Thresholding Bandits with Augmented UCB

Subhojyoti Mukherjee

IIT Madras

July 10, 2017

Overview

Introduction

- The bandit problem is a sequential decision making process where at each timestep we have to choose one action or arm from a set of arms.

Introduction

- The bandit problem is a sequential decision making process where at each timestep we have to choose one action or arm from a set of arms.
- There is a specific reward distribution attached to each arm. After pulling an arm we receive a reward from the reward distribution specific to the arm.

Introduction

- The bandit problem is a sequential decision making process where at each timestep we have to choose one action or arm from a set of arms.
- There is a specific reward distribution attached to each arm. After pulling an arm we receive a reward from the reward distribution specific to the arm.
- After say pulling each arm once, we are presented with an *exploration-exploitation* trade-off, that is whether to continue to pull the arm for which we have observed the highest estimated reward till now(exploitation) or to explore a new arm(exploration).

Introduction

- The bandit problem is a sequential decision making process where at each timestep we have to choose one action or arm from a set of arms.
- There is a specific reward distribution attached to each arm. After pulling an arm we receive a reward from the reward distribution specific to the arm.
- After say pulling each arm once, we are presented with an *exploration-exploitation* trade-off, that is whether to continue to pull the arm for which we have observed the highest estimated reward till now(exploitation) or to explore a new arm(exploration).
- If we become too greedy and always exploit, we may miss the chance of actually finding the optimal arm and get stuck with a sub-optimal arm.

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.
- In SMAB problem, we are presented with a finite set of actions or arms.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.
- In SMAB problem, we are presented with a finite set of actions or arms.
- 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.
- In SMAB problem, we are presented with a finite set of actions or arms.
- 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 of the distributions, denoted by $r_i, \forall i \in A$ and the variance is denoted by σ_i^2.

Stochastic Multi-Armed Bandit Problem (SMAB)

- The thresholding bandit problem falls under the broad area of stochastic multi-armed bandit problem.
- In SMAB problem, we are presented with a finite set of actions or arms.
- 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 of the distributions, denoted by $r_i, \forall i \in A$ and the variance is denoted by σ_i^2.
- The distributions for each of the arms are fixed throughout the time horizon denoted by T.

Problem Definition

- The primary aim in the thresholding bandit problem (TBP) is to identify the arms whose mean of the reward distribution is above a particular threshold τ given as input.

Problem Definition

- The primary aim in the thresholding bandit problem (TBP) is to identify the arms whose mean of the reward distribution is above a particular threshold τ given as input.
- The above goal has to be achieved within T timesteps of exploration/exploitation and this is termed as a fixed-budget problem.

Problem Definition

- The primary aim in the thresholding bandit problem (TBP) is to identify the arms whose mean of the reward distribution is above a particular threshold τ given as input.
- The above goal has to be achieved within T timesteps of exploration/exploitation 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

- We define the set $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$. Note that, S_τ is the set of all arms whose reward mean is greater than τ. Let S_τ^c denote the complement of S_τ, i.e., $S_\tau^c = \{i \in \mathcal{A} : r_i < \tau\}$.

Problem Definition

- We define the set $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$. Note that, S_τ is the set of all arms whose reward mean is greater than τ. Let 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

- We define the set $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$. Note that, S_τ is the set of all arms whose reward mean is greater than τ. Let 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 performance of the learning agent is measured by the accuracy with which it can classify the arms into S_τ and S_τ^c after time horizon T. Equivalently, the *loss* $\mathcal{L}(T)$ is defined as

$$\mathcal{L}(T) = \mathbb{I}\big(\{S_\tau \cap \hat{S}_\tau^c \neq \emptyset\} \cup \{\hat{S}_\tau \cap S_\tau^c \neq \emptyset\}\big).$$

Problem Definition

- We define the set $S_\tau = \{i \in \mathcal{A} : r_i \geq \tau\}$. Note that, S_τ is the set of all arms whose reward mean is greater than τ. Let 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 performance of the learning agent is measured by the accuracy with which it can classify the arms into S_τ and S_τ^c after time horizon T. Equivalently, the *loss* $\mathcal{L}(T)$ is defined as

$$\mathcal{L}(T) = \mathbb{I}\big(\{S_\tau \cap \hat{S}_\tau^c \neq \emptyset\} \cup \{\hat{S}_\tau \cap S_\tau^c \neq \emptyset\}\big).$$

- The goal of the learning agent is to minimize the expected loss:

$$\mathbb{E}[\mathcal{L}(T)] = \mathbb{P}\big(\{S_\tau \cap \hat{S}_\tau^c \neq \emptyset\} \cup \{\hat{S}_\tau \cap S_\tau^c \neq \emptyset\}\big).$$

Some practical 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)]).

Some practical 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.

Some practical 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)Locatelli, Gutzeit, and Carpentier]).

Contribution

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

Contribution

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

Contribution

- We propose the Augmented UCB (AugUCB) [Mukherjee et al.(2017)Mukherjee, Naveen, Nandan, and Ravindran] 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.

Contribution

- We propose the Augmented UCB (AugUCB) [Mukherjee et al.(2017)Mukherjee, Naveen, Nandan, and Ravindran] 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 address an open problem discussed in [Auer and Ortner(2010)] of designing an algorithm that can eliminate arms based on variance estimates.

Contribution

- We propose the Augmented UCB (AugUCB) [Mukherjee et al.(2017)Mukherjee, Naveen, Nandan, and Ravindran] 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 address an open problem discussed in [Auer and Ortner(2010)] of designing an algorithm that can eliminate arms based on variance estimates.
- It is the first algorithm on the larger pure exploration setting which uses empirical variance estimates along with arm elimination with a new problem complexity.

Previous Works

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

Previous Works

- The Anytime Parameter Free (APT) [Locatelli et al.(2016)Locatelli, Gutzeit, and Carpentier] algorithm was proposed for TBP setting in ICML 2016.
- This algorithm uses only mean estimation to find the S_τ.

Previous Works

- The Anytime Parameter Free (APT) [Locatelli et al.(2016)Locatelli, Gutzeit, and Carpentier] 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.

Previous Works

- The Anytime Parameter Free (APT) [Locatelli et al.(2016)Locatelli, Gutzeit, and Carpentier] 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.

Previous Works (Pure Exploration)

- The TBP problem also falls within the larger area called the Pure Exploration problem.

Previous Works (Pure Exploration)

- The TBP problem also falls within the larger area called the Pure Exploration problem.
- In pure exploration problems the learner has to output a set of recommendations either with high confidence (fixed confidence) or after a specified number of rounds (fixed budget).

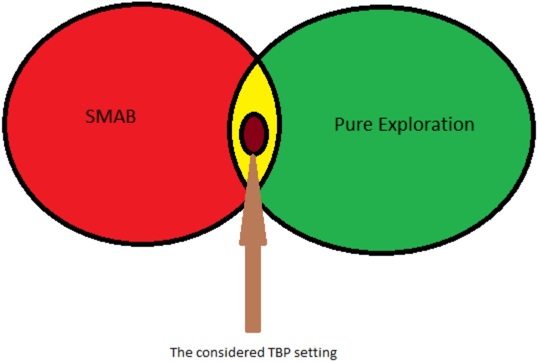
Previous Works (Pure Exploration)

- The TBP problem also falls within the larger area called the Pure Exploration problem.
- In pure exploration problems the learner has to output a set of recommendations either with high confidence (fixed confidence) or after a specified number of rounds (fixed budget).
- Our considered TBP is a fixed budget pure exploration problem.

Previous Works (Pure Exploration)

- The TBP problem also falls within the larger area called the Pure Exploration problem.
- In pure exploration problems the learner has to output a set of recommendations either with high confidence (fixed confidence) or after a specified number of rounds (fixed budget).
- Our considered TBP is a fixed budget pure exploration problem.
- Both APT and AugUCB reuses several ideas from Pure exploration problem.

Previous Works (Diagram)



Approach of UCB-Improved (UCB-Imp)

- There is a strong relation between UCB-Imp and AugUCB where the former is used to find *a single optimal arm as quickly as possible*.

Approach of UCB-Improved (UCB-Imp)

- There is a strong relation between UCB-Imp and AugUCB where the former is used to find *a single optimal arm as quickly as possible*.
- The basic idea of UCB-Improved is to divide the horizon into phases or rounds and initialize parameters.

Approach of UCB-Improved (UCB-Imp)

- There is a strong relation between UCB-Imp and AugUCB where the former is used to find *a single optimal arm as quickly as possible*.
- The basic idea of UCB-Improved is to divide the horizon into phases or rounds and initialize parameters.
- Pull all surviving arms equal number of times during a round.

Approach of UCB-Improved (UCB-Imp)

- There is a strong relation between UCB-Imp and AugUCB where the former is used to find *a single optimal arm as quickly as possible*.
- The basic idea of UCB-Improved is to divide the horizon into phases or rounds and initialize parameters.
- Pull all surviving arms equal number of times during a round.
- At the end of the round eliminate some sub-optimal arms (as judged by learner) based on elimination criteria.

Approach of UCB-Improved (UCB-Imp)

- There is a strong relation between UCB-Imp and AugUCB where the former is used to find *a single optimal arm as quickly as possible*.
- The basic idea of UCB-Improved is to divide the horizon into phases or rounds and initialize parameters.
- Pull all surviving arms equal number of times during a round.
- At the end of the round eliminate some sub-optimal arms (as judged by learner) based on elimination criteria.
- Reset parameters and proceed to next round.

UCB-Improved ([Auer and Ortner(2010)])

Algorithm 1 UCB-Improved

Input: Time horizon T

Initialization: Set $B_0 := A$ and $\epsilon_0 := 1$.

for $m = 0, 1, .. \lfloor \frac{1}{2} \log_2 \frac{T}{e} \rfloor$ **do**

  Pull each arm in B_m, $n_m = \left\lceil \frac{2 \log(T\epsilon_m^2)}{\epsilon_m} \right\rceil$ number of times.

  Arm Elimination

    For each $i \in B_m$, delete arm i from B_m if,

$$\hat{r}_i + \sqrt{\frac{\log(T\epsilon_m^2)}{2n_m}} < \max_{j \in B_m} \left\{ \hat{r}_j - \sqrt{\frac{\log(T\epsilon_m^2)}{2n_m}} \right\}$$

  Set $\epsilon_{m+1} := \frac{\epsilon_m}{2}$, Set $B_{m+1} := B_m$

  Stop if $|B_m| = 1$ and pull $i \in B_m$ till n is reached.

end for

Some technical details of UCB-Improved

- We do not know the true means $r_i, \forall i \in A$ of the distributions so we estimate it by the $\tilde{\epsilon}$ by initializing it from 1.

Some technical details of UCB-Improved

- We do not know the true means $r_i, \forall i \in A$ of the distributions so we estimate it by the $\tilde{\epsilon}$ by initializing it from 1.
- All rewards are assume to be bounded between $[0, 1]$ and so $\Delta_i = (r^* - r_i) \in [0, 1], \forall i \in A$ as well.

Some technical details of UCB-Improved

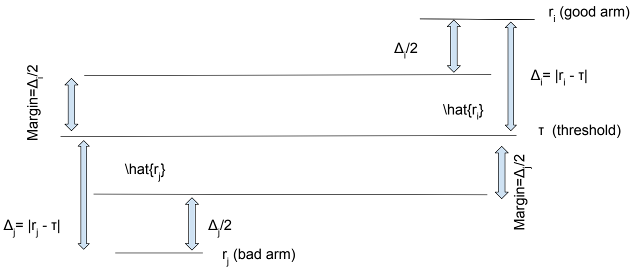
- We do not know the true means $r_i, \forall i \in A$ of the distributions so we estimate it by the $\tilde{\epsilon}$ by initializing it from 1.
- All rewards are assume to be bounded between $[0, 1]$ and so $\Delta_i = (r^* - r_i) \in [0, 1], \forall i \in A$ as well.
- UCB-Improved has fixed confidence interval $c_m = \sqrt{\dfrac{\log(T\epsilon_m^2)}{2n_m}}$ for all arms in a particular phase.

Some technical details of UCB-Improved

- We do not know the true means $r_i, \forall i \in A$ of the distributions so we estimate it by the $\tilde{\epsilon}$ by initializing it from 1.
- All rewards are assume to be bounded between $[0, 1]$ and so $\Delta_i = (r^* - r_i) \in [0, 1], \forall i \in A$ as well.
- UCB-Improved has fixed confidence interval $c_m = \sqrt{\dfrac{\log(T\epsilon_m^2)}{2n_m}}$ for all arms in a particular phase.
- c_m ensures that whenever $\epsilon_m < \dfrac{\Delta_i}{2}$ in the m-th round, the arm i gets eliminated.

AugUCB algorithm

$\Delta_i = |r_i - \tau|$. It is risky to eliminate while estimated mean of arm i ($\hat{r_i}$) is inside Margin. Confidence interval s_i will make sure arm i is not eliminated while inside Margin. So we have to bound $P\{\hat{r_i}\} <= r_i - 2s_i\}$. Till that time arm i will not be accepted as good arm. For arm j we have to bound $P\{\hat{r_j}\} >= r_j + 2s_j\}$ since till that time j will not be rejected as bad arm.



AugUCB algorithm

- Like UCB-Imp, AugUCB also divides the time budget T into rounds.

AugUCB algorithm

- Like UCB-Imp, AugUCB also divides the time budget T into rounds.
- A crucial difference is that in every round instead of pulling all the arms equal number of times we pull the arm that minimizes $j \in \arg\min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$.

AugUCB algorithm

- Like UCB-Imp, AugUCB also divides the time budget T into rounds.
- A crucial difference is that in every round instead of pulling all the arms equal number of times we pull the arm that minimizes $j \in \arg\min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$.
- At every timestep now we run the arm elimination check to eliminate sub-optimal arms.

AugUCB algorithm

- Like UCB-Imp, AugUCB also divides the time budget T into rounds.
- A crucial difference is that in every round instead of pulling all the arms equal number of times we pull the arm that minimizes $j \in \arg\min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$.
- At every timestep now we run the arm elimination check to eliminate sub-optimal arms.
- At the end of the phase we reset the parameters.

AugUCB algorithm

- Like UCB-Imp, AugUCB also divides the time budget T into rounds.
- A crucial difference is that in every round instead of pulling all the arms equal number of times we pull the arm that minimizes $j \in \arg\min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$.
- At every timestep now we run the arm elimination check to eliminate sub-optimal arms.
- At the end of the phase we reset the parameters.
- Note that the length of the phase, the exploration parameters and the confidence interval term $s_i = \sqrt{\frac{\rho\psi_m(\hat{v}_i+1)\log(T\epsilon_m)}{4n_i}}$ are set through detailed theoretical analysis.

AugUCB algorithm I

Input: Time budget T; parameter ρ; threshold τ
Initialization: $B_0 = \mathcal{A}$; $m = 0$; $\epsilon_0 = 1$;

$$M = \left\lfloor \frac{1}{2} \log_2 \frac{T}{e} \right\rfloor; \quad \psi_0 = \frac{T\epsilon_0}{128\left(\log(\frac{3}{16}K\log K)\right)^2};$$

$$\ell_0 = \left\lceil \frac{2\psi_0 \log(T\epsilon_0)}{\epsilon_0} \right\rceil; \quad N_0 = K\ell_0$$

Pull each arm once

for $t = K+1, .., T$ **do**
  Pull arm $j \in \arg\min_{i \in B_m} \left\{ |\hat{r}_i - \tau| - 2s_i \right\}$
  for $i \in B_m$ **do**
    if $(\hat{r}_i + s_i < \tau - s_i)$ or $(\hat{r}_i - s_i > \tau + s_i)$ **then**
      $B_m \leftarrow B_m \backslash \{i\}$ (Arm deletion)
    end if

AugUCB algorithm II

end for

if $t \geq N_m$ and $m \leq M$ **then**

Reset Parameters

$\epsilon_{m+1} \leftarrow \frac{\epsilon_m}{2}$

$B_{m+1} \leftarrow B_m$

$\psi_{m+1} \leftarrow \frac{T\epsilon_{m+1}}{128(\log(\frac{3}{16}K\log K))^2}$

$\ell_{m+1} \leftarrow \left\lceil \frac{2\psi_{m+1}\log(T\epsilon_{m+1})}{\epsilon_{m+1}} \right\rceil$

$N_{m+1} \leftarrow t + |B_{m+1}|\ell_{m+1}$

$m \leftarrow m + 1$

end if

end for

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

Problem Complexity

- We must delve into the notion of hardness which come from the general pure exploration bandit literature.

Problem Complexity

- We must delve into the notion of hardness which come from the general pure exploration bandit literature.
- We define $H_1 = \sum_{i=1}^{K} \dfrac{1}{\Delta_i^2}$ and $H_2 = \min_{i \in \mathcal{A}} \dfrac{i}{\Delta_{(i)}^2}$

Problem Complexity

- We must delve into the notion of hardness which come from the general pure exploration bandit literature.
- We define $H_1 = \sum_{i=1}^{K} \frac{1}{\Delta_i^2}$ and $H_2 = \min_{i \in \mathcal{A}} \frac{i}{\Delta_{(i)}^2}$
- The relationship between H_1 and H_2 can be derived as,

$$H_1 \leq \log(2K) H_2 \text{ and } H_1 \leq \log(K) H_{CSAR,2}.$$

Problem Complexity

- For variance aware algorithm H_1^σ ([Gabillon et al.(2011)Gabillon, Ghavamzadeh, Lazaric, and Bubeck]) 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 variance aware algorithm H_1^σ ([Gabillon et al.(2011)Gabillon, Ghavamzadeh, Lazaric, and Bubeck]) 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 variance aware algorithm H_1^{σ} ([Gabillon et al.(2011)Gabillon, Ghavamzadeh, Lazaric, and Bubeck]) 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

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

Problem Complexity

- For variance aware algorithm H_1^σ ([Gabillon et al.(2011)Gabillon, Ghavamzadeh, Lazaric, and Bubeck]) 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

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

- 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 are very low we can say that $H_{\sigma,1} < H_1$.

Expected Loss

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
AugUCB	$\exp\left(-\frac{T}{4096\log(K\log K)H_{\sigma,2}} + \log(2KT)\right)$
UCBEV	$\exp\left(-\frac{1}{512}\frac{T-2K}{H_{\sigma,1}} + \log(6KT)\right)$
APT	$\exp\left(-\frac{T}{64H_1} + 2\log((\log(T)+1)K)\right)$
CSAR	$\exp\left(-\frac{T-K}{72\log(K)H_{CSAR,2}} + 2\log(K)\right)$

Sketch of the proof

- The proof comprises of two modules. In the first module we investigate the necessary conditions for arm elimination within a specified number of rounds, which is motivated by the technique in UCB-Imp.

Sketch of the proof

- The proof comprises of two modules. In the first module we investigate the necessary conditions for arm elimination within a specified number of rounds, which is motivated by the technique in UCB-Imp.
- We bound the arm-elimination probability by Bernstein inequality (as in [Audibert et al.(2009)Audibert, Munos, and Szepesvári]) rather that the Chernoff-Hoeffding bounds (used in UCB-Imp).

Sketch of the proof

- The proof comprises of two modules. In the first module we investigate the necessary conditions for arm elimination within a specified number of rounds, which is motivated by the technique in UCB-Imp.
- We bound the arm-elimination probability by Bernstein inequality (as in [Audibert et al.(2009)Audibert, Munos, and Szepesvári]) rather that the Chernoff-Hoeffding bounds (used in UCB-Imp).
- In the second module,we define a favourable event that will yield an upper bound on the expected loss and use union bound and module-1 (on the arm elimination probability) to derive the result through a series of simplifications.

Finally, experiment!!!

- We experiment with APT, AugUCB, UCBE, UCBEV, CSAR, UA.

Finally, experiment!!!

- We experiment with APT, AugUCB, UCBE, UCBEV, CSAR, 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 experiment with APT, AugUCB, UCBE, UCBEV, CSAR, 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, CSAR, UA do not require access to H_1 or $H_{\sigma,1}$.

Finally, experiment!!!

- We experiment with APT, AugUCB, UCBE, UCBEV, CSAR, 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, CSAR, 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.

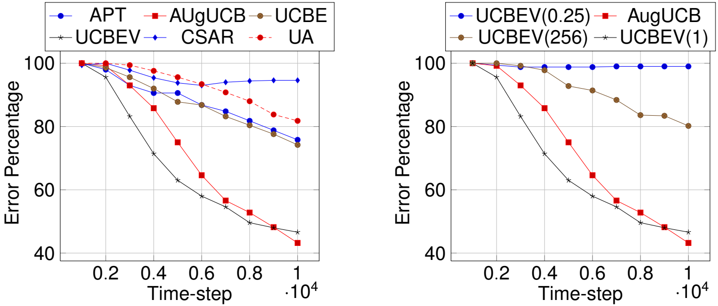
Experimental Setup

- 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 Result



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

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

References I

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári.
Improved algorithms for linear stochastic bandits.
In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Jacob D Abernethy, Kareem Amin, and Ruihao Zhu.
Threshold bandits, with and without censored feedback.
In *Advances In Neural Information Processing Systems*, pages 4889–4897, 2016.

Shipra Agrawal and Navin Goyal.
Analysis of thompson sampling for the multi-armed bandit problem.
arXiv preprint arXiv:1111.1797, 2011.

References II

Jean-Yves Audibert and Sébastien Bubeck.
Minimax policies for adversarial and stochastic bandits.
In *COLT*, pages 217–226, 2009.

Jean-Yves Audibert and Sébastien Bubeck.
Best arm identification in multi-armed bandits.
In *COLT-23th Conference on Learning Theory-2010*, pages 13–p, 2010.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári.
Exploration–exploitation tradeoff using variance estimates in multi-armed bandits.
Theoretical Computer Science, 410(19):1876–1902, 2009.

References III

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

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer.
Finite-time analysis of the multiarmed bandit problem.
Machine learning, 47(2-3):235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire.
The nonstochastic multiarmed bandit problem.
SIAM Journal on Computing, 32(1):48–77, 2002b.

References IV

Dimitri P Bertsekas and John N Tsitsiklis.
Neuro-dynamic programming (optimization and neural computation series, 3).
Athena Scientific, 7:15–23, 1996.

Sébastien Bubeck and Nicolo Cesa-Bianchi.
Regret analysis of stochastic and nonstochastic multi-armed bandit problems.
arXiv preprint arXiv:1204.5721, 2012.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz.
Pure exploration in finitely-armed and continuous-armed bandits.
Theoretical Computer Science, 412(19):1832–1852, 2011.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Gábor Lugosi.
Bandits with heavy tail.
arXiv preprint arXiv:1209.1727, 2012.

References V

Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet.
Bounded regret in stochastic multi-armed bandits.
arXiv preprint arXiv:1302.1611, 2013a.

Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan.
Multiple identifications in multi-armed bandits.
In *ICML (1)*, pages 258–265, 2013b.

Olivier Cappe, Aurelien Garivier, and Emilie Kaufmann.
pymabandits, 2012.
http://mloss.org/software/view/415/.

Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen.
Combinatorial pure exploration of multi-armed bandits.
In *Advances in Neural Information Processing Systems*, pages 379–387, 2014.

References VI

Eyal Even-Dar, Shie Mannor, and Yishay Mansour.
Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems.
The Journal of Machine Learning Research, 7:1079–1105, 2006.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani.
The elements of statistical learning, volume 1.
Springer series in statistics Springer, Berlin, 2001.

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck.
Multi-bandit best arm identification.
In *Advances in Neural Information Processing Systems*, pages 2222–2230, 2011.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric.
Best arm identification: A unified approach to fixed budget and fixed confidence.
In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2012.

Aurélien Garivier and Olivier Cappé.
The kl-ucb algorithm for bounded stochastic bandits and beyond.
arXiv preprint arXiv:1102.2490, 2011.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al.
Bayesian reinforcement learning: a survey.
World Scientific, 2015.

References VIII

Junya Honda and Akimichi Takemura.
An asymptotically optimal bandit algorithm for bounded support models.
In *COLT*, pages 67–79. Citeseer, 2010.

Kevin Jamieson and Robert Nowak.
Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting.
In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–6. IEEE, 2014.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone.
Pac subset selection in stochastic multi-armed bandits.
In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 655–662, 2012.

References IX

Tze Leung Lai and Herbert Robbins.
Asymptotically efficient adaptive allocation rules.
Advances in applied mathematics, 6(1):4–22, 1985.

Tor Lattimore.
Optimally confident ucb: Improved regret for finite-armed bandits.
arXiv preprint arXiv:1507.07880, 2015.

Yun-Ching Liu and Yoshimasa Tsuruoka.
Modification of improved upper confidence bounds for regulating exploration in monte-carlo tree search.
Theoretical Computer Science, 2016.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier.
An optimal algorithm for the thresholding bandit problem.
arXiv preprint arXiv:1605.08671, 2016.

Shie Mannor and John N Tsitsiklis.
The sample complexity of exploration in the multi-armed bandit problem.
Journal of Machine Learning Research, 5(Jun):623–648, 2004.

Mukherjee, Naveen, Nandan, and Ravindran.
Thresholding bandits with augmented UCB.
CoRR, abs/1704.02281, 2017.
URL http://arxiv.org/abs/1704.02281.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg.
Batched bandit problems.
arXiv preprint arXiv:1505.00369, 2015.

Herbert Robbins.
Some aspects of the sequential design of experiments.
In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1952.

Richard S Sutton and Andrew G Barto.
Reinforcement learning: An introduction.
MIT press, 1998.

William R Thompson.
On the likelihood that one unknown probability exceeds another in view of the evidence of two samples.
Biometrika, pages 285–294, 1933.

David Tolpin and Solomon Eyal Shimony.
Mcts based on simple regret.
In *AAAI*, 2012.

Thank You