The thresholding Bandit Problem under Shape Constraints

Alexandra Carpentier[1]

OvGU Magdeburg

Based on joint works with **James Cheshire, Andréa Locatelli, Maurilio Gutzeit, Pierre Ménard**

Dec. 15th, 2020

[1]Partly funded by the DFG EN CA1488, the CRC 1294, the GK 2297, the GK 2331, and the SFI RE-BCI.

Introduction

Topic of this talk: Sequential learning and shape constraints.

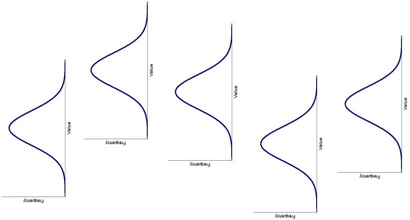
Based on the following joint works:

- Locatelli, Andrea, Maurilio Gutzeit, and Alexandra Carpentier. "An optimal algorithm for the thresholding bandit problem." ICML 2016.
- Cheshire, James, Pierre Menard, and Alexandra Carpentier. "The Influence of Shape Constraints on the Thresholding Bandit Problem." COLT, 2020.
- Cheshire, James, Pierre Menard, and Alexandra Carpentier. "Problem Dependent View on Structured Thresholding Bandit Problems." Working paper, 2020+.

Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

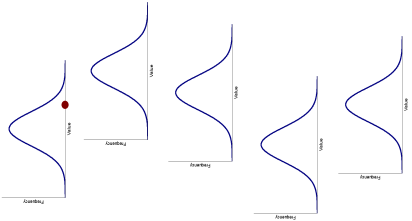
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

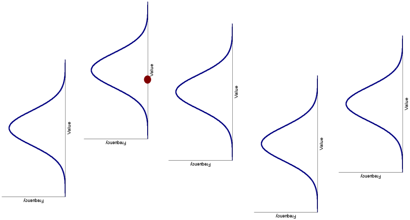
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

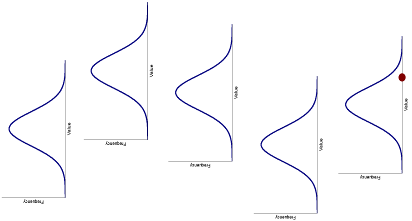
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

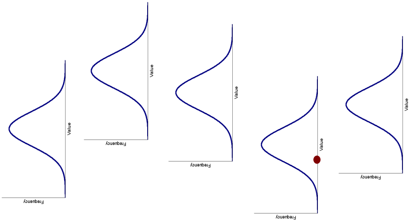
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

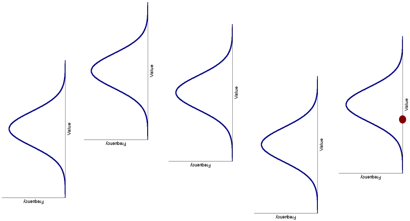
- Distributions $(\nu_k)_{k\leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

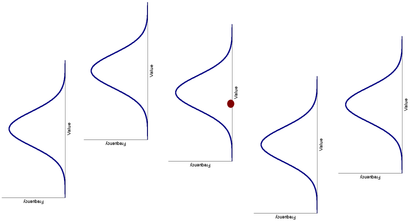
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

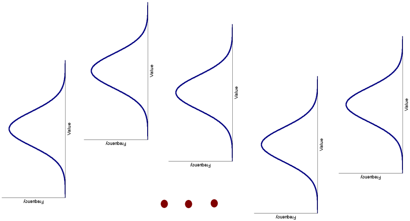
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

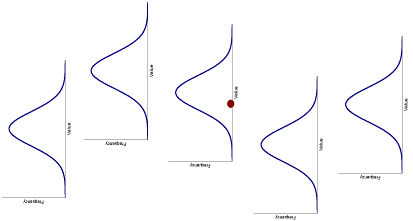
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

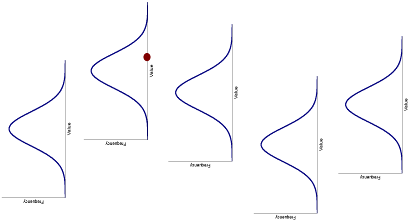
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

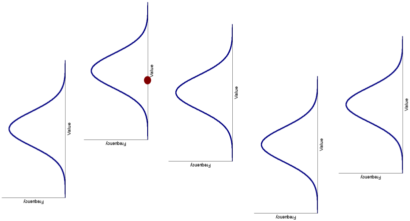
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

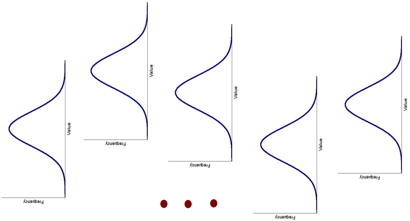
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

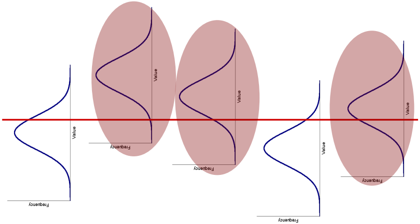
- Distributions $(\nu_k)_{k\leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil

This is the *thresholding bandit problem*, i.e. given a threshold τ, and writing μ_k for the mean of distribution k, we aim at predicting

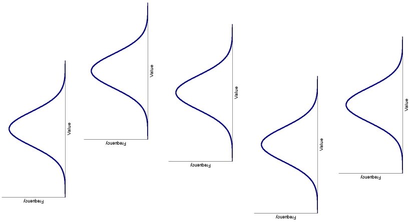
$$Q = (\mathrm{sign}(\mu_k - \tau))_k \,.$$

Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil

Bandit vocabulary:



Sequential learning

Resource optimisation in face of uncertainty : See [Thompson (1933)], [Robbins (1952)], [Gittins (1979)], [Whittle (1988)], [Cappé et al. (2013)], [Munos (2014)], etc.

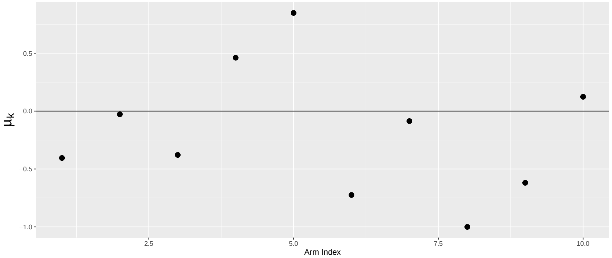
- Distributions $(\nu_k)_{k \leq K}$ with *unknown* means μ_k
- Limited sampling resources T
- At each time t, choose k_t and collect $X_t \sim \nu_{k_t}$
- Objective to fulfil

Bandit vocabulary:

Outline

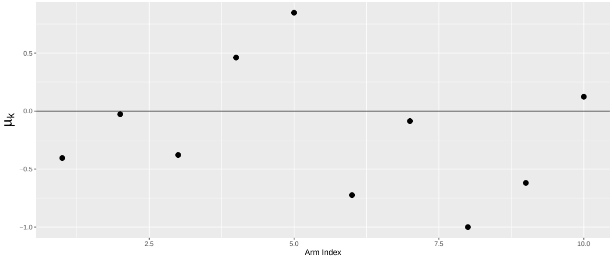
Problem setting: K arms, budget T, threshold $\tau = 0$

- Each arm $k \in [K]$ corresponds to a distribution $\mathcal{N}(\mu_k, 1)$ with mean $\mu_k \in [-1, 1]$ - and we set $\tau = 0$.
- At each round $t < T$ the learner pulls an arm $k_t \in [K]$ and observes a sample $X_t \sim \mathcal{N}(\mu_{k_t}, 1)$.
- Upon exhaustion of the budget the learner is required to output a prediction $\hat{Q} \in \{-1, 1\}^K$ of $Q = \text{sign}(\mu_k)$.



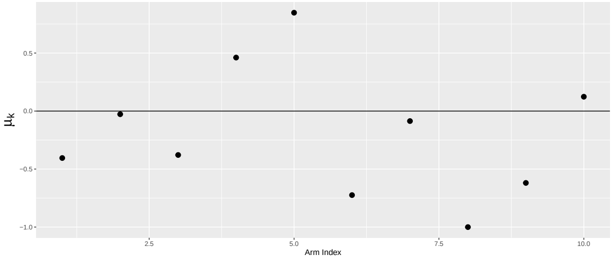
Problem setting: K arms, budget T, threshold $\tau = 0$

- Each arm $k \in [K]$ corresponds to a distribution $\mathcal{N}(\mu_k, 1)$ with mean $\mu_k \in [-1, 1]$ - and we set $\tau = 0$.
- At each round $t < T$ the learner pulls an arm $k_t \in [K]$ and observes a sample $X_t \sim \mathcal{N}(\mu_{k_t}, 1)$.
- Upon exhaustion of the budget the learner is required to output a prediction $\hat{Q} \in \{-1, 1\}^K$ of $Q = \text{sign}(\mu_k)$.



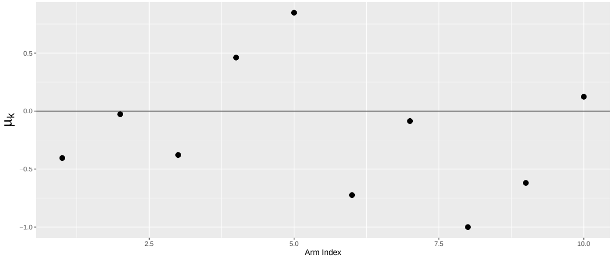
Problem setting: K arms, budget T, threshold $\tau = 0$

- Each arm $k \in [K]$ corresponds to a distribution $\mathcal{N}(\mu_k, 1)$ with mean $\mu_k \in [-1, 1]$ - and we set $\tau = 0$.
- At each round $t < T$ the learner pulls an arm $k_t \in [K]$ and observes a sample $X_t \sim \mathcal{N}(\mu_{k_t}, 1)$.
- Upon exhaustion of the budget the learner is required to output a prediction $\hat{Q} \in \{-1, 1\}^K$ of $Q = \text{sign}(\mu_k)$.



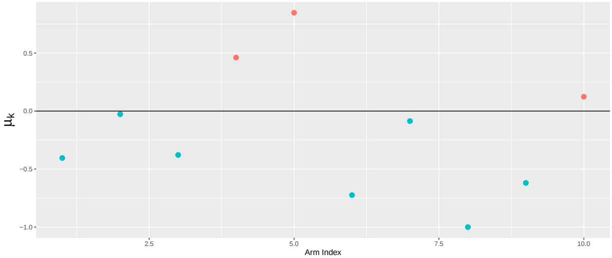
Problem setting: K arms, budget T, threshold $\tau = 0$

- Each arm $k \in [K]$ corresponds to a distribution $\mathcal{N}(\mu_k, 1)$ with mean $\mu_k \in [-1, 1]$ - and we set $\tau = 0$.
- At each round $t < T$ the learner pulls an arm $k_t \in [K]$ and observes a sample $X_t \sim \mathcal{N}(\mu_{k_t}, 1)$.
- Upon exhaustion of the budget the learner is required to output a prediction $\hat{Q} \in \{-1, 1\}^K$ of $Q = \mathrm{sign}(\mu_k)$.

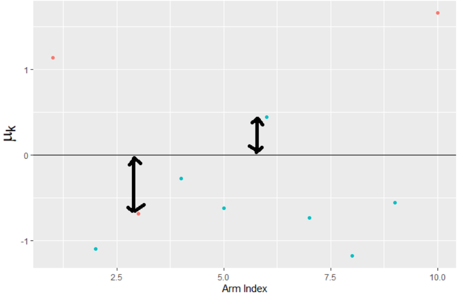


Problem setting: K arms, budget T, threshold $\tau = 0$

- Each arm $k \in [K]$ corresponds to a distribution $\mathcal{N}(\mu_k, 1)$ with mean $\mu_k \in [-1, 1]$ - and we set $\tau = 0$.
- At each round $t < T$ the learner pulls an arm $k_t \in [K]$ and observes a sample $X_t \sim \mathcal{N}(\mu_{k_t}, 1)$.
- Upon exhaustion of the budget the learner is required to output a prediction $\hat{Q} \in \{-1, 1\}^K$ of $Q = \text{sign}(\mu_k)$.



Regret



Two measures of regret:

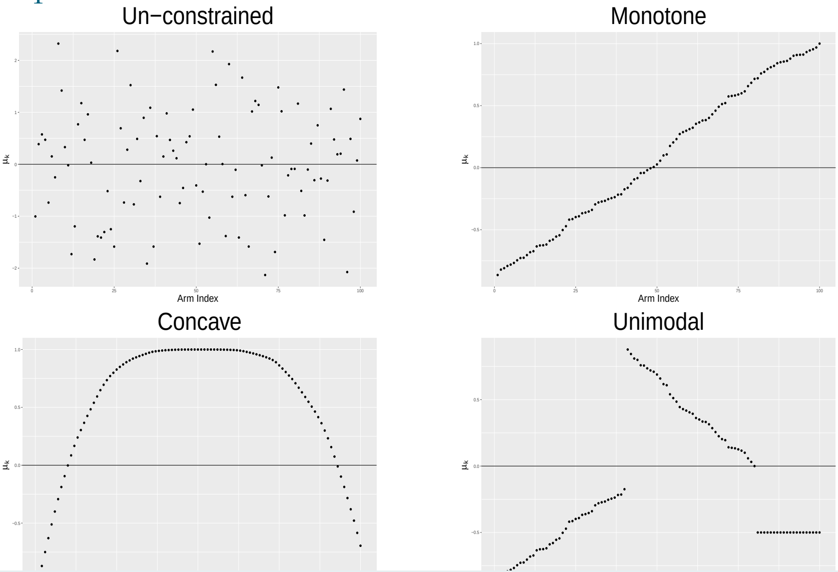
- Probability of error:

$$e_T := \mathbb{P}(\hat{Q} \neq Q).$$

- Simple regret:

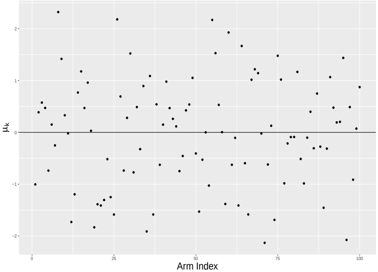
$$r_T := \mathbb{E} \max_{k:\hat{Q}[k] \neq Q[k]} |\mu_k|.$$

Shape constraints

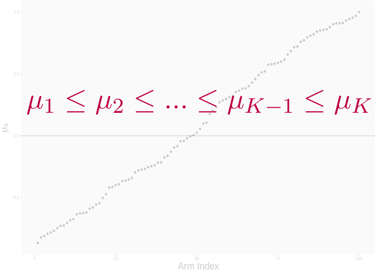


Shape constraints

Un−constrained

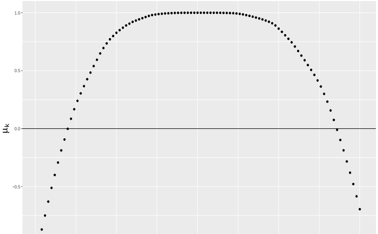


Monotone

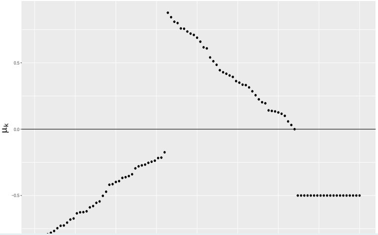


$$\mu_1 \leq \mu_2 \leq \ldots \leq \mu_{K-1} \leq \mu_K$$

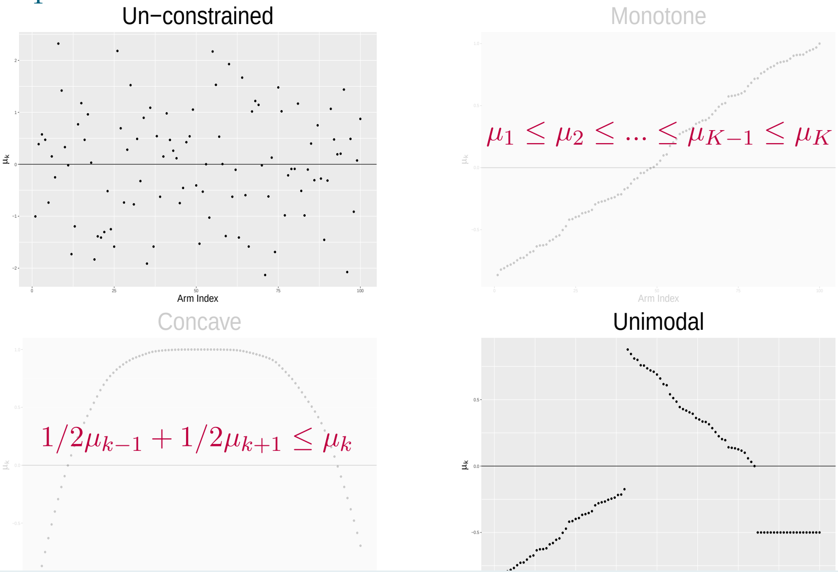
Concave



Unimodal



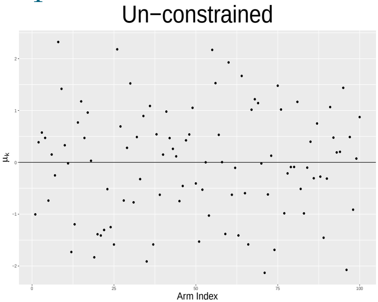
Shape constraints

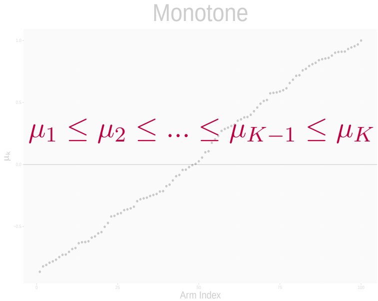


$$\mu_1 \leq \mu_2 \leq \ldots \leq \mu_{K-1} \leq \mu_K$$

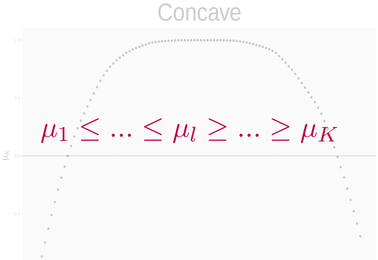
$$1/2\mu_{k-1} + 1/2\mu_{k+1} \leq \mu_k$$

Shape constraints

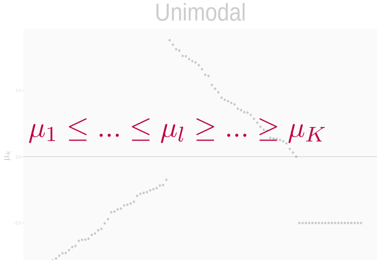




$$\mu_1 \leq \mu_2 \leq \ldots \leq \mu_{K-1} \leq \mu_K$$



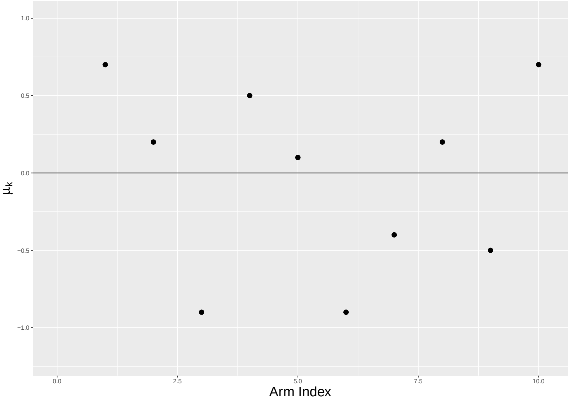
$$\mu_1 \leq \ldots \leq \mu_l \geq \ldots \geq \mu_K$$



$$\mu_1 \leq \ldots \leq \mu_l \geq \ldots \geq \mu_K$$

Outline

Unconstrained setting



Unconstrained setting

- **Fixed confidence setting:** [Chen et al, 2016], etc
- **Fixed budget setting:** [Chen et al, 2014], [Locatelli et al., 2016], [Mukherjee, et al. 2017], [Jie et al, 2017], etc.

  In all these papers: problem dependent results.

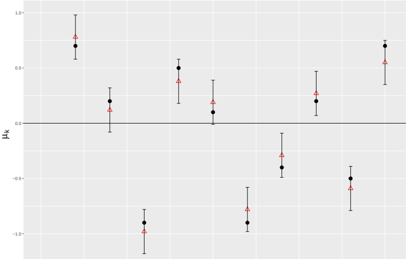
Unconstrained setting: problem independent results

Theorem (Cheshire et. al, 2020)

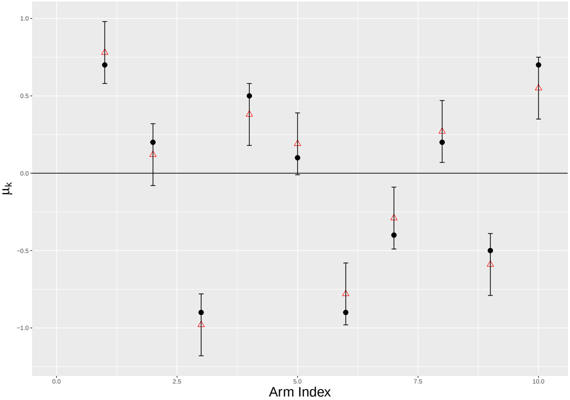
It holds that

$$\inf_{\text{algo}} \sup_{\text{problem}} r_T \asymp \sqrt{\frac{K\log(K)}{T}}.$$

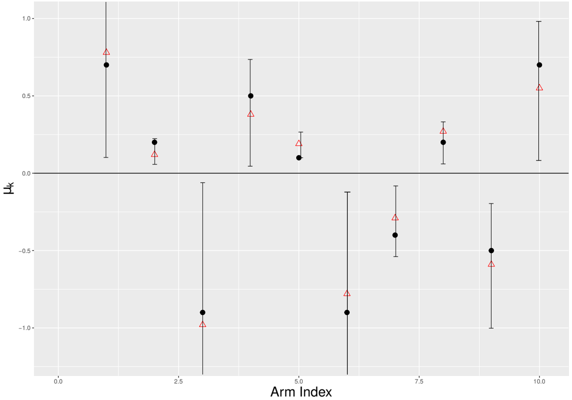
Upper bound trivial (uniform sampling), lower bound somewhat more tricky than in batch setting.



Unconstrained setting: problem dependent results



Unconstrained setting: problem dependent results



Unconstrained setting: problem dependent results

In what follows: write the gaps

$$\Delta_i = |\mu_i|,$$

and $\mathcal{M}_\Delta$ the set of problems with gaps Δ.

Theorem (Locatelli et al., 2016)

For any vector of gaps Δ it holds that

$$K \log(n) \exp(-\Box T/H) \gtrsim \inf_{\text{algo}} \sup_{\text{problem in } \mathcal{M}_\Delta} e_T \gtrsim \exp(-\Box T/H),$$

where $H = \sum_i \Delta_i^{-2}$.

Unconstrained setting: problem dependent results

In what follows: write the gaps

$$\Delta_i = |\mu_i|,$$

and $\mathcal{M}_\Delta$ the set of problems with gaps Δ.

Theorem (Locatelli et al., 2016)

For any vector of gaps Δ it holds that

$$K \log(n) \exp(-\Box T/H) \gtrsim \inf_{\text{algo}} \sup_{\text{problem in } \mathcal{M}_\Delta} e_T \gtrsim \exp(-\Box T/H),$$

where $H = \sum_i \Delta_i^{-2}$.

APT algorithm: sample at time t

$$k_t \in \arg\min_k T_{k,t} |\hat{\mu}_{k,t}|^2.$$

Conclusion

Theorem (Unconstrained setting, (Locatelli et al, 2016), (Cheshire et al, 2020))

It holds that

$$\inf_{\text{algo}} \sup_{\text{problem}} r_T \approx \sqrt{\frac{K \log K}{T}},$$

and for $T \gtrsim \log K \vee \log \log n$ *and any* $\bar{\Delta}$

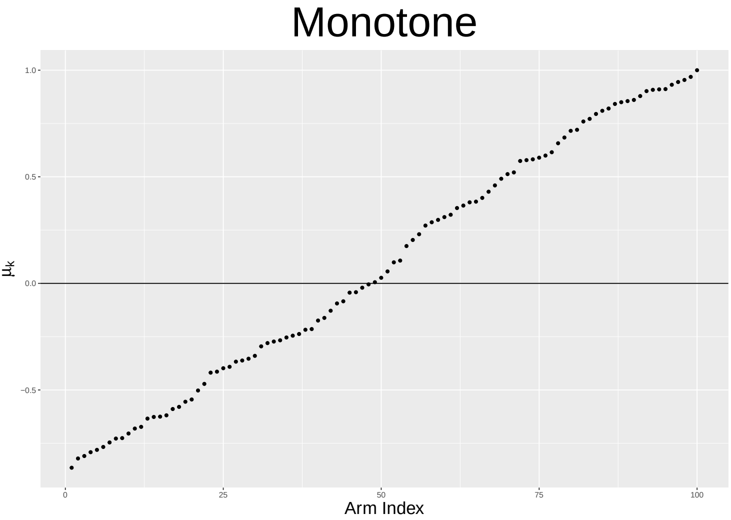
$$\inf_{\text{algo}} \sup_{\bar{\Delta}-\text{problem}} \log e_T \asymp -T/H.$$

Summary until now...

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K \log K}{T}}$			
Pr. dep. $\log e_t$	$-T/H$			

Outline

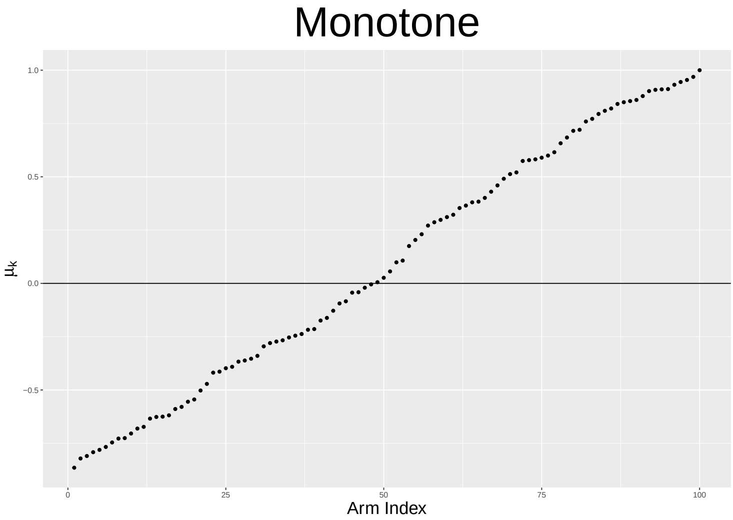
Monotone setting



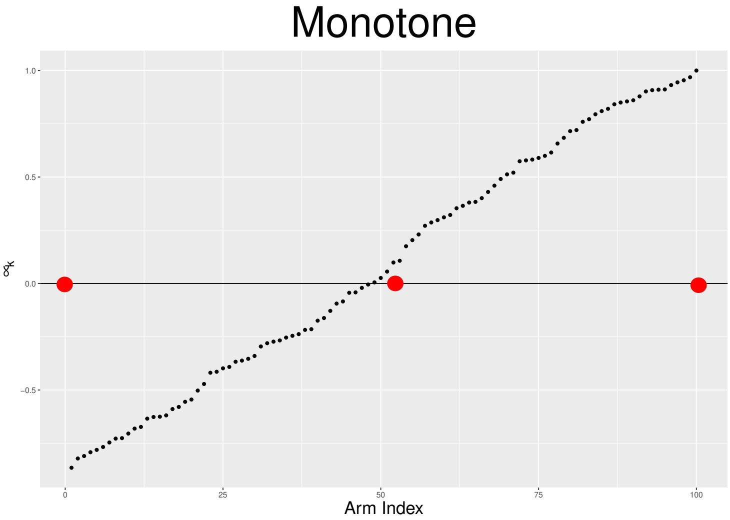
Monotone setting: related literature

- **Noisy binary search with corrections:** [Feige et.al, 1994], [Nowak et.al, 2009], [Karp et,al, 2007], [Ben et.al, 2008], [Emamjomeh et.al, 2016], etc
- **Problem dependent and fixed confidence (non-explicit):** [Garivier et.al, 2017]

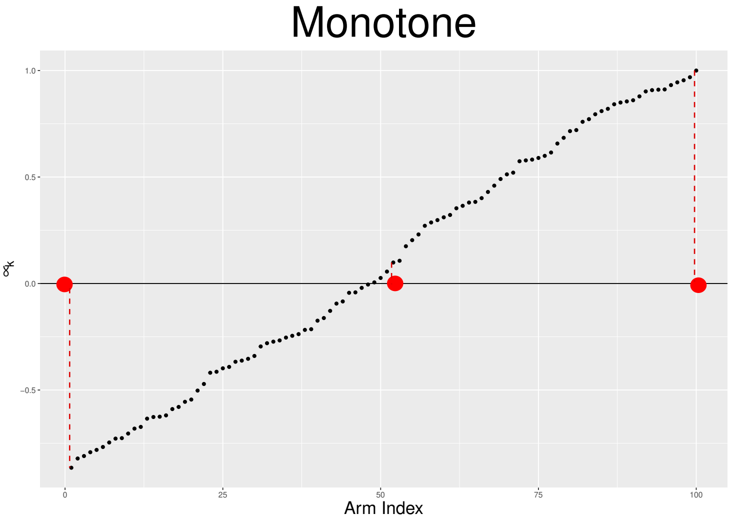
Monotone setting - naive binary search



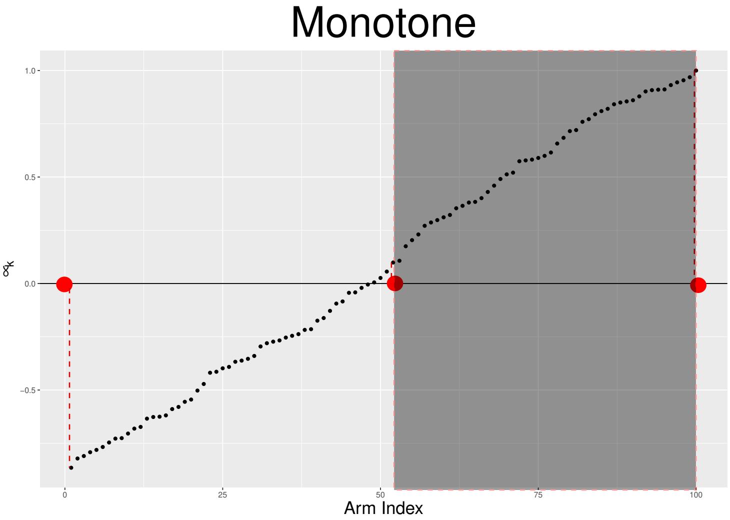
Monotone setting - naive binary search



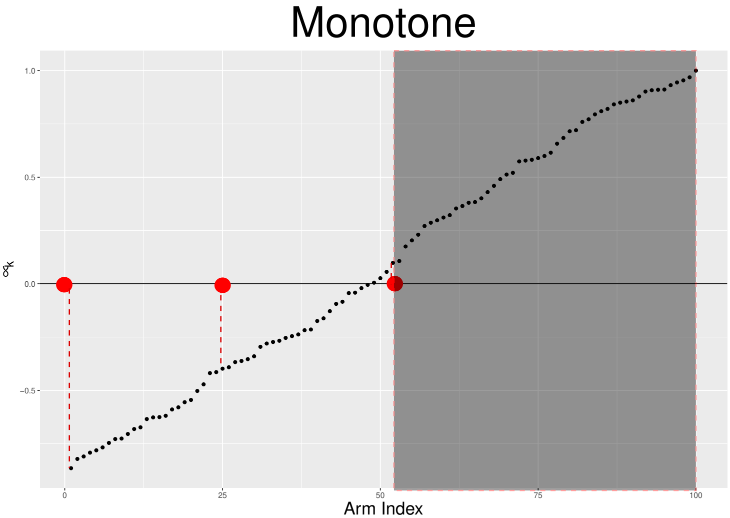
Monotone setting - naive binary search



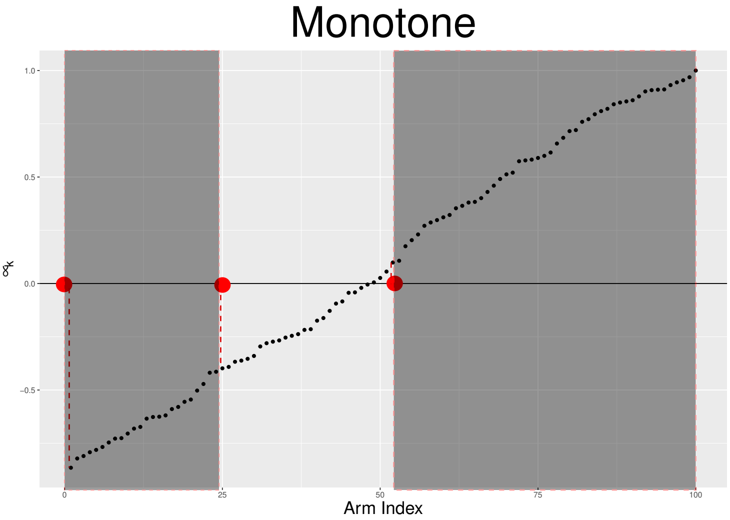
Monotone setting - naive binary search



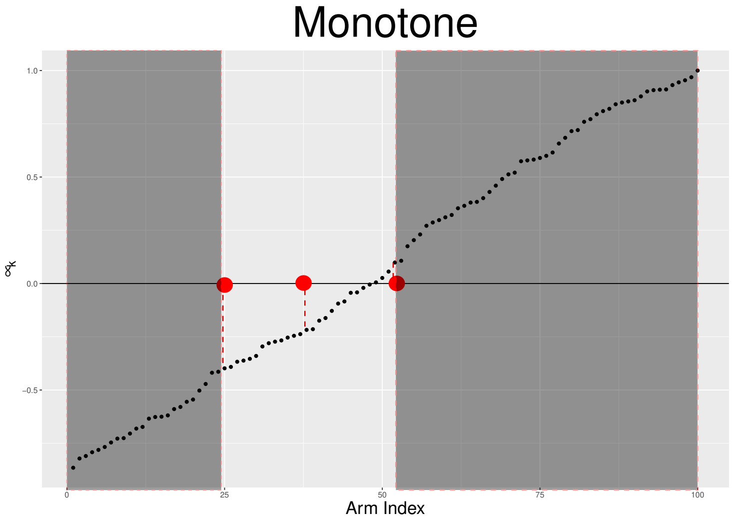
Monotone setting - naive binary search



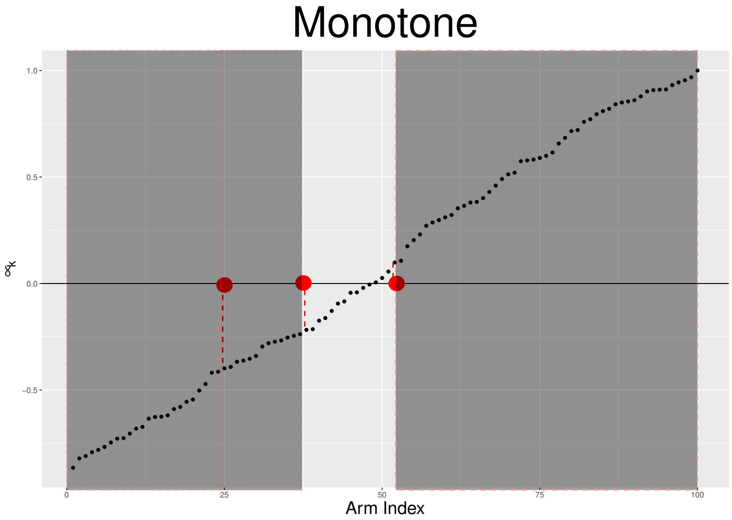
Monotone setting - naive binary search



Monotone setting - naive binary search



Monotone setting - naive binary search



Monotone setting - naive binary search

- Unconstrained setting, uniform sampling on K arms

$$r_T \lesssim \sqrt{\frac{K\log(K)}{T}},$$

and

$$e_T \lesssim \exp(-\square T/H).$$

- Monotone setting, naive binary search (sampling $\log(K)$ arms)

$$r_T \leq \sqrt{\frac{\log(K)\log(\log(K))}{T}},$$

and for $T \geq \log K$

$$e_T \lesssim \log K \exp(-\square T \min_k \Delta_k^2/\log K).$$

Monotone setting - naive binary search

- Unconstrained setting, uniform sampling on K arms

$$r_T \lesssim \sqrt{\frac{K \log(K)}{T}},$$

and

$$e_T \lesssim \exp(-\Box T/H).$$

- Monotone setting, naive binary search (sampling $\log(K)$ arms)

$$r_T \leq \sqrt{\frac{\log(K) \log\left(\log(K)\right)}{T}},$$

and for $T \geq \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2 / \log K).$$

Monotone setting - lower bound

Theorem (Lower bound, monotone case (Cheshire et al, 2020, 2020+))

For any algorithm, there exists a monotone problem such that:

$$r_T \gtrsim \sqrt{\frac{\log(K)}{T}},$$

and a monotone problem such that $\Delta_i = \bar{\Delta}_i, \forall i$ *and such that*

$$e_T \gtrsim \exp(-\square T \min_k \bar{\Delta}_k^2).$$

Reminder: naive binary search reaches

$$r_T \leq \sqrt{\frac{\log(K) \log(\log(K))}{T}}, \quad e_T \lesssim \log K \exp(-\square T \min_k \Delta_k^2 / \log K),$$

(for second one, if $T \geq \log K$).

Monotone setting - beyond naive binary search

We can do better! **Idea:**

- Do a ‘slightly longer’ binary search on a tree - maintain an ‘active segment’
- At each step of the binary search sample arms $\{l, m, r\}$
- If an inconsistency is detected, i.e. $\hat{\mu}_l > 0$ backtrack
- **Good decisions will outweigh bad decisions on average**

Monotone setting - beyond naive binary search

We can do better! **Idea:**

- Do a 'slightly longer' binary search on a tree - maintain an 'active segment'
- At each step of the binary search sample arms $\{l, m, r\}$
- If an inconsistency is detected, i.e. $\hat{\mu}_l > 0$ backtrack
- **Good decisions will outweigh bad decisions on average**

Monotone setting - beyond naive binary search

We can do better! **Idea:**

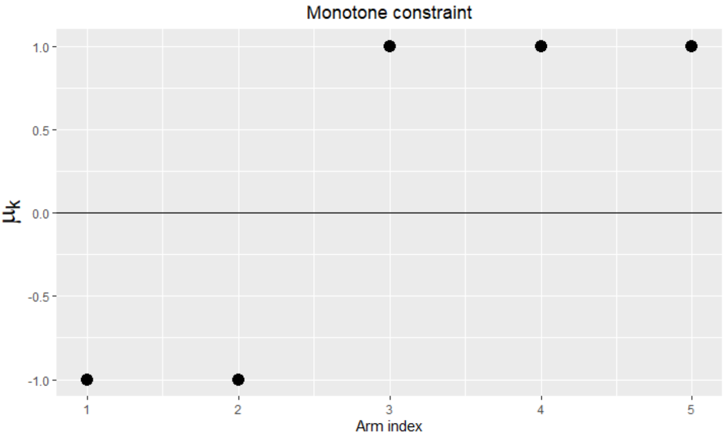
- Do a 'slightly longer' binary search on a tree - maintain an 'active segment'
- At each step of the binary search sample arms $\{l, m, r\}$
- If an inconsistency is detected, i.e. $\hat{\mu}_l > 0$ backtrack
- **Good decisions will outweigh bad decisions on average**

Monotone setting - beyond naive binary search

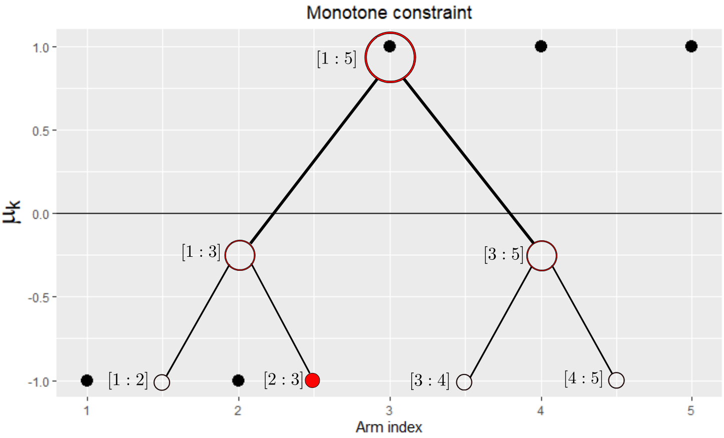
We can do better! **Idea:**

- Do a ‘slightly longer’ binary search on a tree - maintain an ‘active segment’
- At each step of the binary search sample arms $\{l, m, r\}$
- If an inconsistency is detected, i.e. $\hat{\mu}_l > 0$ backtrack
- **Good decisions will outweigh bad decisions on average**

Binary tree



Binary tree

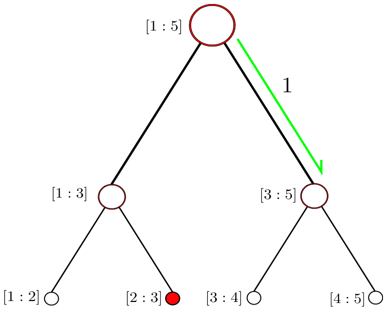

Monotone constraint
1.0
0.5
0.0
-0.5
-1.0
μ_k
[1 : 5]
[1 : 3]
[3 : 5]
[1 : 2]
[2 : 3]
[3 : 4]
[4 : 5]
1
2
3
4
5
Arm index

Binary search with corrections

Sample each arm $c\frac{T}{\log K}$ for a small $c > 0$

For a given time step define a *good decision* as the event,

$$\forall k \in \{l, m, r\}, |\hat{\mu}_k - \mu_k| \leq \sqrt{\frac{\log(K)}{T}}$$



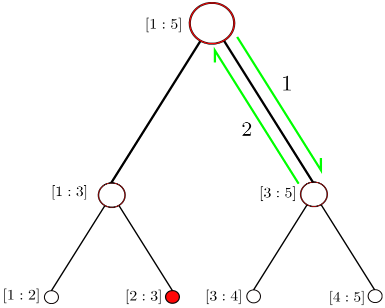
$$1: \quad \hat{\mu}_1 < 0, \; \hat{\mu}_3 < 0, \; \hat{\mu}_5 > 0$$

Binary search with corrections

Sample each arm $c\frac{T}{\log K}$ for a small $c > 0$

For a given time step define a *good decision* as the event,

$$\forall k \in \{l, m, r\}, |\hat{\mu}_k - \mu_k| \leq \sqrt{\frac{\log(K)}{T}}$$



$$1: \quad \hat{\mu}_1 < 0, \ \hat{\mu}_3 < 0, \ \hat{\mu}_5 > 0$$

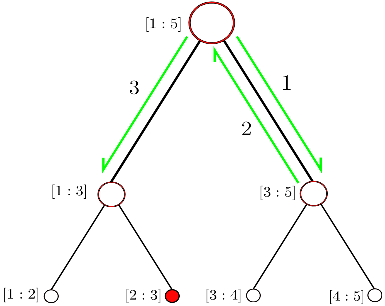
$$2: \quad \hat{\mu}_3 > 0, \ \hat{\mu}_4 > 0, \ \hat{\mu}_5 > 0$$

Binary search with corrections

Sample each arm $c\frac{T}{\log K}$ for a small $c > 0$

For a given time step define a *good decision* as the event,

$$\forall k \in \{l, m, r\}, |\hat{\mu}_k - \mu_k| \leq \sqrt{\frac{\log(K)}{T}}$$



$$1 : \quad \hat{\mu}_1 < 0, \ \hat{\mu}_3 < 0, \ \hat{\mu}_5 > 0$$

$$2 : \quad \hat{\mu}_3 > 0, \ \hat{\mu}_4 > 0, \ \hat{\mu}_5 > 0$$

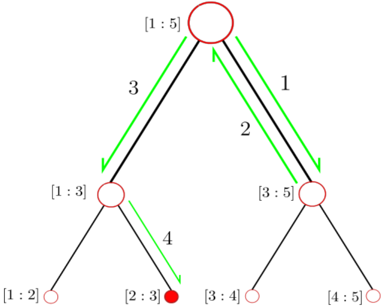
$$3 : \quad \hat{\mu}_1 < 0, \ \hat{\mu}_3 > 0, \ \hat{\mu}_5 > 0$$

Binary search with corrections

Sample each arm $c\frac{T}{\log K}$ for a small $c > 0$

For a given time step define a *good decision* as the event,

$$\forall k \in \{l, m, r\}, |\hat{\mu}_k - \mu_k| \leq \sqrt{\frac{\log(K)}{T}}$$



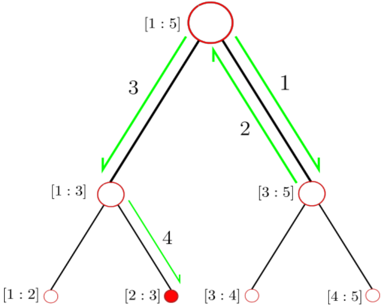
$$\begin{aligned} 1 :& \quad \hat{\mu}_1 < 0,\ \hat{\mu}_3 < 0,\ \hat{\mu}_5 > 0 \\ 2 :& \quad \hat{\mu}_3 > 0,\ \hat{\mu}_4 > 0,\ \hat{\mu}_5 > 0 \\ 3 :& \quad \hat{\mu}_1 < 0,\ \hat{\mu}_3 > 0,\ \hat{\mu}_5 > 0 \\ 4 :& \quad \hat{\mu}_1 < 0,\ \hat{\mu}_2 < 0,\ \hat{\mu}_3 > 0 \end{aligned}$$

Binary search with corrections

Sample each arm $c\frac{T}{\log K}$ for a small $c > 0$

For a given time step define a *good decision* as the event,

$$\forall k \in \{l, m, r\}, |\hat{\mu}_k - \mu_k| \leq \sqrt{\frac{\log(K)}{T}}$$



On average, e.g. four time more good decisions than bad decisions! → auto-correction.

Monotone case - upper bounds

Theorem (Upper bounds (Cheshire et. al, 2020,2020+))

The strategy with auto-correction and appropriated choice of cut-off satisfies on any problem

$$r_T \lesssim \sqrt{\frac{\log K}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \exp(-\Box T \min_k \Delta_k^2).$$

Remark: the cut-off has to be taken carefully.

Conclusion

Theorem (Monotone setting (Cheshire et. al, 2020,2020+))

It holds that

$$\inf_{\text{algo}} \sup_{\text{problem}} r_T \asymp \sqrt{\frac{\log K}{T}},$$

and for $T \geq \log K$ *and any* $\bar{\Delta}$

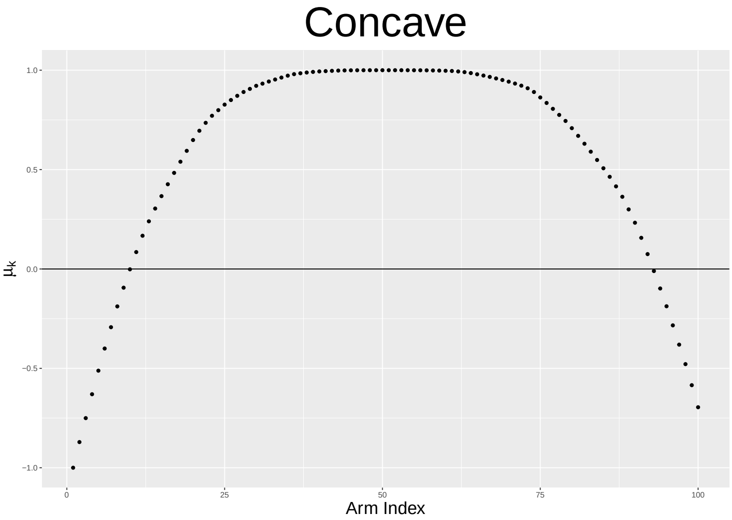
$$\inf_{\text{algo}} \sup_{\bar{\Delta}-\text{problem}} \log e_T \asymp -T \min_k \bar{\Delta}_k^2.$$

Summary until now...

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K \log K}{T}}$	$\sqrt{\frac{\log K}{T}}$		
Pr. dep. $\log e_t$	$-T/H$	$-T \min_k \Delta_k^2$		

Outline

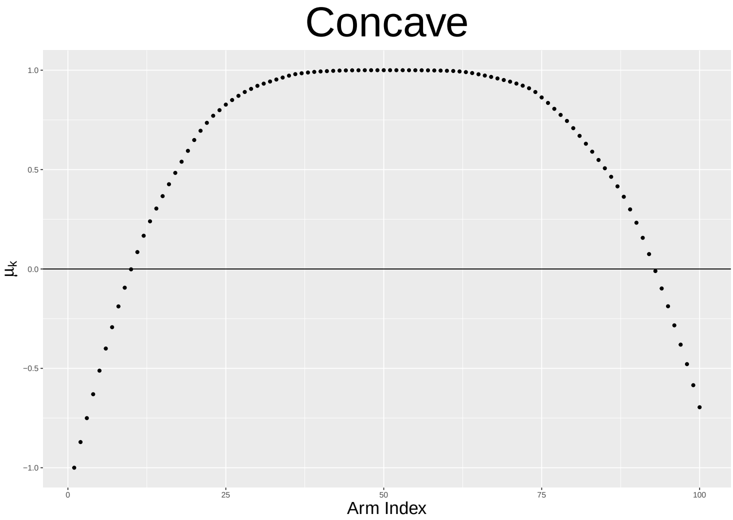
Concave setting



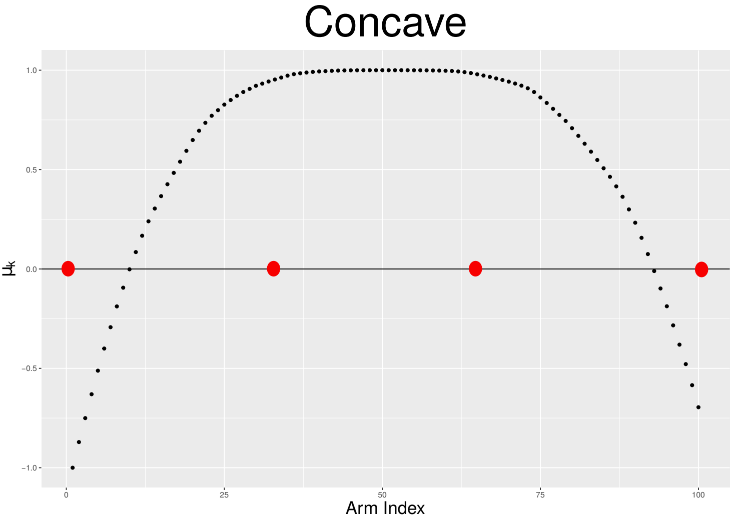
Concave setting: related literature

- **zeroth order noisy convex optimisation:** [Nemirovski and Yudin, 1983], [Wang et al., 2017], [Agarwal et al., 2011], [Liang et al., 2014], etc
- **Estimation of concave functions:** [Simchowitz et al, 2018], etc

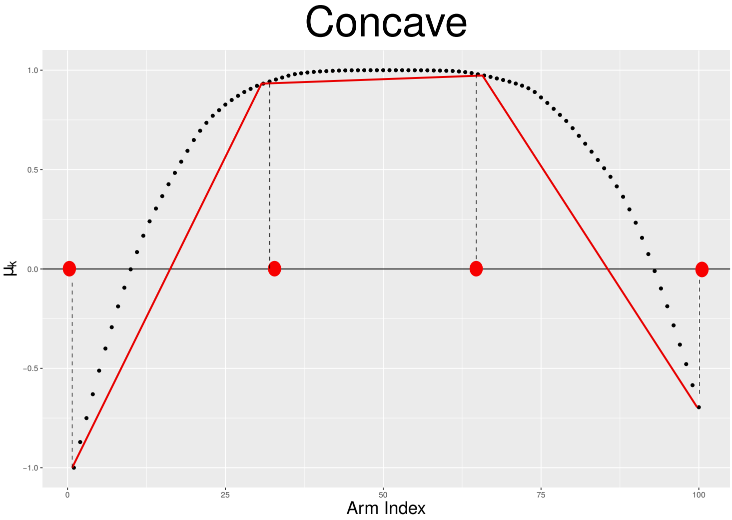
Concave setting - adaptation of the naive binary search



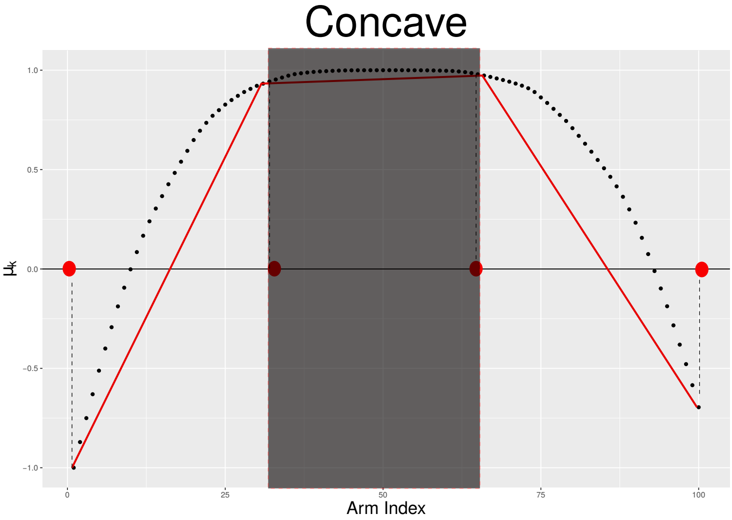
Concave setting - adaptation of the naive binary search



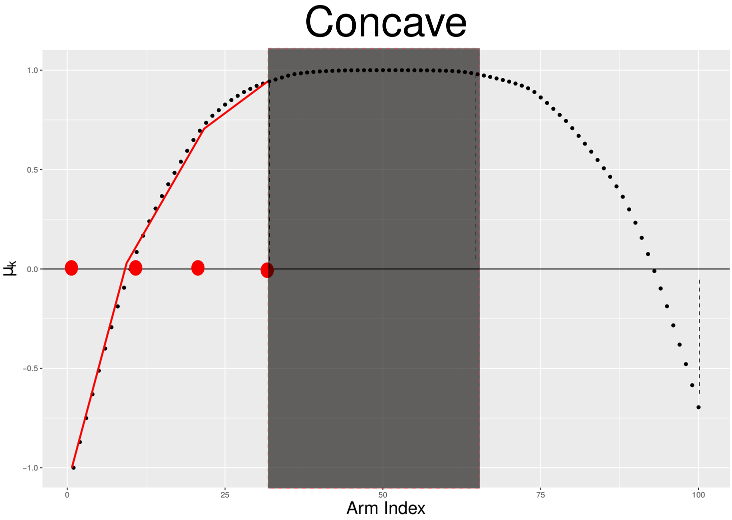
Concave setting - adaptation of the naive binary search



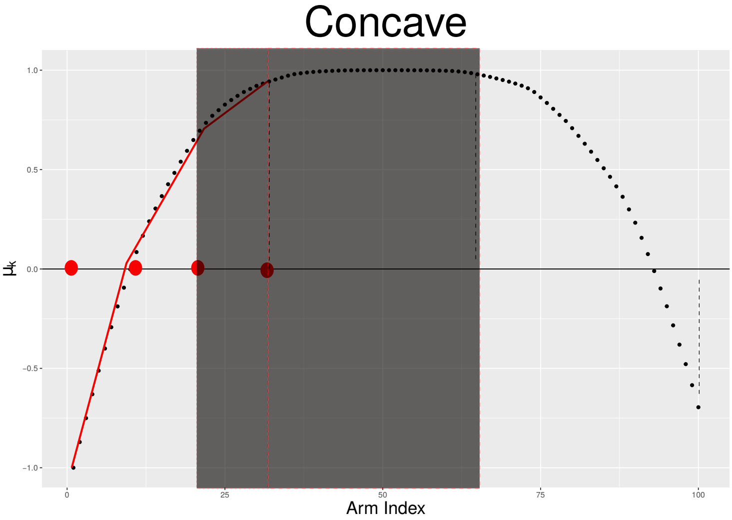
Concave setting - adaptation of the naive binary search



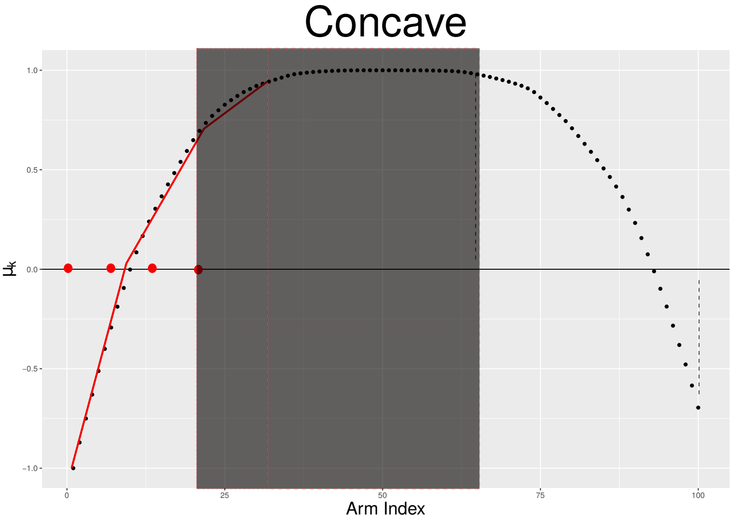
Concave setting - adaptation of the naive binary search



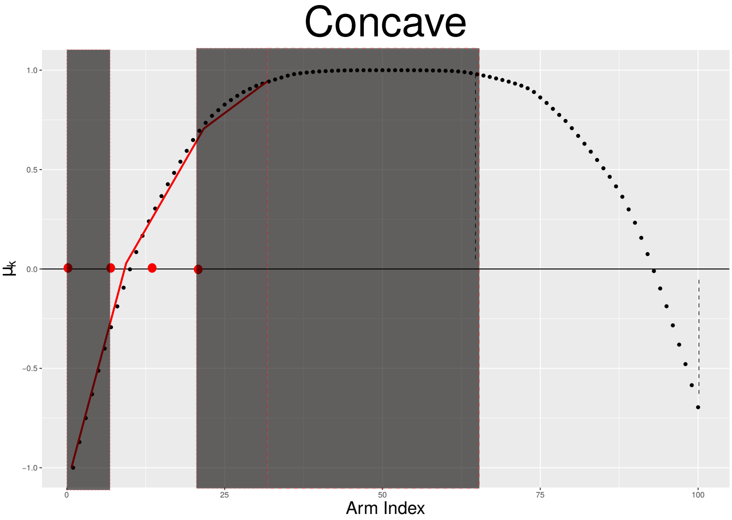
Concave setting - adaptation of the naive binary search



Concave setting - adaptation of the naive binary search



Concave setting - adaptation of the naive binary search



Concave setting - naive binary search

- Monotone setting, naive binary search (sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)\log\left(\log(K)\right)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2 / \log K).$$

- Concave setting, gradient adaptation of naive binary search (also sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)\log\left(\log(K)\right)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2 / \log K).$$

Concave setting - naive binary search

- Monotone setting, naive binary search (sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)\log\left(\log(K)\right)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\square T \min_k \Delta_k^2 / \log K).$$

- Concave setting, gradient adaptation of naive binary search (also sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)\log\left(\log(K)\right)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\square T \min_k \Delta_k^2 / \log K).$$

Concave setting - corrected binary search

- Monotone setting, corrected binary search (sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2).$$

- Concave setting, gradient adaptation of corrected binary search (also sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2).$$

Concave setting - corrected binary search

- Monotone setting, corrected binary search (sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2).$$

- Concave setting, gradient adaptation of corrected binary search (also sampling $\log(K)$ arms)

$$r_T \lesssim \sqrt{\frac{\log(K)}{T}},$$

and for $T \gtrsim \log K$

$$e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2).$$

Concave setting - lower bound

Theorem (Lower bound, concave setting (Cheshire et al, 2020, 2020+))

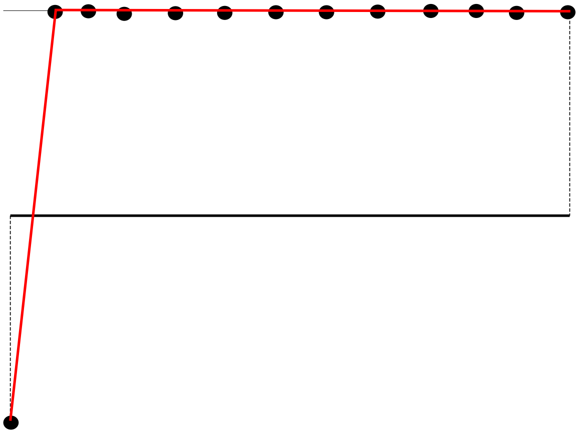
For any algorithm, there exists a concave problem such that:

$$r_T \gtrsim \sqrt{\frac{\log\log(K)}{T}},$$

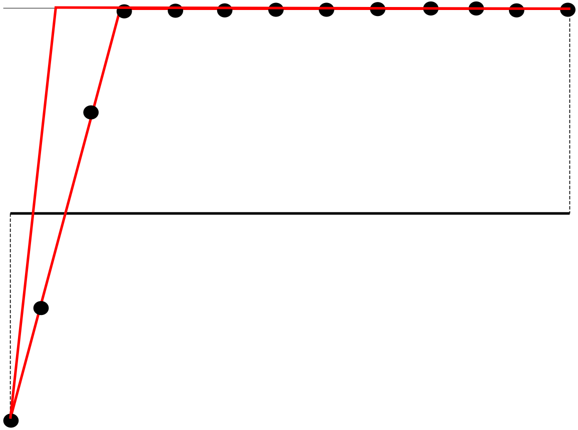
and a concave problem such that $\Delta_i \in [c\bar{\Delta}_i, C\bar{\Delta}_i], \forall i$ *and such that*

$$e_T \gtrsim \exp(-\square T \min_k \bar{\Delta}_k^2).$$

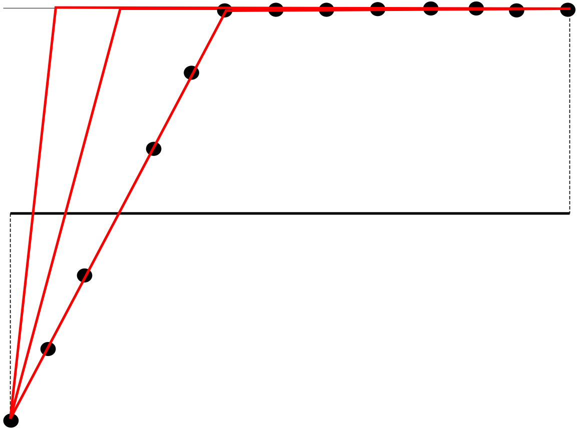
Concave setting - lower bound



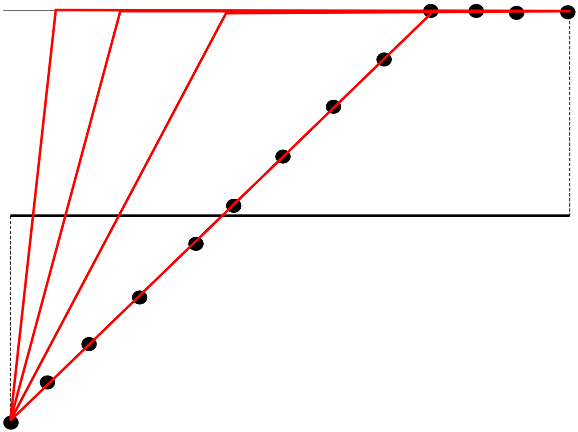
Concave setting - lower bound



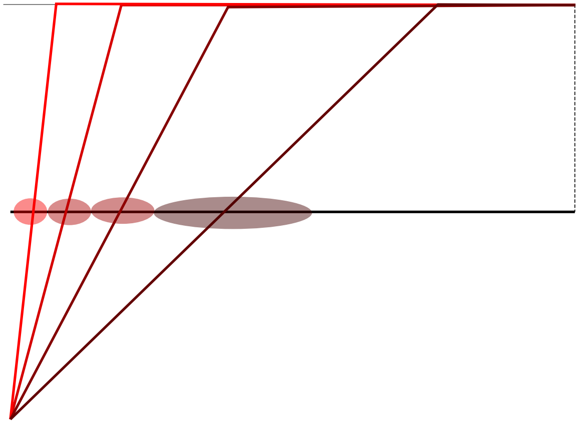
Concave setting - lower bound



Concave setting - lower bound



Concave setting - lower bound



Concave setting - lower bound

Theorem (Lower bound, concave setting (Cheshire et al, 2020, 2020+))

For any algorithm, there exists a concave problem such that:

$$r_T \gtrsim \sqrt{\frac{\log\log(K)}{T}},$$

and a concave problem such that $\Delta_i \in [c\bar{\Delta}_i, C\bar{\Delta}_i], \forall i$ *and such that*

$$e_T \gtrsim \exp(-\Box T \min_k \bar{\Delta}_k^2).$$

Reminder: corrected gradient adaptation of binary search reaches

$$r_T \lesssim \sqrt{\frac{\log(K)}{T}}, \qquad e_T \lesssim \log K \exp(-\Box T \min_k \Delta_k^2).$$

Concave setting - beyond corrected gradient adaptation of binary search

We can do better! **Idea:**

- Use this "log-scale" idea from the lower bound...
- .... to do a corrected binary search on a log-scale....
- .... but so that log-scale approximation is sufficient, we can only reduce distance to threshold by a factor.

Algorithm: perform iterative corrected binary search on log-scale and refine gradually the level set.

Concave setting - beyond corrected gradient adaptation of binary search

We can do better! **Idea:**

- Use this "log-scale" idea from the lower bound...
- .... to do a corrected binary search on a log-scale....
- .... but so that log-scale approximation is sufficient, we can only reduce distance to threshold by a factor.

Algorithm: perform iterative corrected binary search on log-scale and refine gradually the level set.

Concave setting - beyond corrected gradient adaptation of binary search

We can do better! **Idea:**

- Use this "log-scale" idea from the lower bound...
- .... to do a corrected binary search on a log-scale....
- .... but so that log-scale approximation is sufficient, we can only reduce distance to threshold by a factor.

Algorithm: perform iterative corrected binary search on log-scale and refine gradually the level set.

Concave setting - beyond corrected gradient adaptation of binary search

We can do better! **Idea:**

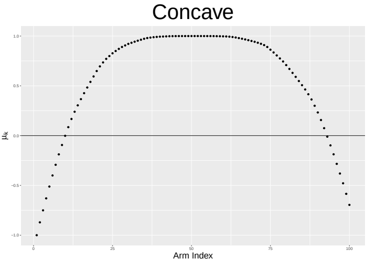
- Use this "log-scale" idea from the lower bound...
- .... to do a corrected binary search on a log-scale....
- .... but so that log-scale approximation is sufficient, we can only reduce distance to threshold by a factor.

Algorithm: perform iterative corrected binary search on log-scale and refine gradually the level set.

Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

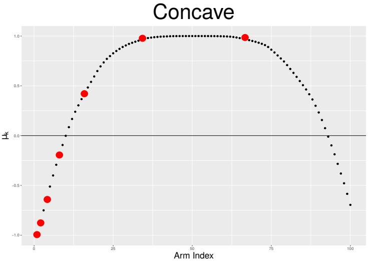
Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \rightarrow$ then refine iteratively



Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

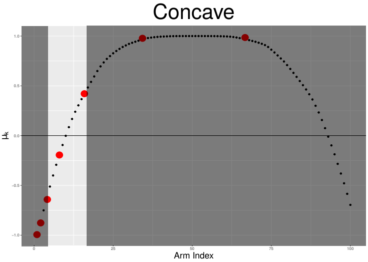
Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \rightarrow$ then refine iteratively



Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

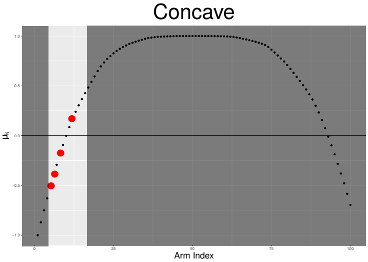
Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \to$ then refine iteratively



Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

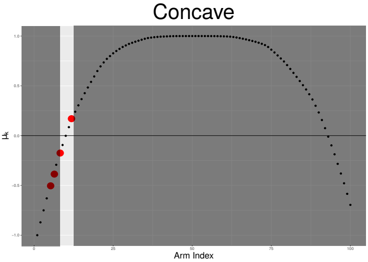
Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \to$ then refine iteratively



Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

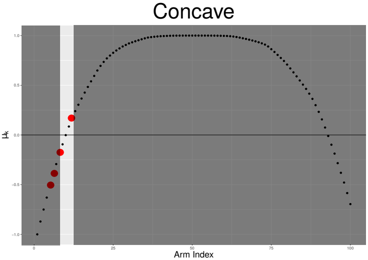
Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \rightarrow$ then refine iteratively



Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \to$ then refine iteratively

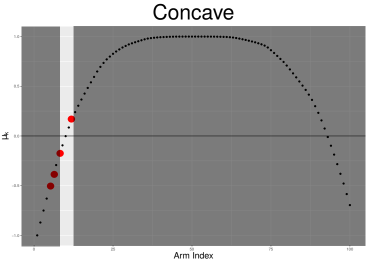


'Rough' level sets (large ϵ_i): minimal amount of budget used, high probability of refining.

Iterative binary search on log-scale

Aim: at iterative step $i \leq M$, obtain a $\epsilon_i = (7/8)^i$ level set above 0

Cover level set at log-scale and do binary search giving budget $\epsilon_i^{-2}(M-i) \to$ then refine iteratively



'Rough' level sets (large ϵ_i): minimal amount of budget used, high probability of refining.

Last level set: M such that $\epsilon_M \asymp \sqrt{\frac{\log\log K}{T}}$ and at most K arms $\to \log K$ on log-scale. **So corrected gradient binary search on $\log\log K$ arms with remaining budget of $T/2$!**

Concave setting - problem independent upper bound

Theorem (Problem independent upper bound (Cheshire et al, 2020))

The strategy with iterative refinement, doing binary search at log-scale with auto-correction and appropriated choice of cut-off satisfies on any problem

$$r_T \lesssim \sqrt{\frac{\log \log K}{T}}.$$

Minimax rates in the concave setting

Theorem (Concave setting (Cheshire et al, 2020))

It holds that

$$\inf_{\text{algo}} \sup_{\text{problem}} r_T \approx \sqrt{\frac{\log\log K}{T}},$$

and for $T \gtrsim \log K$ *and any* $\bar{\Delta}$

$$\inf_{\text{algo}} \sup_{\bar{\Delta}-\text{problem}} \log e_T \asymp -T \min_k \bar{\Delta}_k^2.$$

Remark: the cut-off has to be taken carefully.

Summary until now...

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K \log K}{T}}$	$\sqrt{\frac{\log K}{T}}$	$\sqrt{\frac{\log \log K}{T}}$	
Pr. dep. $\log e_t$	$-T/H$	$-T \min_k \Delta_k^2$	$-T \min_k \Delta_k^2$	

Outline

TBP: setting

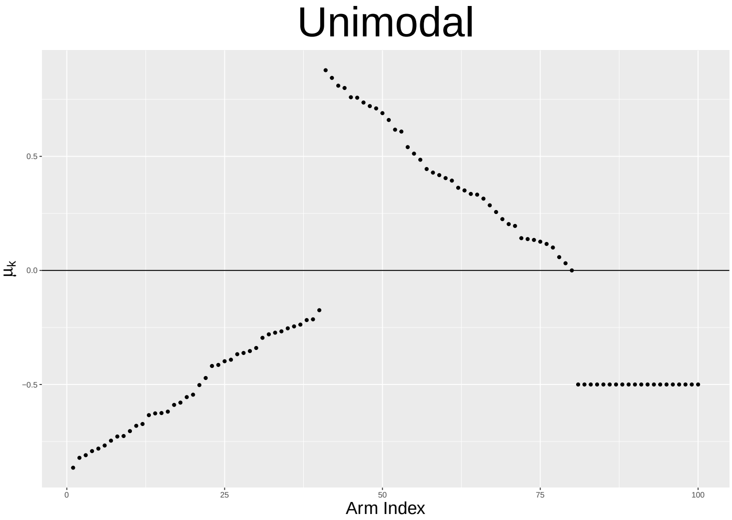
Unconstrained setting

Monotone setting

Concave setting

Unimodal setting

Unimodal setting



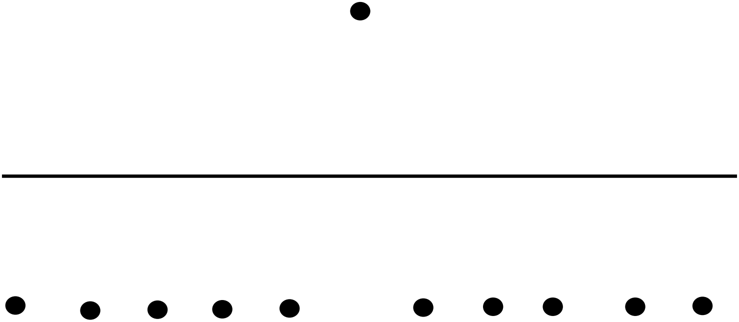
Unimodal?

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K \log K}{T}}$	$\sqrt{\frac{\log K}{T}}$	$\sqrt{\frac{\log \log K}{T}}$	
Pr. dep. $\log e_t$	$-T/H$	$-T \min_k \Delta_k^2$	$-T \min_k \Delta_k^2$	

Unimodal?

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K\log K}{T}}$	$\sqrt{\frac{\log K}{T}}$	$\sqrt{\frac{\log\log K}{T}}$	$\sqrt{\frac{K}{T}}$
Pr. dep. $\log e_t$	$-T/H$	$-T\min_k \Delta_k^2$	$-T\min_k \Delta_k^2$	

Unimodal setting: problem independent



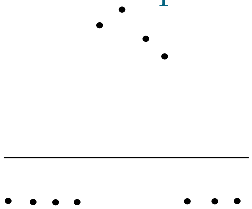
Difficulty in unimodal case: find an arm above threshold...
Interestingly we do not pay the additional $\log K$ here as in unconstrained $\rightarrow$ uniform allocation to find best arm is not the best!

Unimodal?

Results	Unstructured	Monotone	Concave	Unimodal
Pr. indep. r_t	$\sqrt{\frac{K \log K}{T}}$	$\sqrt{\frac{\log K}{T}}$	$\sqrt{\frac{\log \log K}{T}}$	$\sqrt{\frac{K}{T}}$
Pr. dep. $\log e_t$	$-T/H$	$-T \min_k \Delta_k^2$	$-T \min_k \Delta_k^2$	??

Unclear what the right problem dependent class is in this setting....

Unimodal setting: problem dependent



Consider $\bar{\mu} \in \mathbb{R}^{+M}, \bar{\Delta}_{\min}$ and the class $\mathcal{B}_u(\bar{\mu}, \bar{\Delta}_{\min})$ of problems such that

- the bump of M arms above threshold corresponds to $\bar{\mu}$
- all arms below threshold are lower than $-\Delta_{\min}$

In this context:

$$\log e_T \asymp (-\Delta_{\min}^2 T) \vee \left(-TD^2\right),$$

where $D^2 = \frac{M}{K} \frac{1}{M} \sum_{i=1}^{M} \bar{\mu}_i)^2$.

Conclusion

We have:

- characterised in many regimes the minimax rate of TBP under shape constraints.
- (Much) faster than in batch under shape constraints.
- Unlike often in bandits: no conditioning on single arms but on 'successful events'.
- Study of the problem independent continuous case (in (Cheshire et al, 2020)).

What is unclear:

- Right class for unimodal?
- Higher dimension?
- Limiting regimes?
- Concave not in $[0, 1]$, problem independent?

Thank you very much!

Conclusion

We have:

- characterised in many regimes the minimax rate of TBP under shape constraints.
- (Much) faster than in batch under shape constraints.
- Unlike often in bandits: no conditioning on single arms but on 'successful events'.
- Study of the problem independent continuous case (in (Cheshire et al, 2020)).

What is unclear:

- Right class for unimodal?
- Higher dimension?
- Limiting regimes?
- Concave not in $[0, 1]$, problem independent?

Thank you very much!

Conclusion

We have:

- characterised in many regimes the minimax rate of TBP under shape constraints.
- (Much) faster than in batch under shape constraints.
- Unlike often in bandits: no conditioning on single arms but on ‘successful events’.
- Study of the problem independent continuous case (in (Cheshire et al, 2020)).

What is unclear:

- Right class for unimodal?
- Higher dimension?
- Limiting regimes?
- Concave not in $[0, 1]$, problem independent?

Thank you very much!