arm $j \in \arg\max_{i \in A}\{\hat{r}_i + c_i\}$ and this will ensure that proper exploration is done.

The UCB Approach



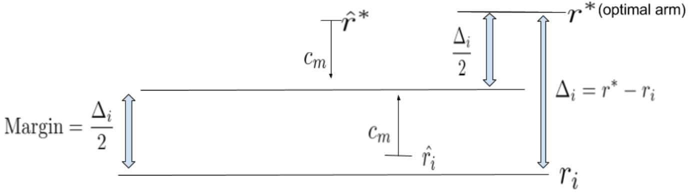


Approach of UCB-Improved (UCB-Imp)



Intuition of UCB-Improved (UCB-Imp)

Arm Elimination: $\hat{r}_i + c_m < \hat{r}_{max} - c_m$



APT Approach

- 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_τ.

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_τ.
- Theoretically they proved this algorithm to be almost optimal when only mean estimation is used as a metric of comparison.

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_τ.
- 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.

APT Algorithm

Algorithm 1 APT

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

Pull each arm once

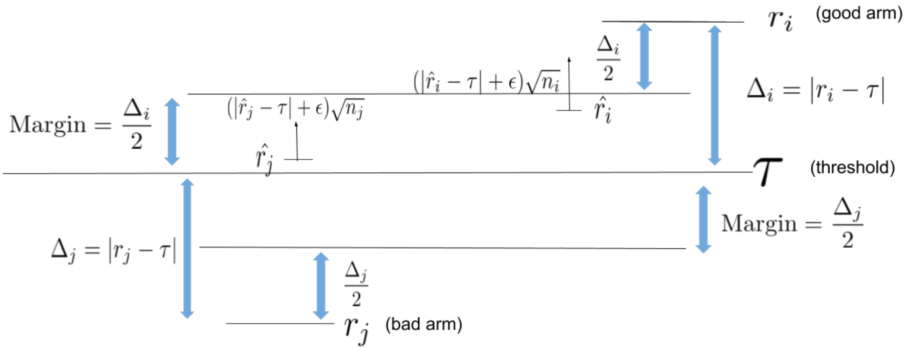
for $t = K + 1, .., T$ **do**

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.

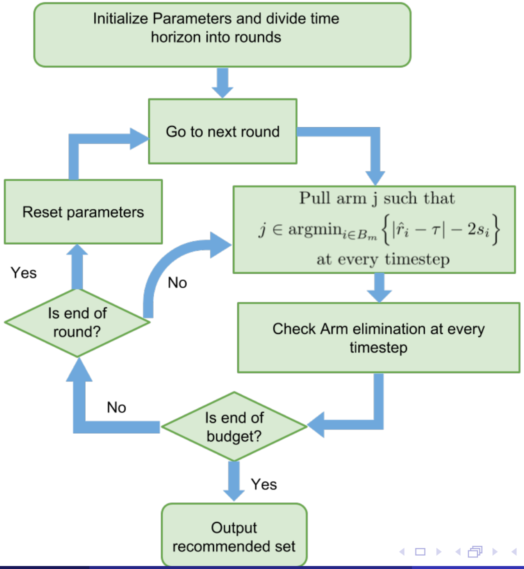
end for

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

Intuition of APT

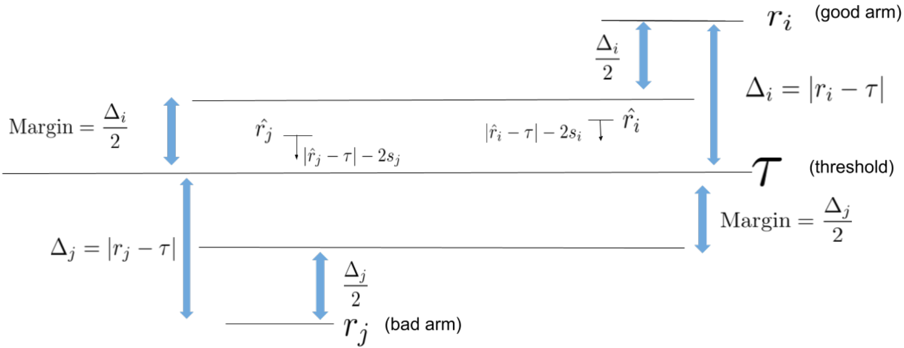


AugUCB algorithm



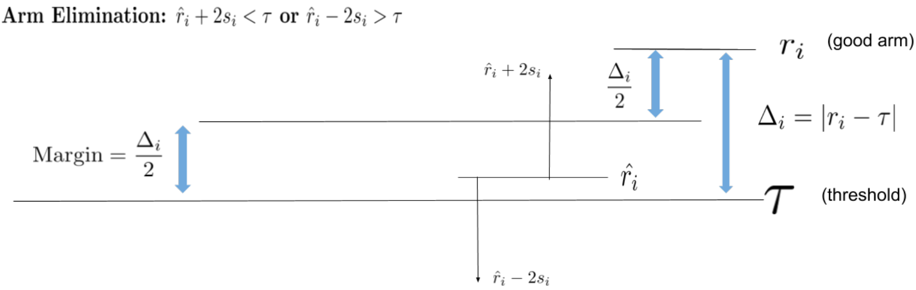
AugUCB algorithm (Intuition, Arm pulling)

- 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).



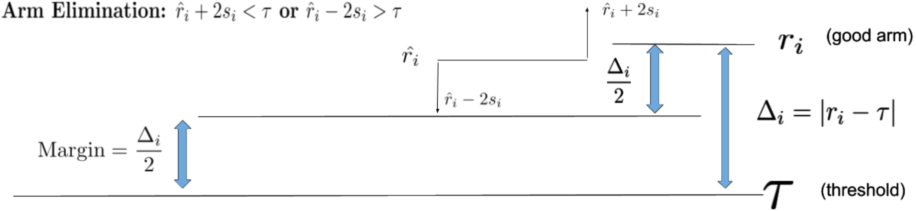
AugUCB algorithm (Intuition, Arm Elimination)

- It is risky to eliminate arm i while $\hat{r}_i$ is inside *Margin*.
- Confidence interval s_i will make sure arm i is not eliminated while inside Margin with a high probability.

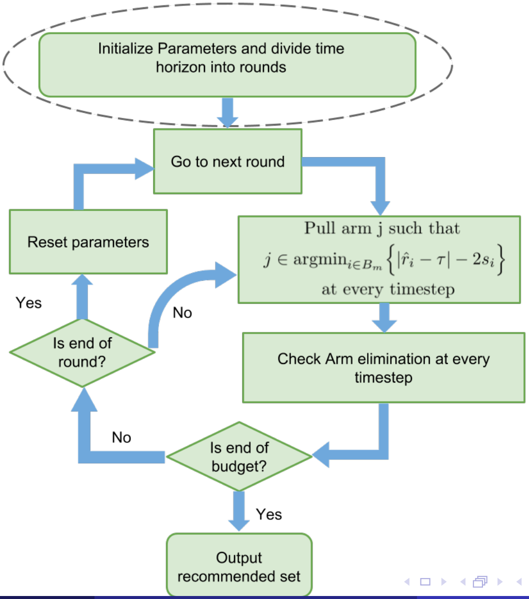


AugUCB algorithm (Intuition, Arm Elimination)

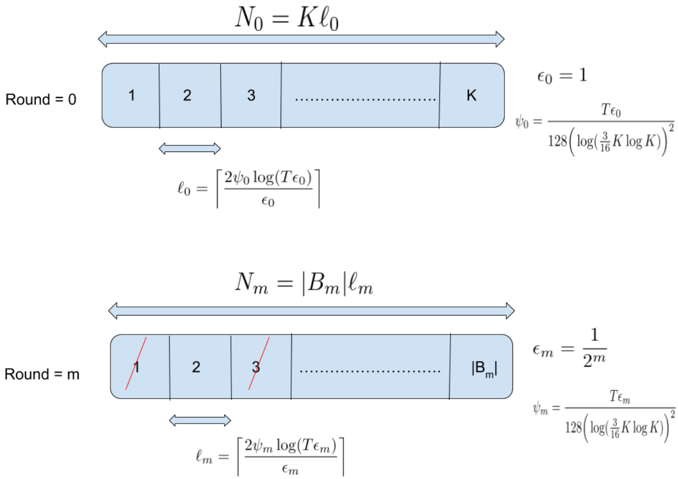
Arm Elimination: $\hat{r}_i + 2s_i < \tau$ **or** $\hat{r}_i - 2s_i > \tau$



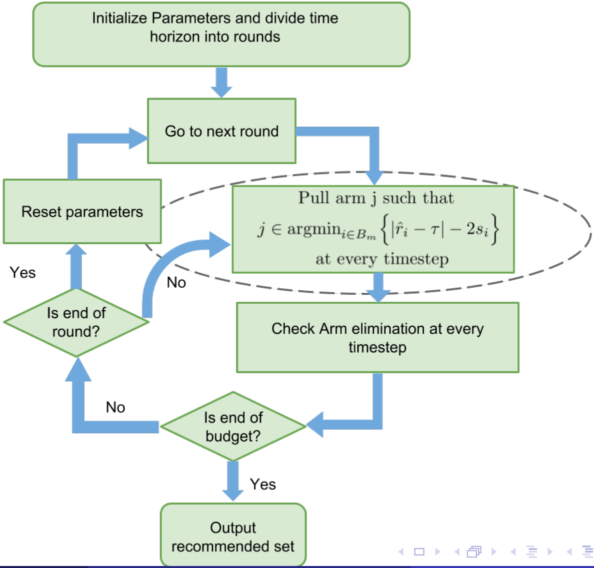
AugUCB parameter initialization



Parameter initialization



AugUCB arm pull



Arm pull

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

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 $s_i = \sqrt{\frac{\rho \psi_m (\hat{v}_i + 1) \log(T \epsilon_m)}{4 n_i}}$.

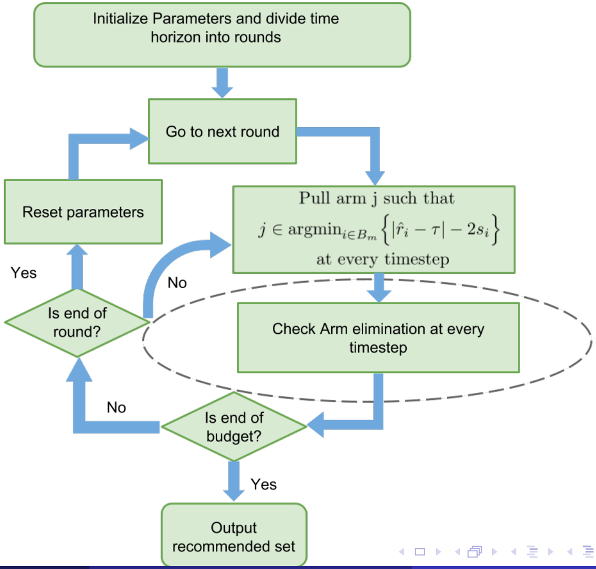
Arm pull

- 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 ψ_m and ρ 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 ψ_m and ρ 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.

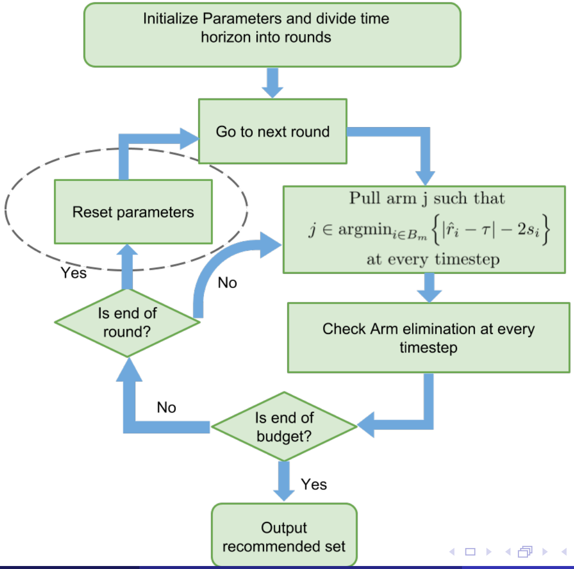
Arm elimination

- Arm elimination condition is checked at every timestep.
- It eliminated arm which are above or below margin.

Arm elimination

- Arm elimination condition is checked at every timestep.
- It eliminated arm which are above or below margin.
- Re-allocates the remaining budget for surviving arms.

AugUCB parameter reset



AugUCB parameter reset

- Reduce the exploration factor ψ_m for next round.

AugUCB parameter reset

- Reduce the exploration factor ψ_m for next round.
- Increase the allocated pulls ℓ_m for each surviving arms.

AugUCB parameter reset

- Reduce the exploration factor ψ_m for next round.
- Increase the allocated pulls ℓ_m for each surviving arms.
- Recalculate the length of next round on the number of surviving arms.

•

Problem Complexity

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

Problem Complexity

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

Problem Complexity

- We have defined $\Delta_i = |\tau - r_i|$.
- 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 Δ_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 (Δ_i) are very low we can say that $H_{\sigma,1} < H_1$.

Expected Loss of AugUCB

Theorem

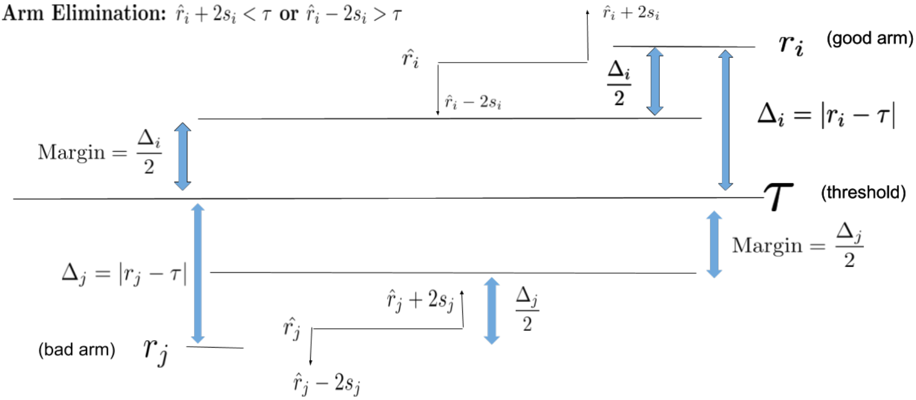
For $K \geq 4$ and $\rho = 1/3$, the expected loss of the AugUCB algorithm is given by,

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

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



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 $H_{\sigma,1}$.

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 $H_{\sigma,1}$.
- UCBE, UCBEV, CSAR and UA come from the pure exploration lineage and are modified suitably to perform in TBP setting.

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.

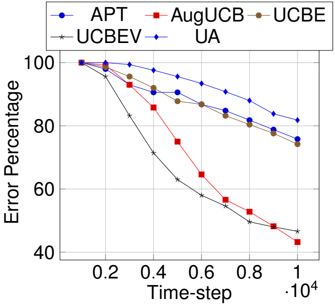
Experimental Setup

- 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 $r_{1:5} = 0.45$, $r_{6:10} = 0.55$.

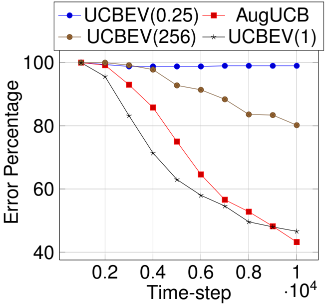
Experimental Setup

- 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 $\sigma^2_{1:5} = 0.3$ and $\sigma^2_{6:10} = 0.8$; $\sigma^2_{11:100}$ 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)

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.
- This work has been accepted in the proceedings of IJCAI 2017.

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.
- This work has been accepted in the proceedings of IJCAI 2017.
- 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.
- This work has been accepted in the proceedings of IJCAI 2017.
- 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