# Bridging stochastic and adversarial bandits

*Thodoris Lykouris*

MIT MANAGEMENT SLOAN SCHOOL

https://www.regendus.com/best-random-number-generator-apps/
https://upload.wikimedia.org/wikipedia/commons/thumb/e/e1/Gaoliang_Bridge.JPG/1200px-Gaoliang_Bridge.JPG
https://physicsworld.com/wp-content/uploads/2022/04/Crease-demon-or-devil-110425232-Shutterstock_ChromaCo.jpg

# Multi-armed bandits

For $t = 1 \dots T$:

1. Learner selects a distribution $p(t)$ across arms
2. Each arm $a$ gets a reward $r_a(t)$
3. Learner (randomly) selects arm $A(t) \sim p(t)$
4. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
5. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

# Multi-armed bandits

<u>For $t = 1 \dots T$:</u>

1. Learner selects a distribution $p(t)$ across arms
2. Each arm $a$ gets a reward $r_a(t)$
3. Learner (randomly) selects arm $A(t) \sim p(t)$
4. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
5. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

**Stochastic bandits**

*i.i.d. rewards for each arm*

$r_a(t) \sim F_a$

# Multi-armed bandits

<u>For $t = 1 \dots T$:</u>

1. Learner selects a distribution $p(t)$ across arms
2. Each arm $a$ gets a reward $r_a(t)$
3. Learner (randomly) selects arm $A(t) \sim p(t)$
4. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
5. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

**Stochastic bandits**

*i.i.d. rewards for each arm*

$r_a(t) \sim F_a$

**Adversarial bandits**

*rewards function of entire history*

$r_a(t) \sim F_a(H_{1\dots t-1})$

# Multi-armed bandits

For $t = 1 \dots T$:

1. Learner selects a distribution $p(t)$ across arms
2. Each arm $a$ gets a reward $r_a(t)$
3. Learner (randomly) selects arm $A(t) \sim p(t)$
4. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
5. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

**This talk**

**Stochastic bandits**

*i.i.d. rewards for each arm*

$r_a(t) \sim F_a$

**Adversarial bandits**

*rewards function of entire history*

$r_a(t) \sim F_a(H_{1 \dots t-1})$

# Main questions

*Q1 (Best of both worlds)*

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

# Main questions

*Q1 (Best of both worlds)*

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

*Q2 (Bridging the two worlds)*

*What are models that interpolate between the two worlds? What are design principles that adapt to the difficulty of such stochastic-adversarial models?*

# Main questions

*Q1 (Best of both worlds)*

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

*Q2 (Bridging the two worlds)*

*What are models that interpolate between the two worlds? What are design principles that adapt to the difficulty of such stochastic-adversarial models?*

*Q3 (Beyond multi-armed bandits)*

*How do these design principles extend beyond multi-armed bandits to more complex reward and feedback structures?*

# Performance metrics

$$Regret = \max_{a^\star} \sum_t r_{a^\star}(t) - \sum_t r_{A(t)}(t)$$

*compares to hindsight-optimal arm $a^\star$*

- *depends on the realized rewards*
- *also depends on the algorithm in adversarial bandits*

# Performance metrics

$$Regret = \max_{a^\star} \sum_t r_{a^\star}(t) - \sum_t r_{A(t)}(t)$$

*compares to hindsight-optimal arm $a^\star$*

- *depends on the realized rewards*
- *also depends on the algorithm in adversarial bandits*

$$PseudoRegret = \max_{a^\star} \mathrm{E}\left[\sum_t r_{a^\star}(t)\right] - E\left[\sum_t r_{A(t)}(t)\right]$$

*compares to ex-ante optimal arm $a^\star$*

- *highest mean in stochastic bandits (only function of reward distributions)*
- *still depends on algorithm but not on realizations in adversarial bandits*

# The two worlds

Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

# The two worlds

Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- Example: Online advertising

  $K$ *arms => ads,* $F(a)$ *=> click propensity,*

  *mean* $\mu(a)$ *=> click-through-rate*

# The two worlds

Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- Example: Online advertising

  $K$ *arms => ads,* $F(a)$ *=> click propensity,*

  *mean* $\mu(a)$ *=> click-through-rate*

- Algorithms

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

# The two worlds

## Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- Example: Online advertising

  $K$ *arms => ads,* $F(a)$ *=> click propensity,*

  *mean* $\mu(a)$ *=> click-through-rate*

- Algorithms

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

- Performance guarantee: $\Delta(a) = \max_{a^*} \mu(a^*) - \mu(a)$

  $Pseudoregret \approx \sum_a \min\left(\frac{\log T}{\Delta(a)}, \Delta(a)\, T\right)$

  $Regret \approx \sum_a \min\left(\frac{\log(KT/\delta)}{\Delta(a)}, \sqrt{T}\right)$ *with prob.* $\geq 1 - \delta$

# The two worlds

**Stochastic bandits**

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

**Adversarial bandits**

*function of entire history:* $r_a(t) \sim F_a(H_{1\ldots t-1})$

- Example: Online advertising

  $K$ *arms => ads,* $F(a)$ *=> click propensity,*

  *mean* $\mu(a)$ *=> click-through-rate*

- Algorithms

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

- Performance guarantee: $\Delta(a) = \max_{a^*} \mu(a^*) - \mu(a)$

  $$Pseudoregret \approx \sum_a \min\left(\frac{\log T}{\Delta(a)}, \Delta(a)\, T\right)$$

  $$Regret \approx \sum_a \min\left(\frac{\log(KT/\delta)}{\Delta(a)}, \sqrt{T}\right)$$ *with prob.* $\geq 1 - \delta$

# The two worlds

## Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- Example: Online advertising

  *$K$ arms => ads, $F(a)$ => click propensity,*

  *mean $\mu(a)$ => click-through-rate*

- Algorithms

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

- Performance guarantee: $\Delta(a) = \max_{a^*} \mu(a^*) - \mu(a)$

  $Pseudoregret \approx \sum_a \min\left(\frac{\log T}{\Delta(a)}, \Delta(a)\,T\right)$

  $Regret \approx \sum_a \min\left(\frac{\log(KT/\delta)}{\Delta(a)}, \sqrt{T}\right)$ *with prob.* $\geq 1-\delta$

## Adversarial bandits

*function of entire history:* $r_a(t) \sim F_a(H_{1\ldots t-1})$

- Example: Learning in games

  *arms => bidding strategies,*

  *other agents makes rewards non-stochastic*

# The two worlds

## Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- Example: Online advertising

  *$K$ arms => ads, $F(a)$ => click propensity,*

  *mean $\mu(a)$ => click-through-rate*

- Algorithms

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

- Performance guarantee: $\Delta(a) = \max_{a^*} \mu(a^*) - \mu(a)$

  $$Pseudoregret \approx \sum_a \min\left(\frac{\log T}{\Delta(a)}, \Delta(a)\, T\right)$$

  $$Regret \approx \sum_a \min\left(\frac{\log(KT/\delta)}{\Delta(a)}, \sqrt{T}\right) \quad \textit{with prob.} \geq 1-\delta$$

## Adversarial bandits

*function of entire history:* $r_a(t) \sim F_a(H_{1\ldots t-1})$

- Example: Learning in games

  *arms => bidding strategies,*

  *other agents makes rewards non-stochastic*

- Algorithms

  *EXP3.P* [Auer, Cesa-Bianchi, Freund, Schapire, SICOMP '02]

  Tsallis-INF [Audibert & Bubeck, JMLR'10] [Abernethy, Lee, Tewari, NeurIPS'15]

  *Log-barrier* [Foster, Li, ***L***, Sridharan, Tardos, NeurIPS'16]

# The two worlds

## Stochastic bandits

*i.i.d. rewards for each arm:* $r_a(t) \sim F(a)$

- <u>Example: Online advertising</u>

  *$K$ arms => ads, $F(a)$ => click propensity,*

  *mean $\mu(a)$ => click-through-rate*

- <u>Algorithms</u>

  *UCB.* [Auer, Cesa-Bianchi, Fischer, Machine Learning '02]

  *Successive Elimination* [Even-Dar, Mannor, Mansour, JMLR'06]

  *Thompson Sampling* [Agrawal & Goyal, JACM'17]

- <u>Performance guarantee:</u> $\Delta(a) = \max_{a^*} \mu(a^*) - \mu(a)$

  $Pseudoregret \approx \sum_a \min\left(\frac{\log T}{\Delta(a)}, \Delta(a)\, T\right)$

  $Regret \approx \sum_a \min\left(\frac{\log(KT/\delta)}{\Delta(a)}, \sqrt{T}\right)$ *with prob.* $\geq 1-\delta$

## Adversarial bandits

*function of entire history:* $r_a(t) \sim F_a(H_{1\ldots t-1})$

- <u>Example: Learning in games</u>

  *arms => bidding strategies,*

  *other agents makes rewards non-stochastic*

- <u>Algorithms</u>

  *EXP3.P* [Auer, Cesa-Bianchi, Freund, Schapire, SICOMP '02]

  Tsallis-INF [Audibert & Bubeck, JMLR'10]

  [Abernethy, Lee, Tewari, NeurIPS'15]

  *Log-barrier* [Foster, Li, ***L***, Sridharan, Tardos, NeurIPS'16]

- <u>Performance guarantee:</u>

  $Pseudoregret \approx \sqrt{KT}$

  $Regret \approx \sqrt{KT\, log(KT/\delta)}$ *with prob.* $\geq 1-\delta$

# Best of both worlds

*<u>Q1 (Best of both worlds)</u>* [Bubeck & Slivkins, COLT'12]

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

# Best of both worlds

*Q1 (Best of both worlds)* [Bubeck & Slivkins, COLT'12]

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

**Stochastic-based approach**

1. *Run stochastic bandit algorithm*
2. *Test if stochasticity holds*
3. *If test fails, switch to adversarial bandits*

# Best of both worlds

*Q1 (Best of both worlds)* [Bubeck & Slivkins, COLT'12]

*How can we simultaneously obtain the stochastic guarantee for stochastic environment and the adversarial guarantee for adversarial environment?*

**Stochastic-based approach**

1. *Run stochastic bandit algorithm*
2. *Test if stochasticity holds*
3. *If test fails, switch to adversarial bandits*

**Adversarial-based approach**

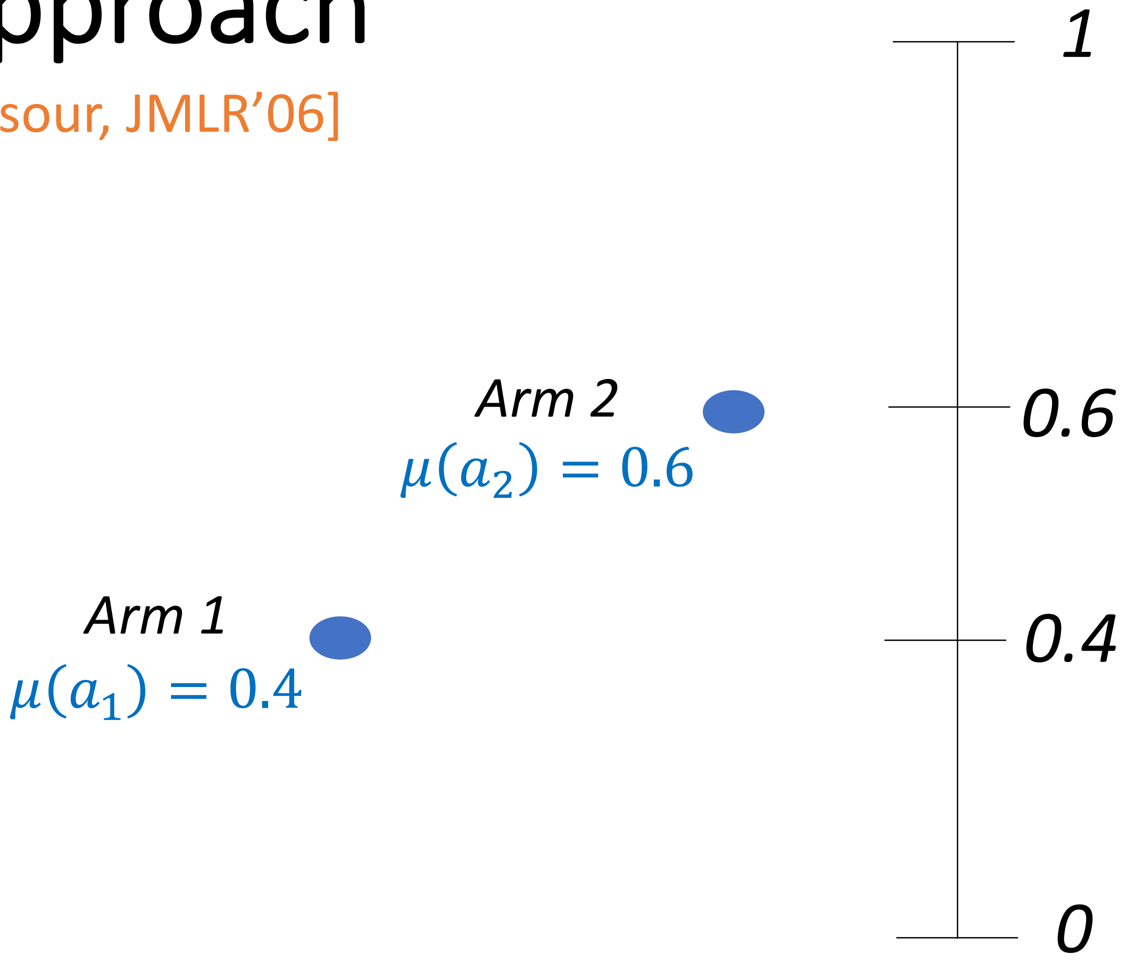
1. *Run adversarial bandit algorithm*
2. *Exploration adapts to empirical gap*

# Stochastic approach

## Successive Elimination

[Even-Dar, Mannor, Mansour, JMLR'06]

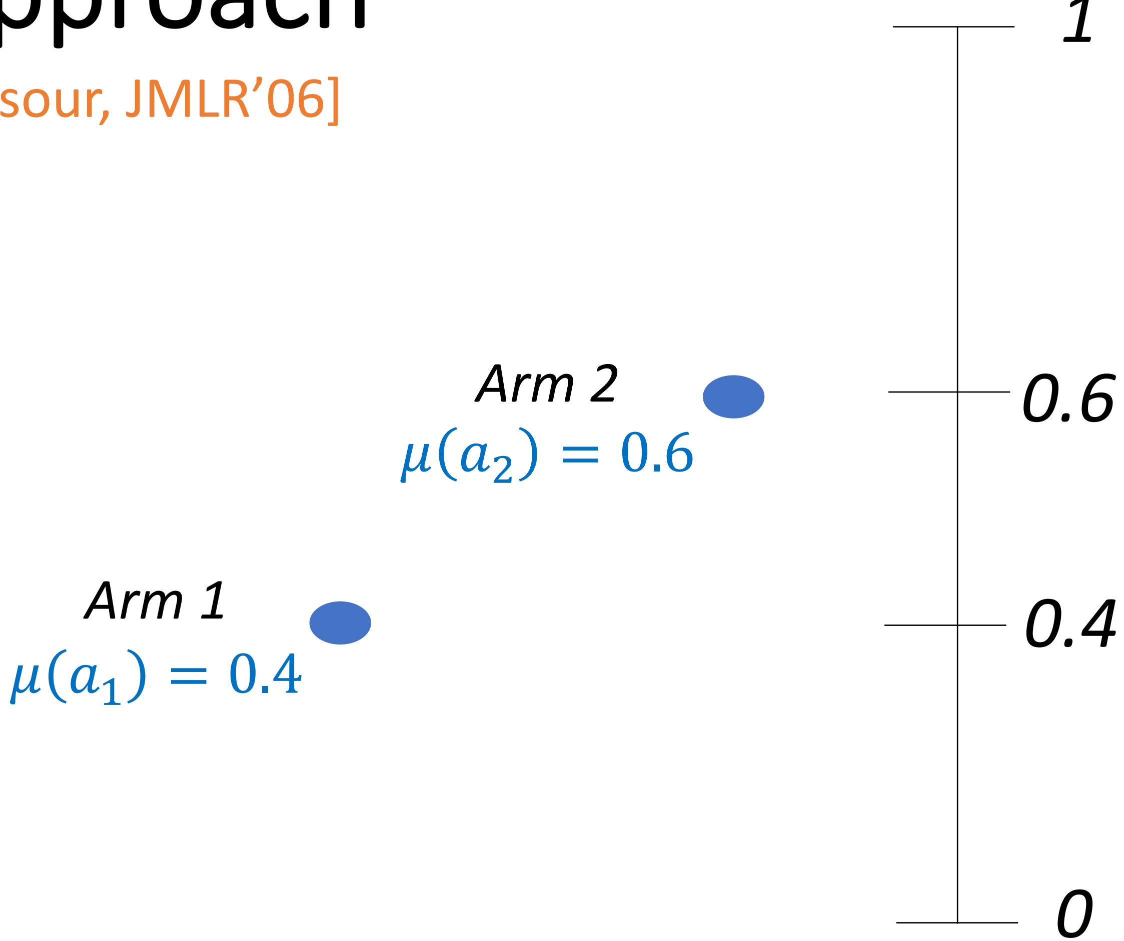
- Each arm has a mean $\mu(a)$



# Stochastic approach

Successive Elimination  [Even-Dar, Mannor, Mansour, JMLR’06]

- Each arm has a mean $\mu(a)$
- Keep a set of “active” arms (initially all)

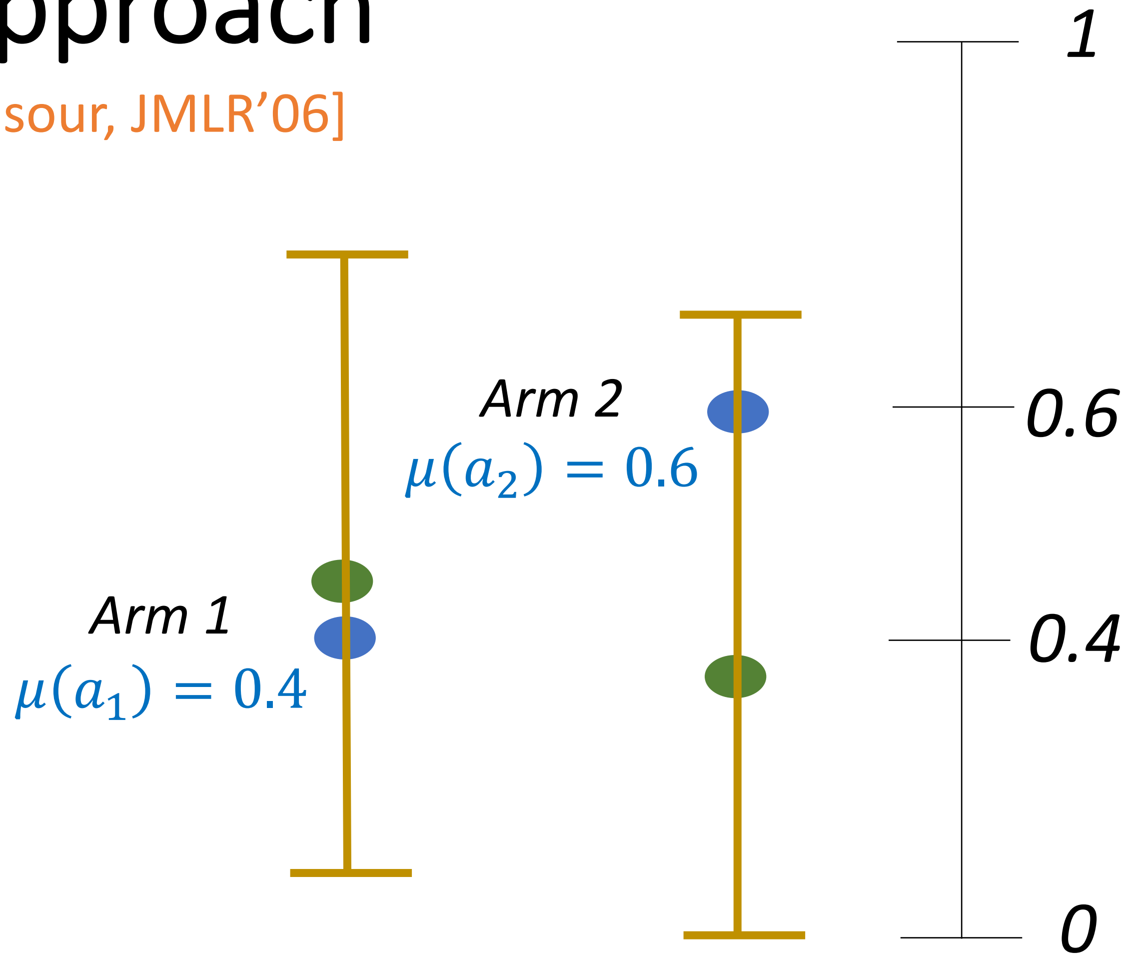


# Stochastic approach

## Successive Elimination

[Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

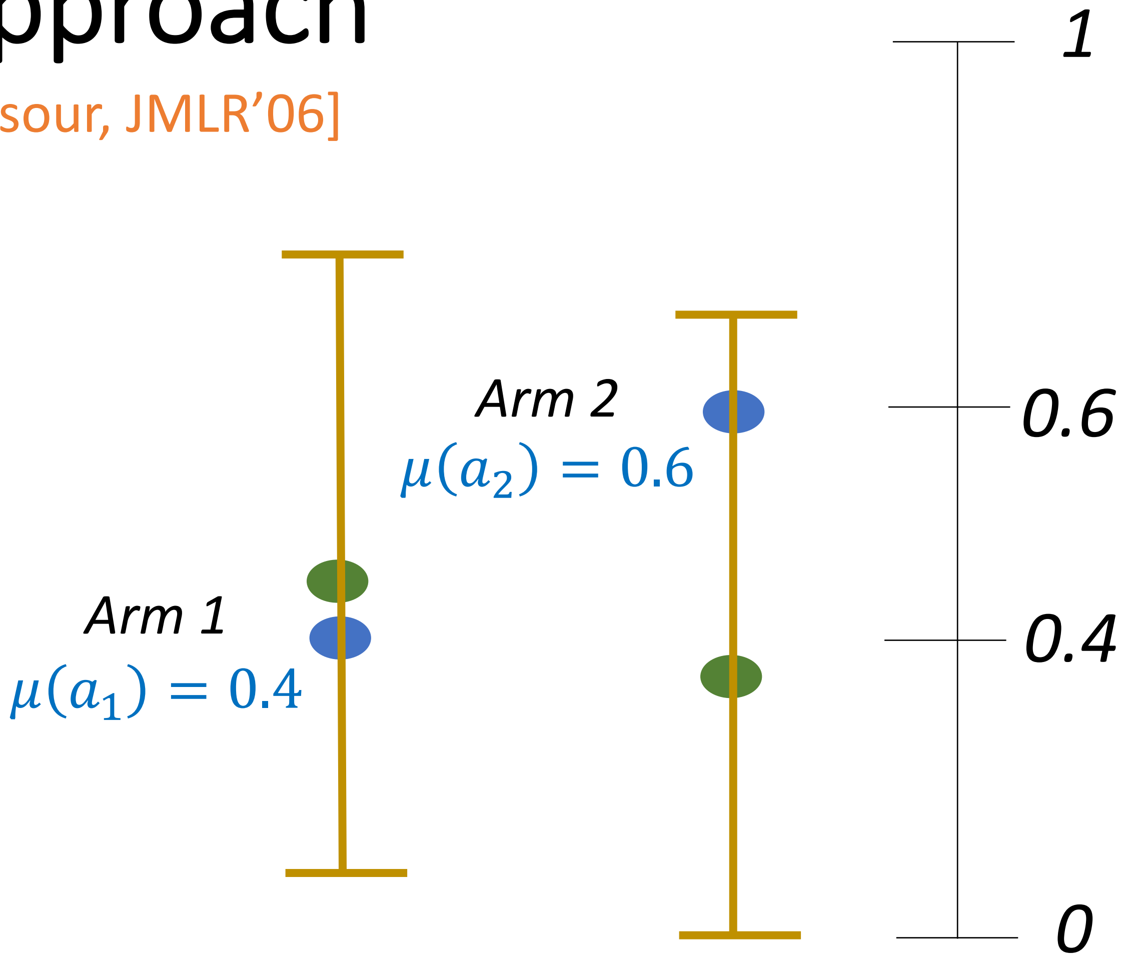


# Stochastic approach

<u>Successive Elimination</u> [Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another



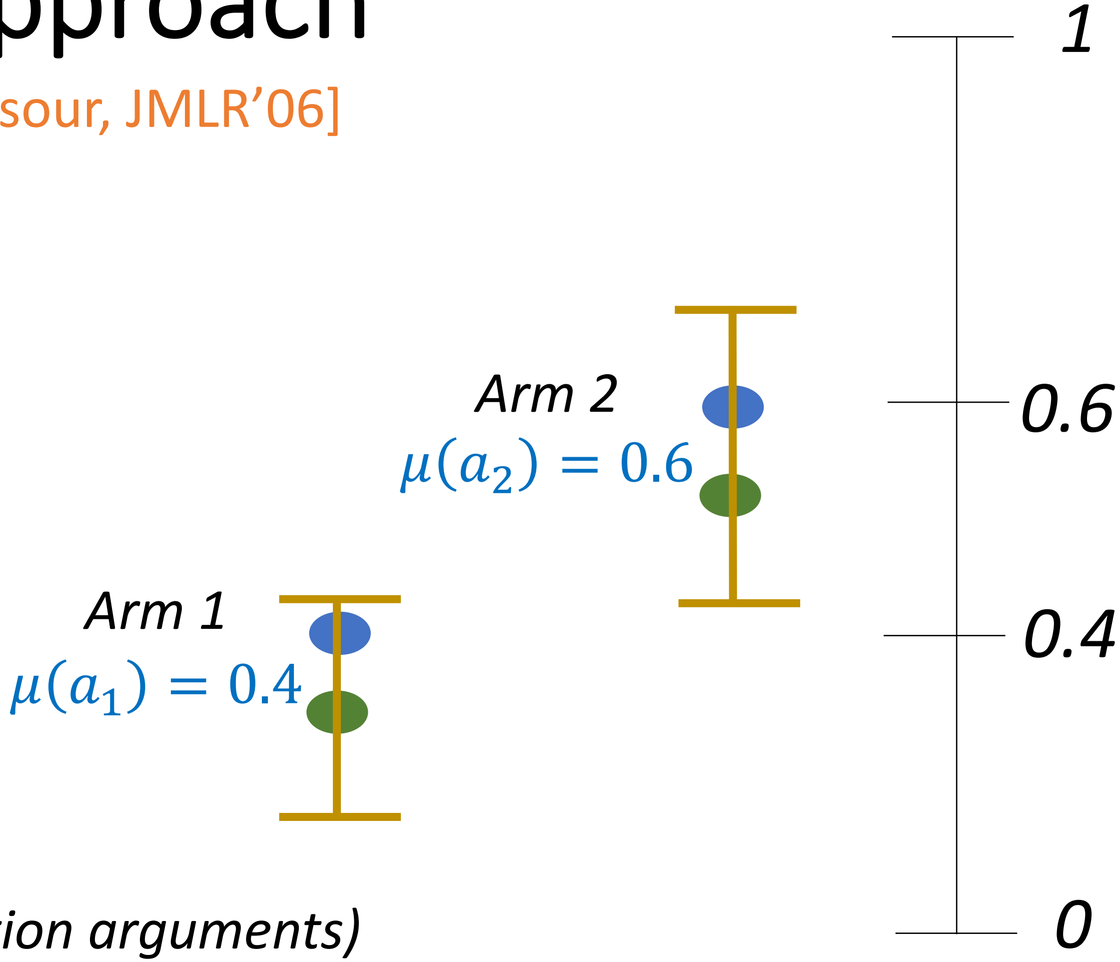
# Stochastic approach

## Successive Elimination

[Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another




## Crux of analysis

- *W.h.p. actual mean in confidence interval (concentration arguments)*
- *Subopimal arm* $a$ *is deactivated after* $\frac{\log(KT/\delta)}{(\Delta_a)^2}$ *rounds w.h.p.*
  - *Contributes* $\frac{\log(KT/\delta)}{(\Delta_a)^2} \cdot \Delta_a = \frac{\log(KT/\delta)}{\Delta_a}$ *to regret*

# Stochastic-based best of both worlds

Stochastic and Adversarial Optimal (SAO) algorithm [Bubeck & Slivkins, COLT'12]

- Run Successive Elimination
- For deactivated arms, randomly test if rewards are consistent with confidence interval
- If not: switch to EXP3.P
- Guarantee: Stochastic pseudoregret of $\tilde{O}\left(\frac{K \cdot \log^2(T)}{\Delta}\right)$ and adversarial regret of $\tilde{O}(KT)$

# Stochastic-based best of both worlds

Stochastic and Adversarial Optimal (SAO) algorithm [Bubeck & Slivkins, COLT'12]

- Run Successive Elimination
- For deactivated arms, randomly test if rewards are consistent with confidence interval
  - *Tests should not be very frequent (to maintain logarithmic guarantee)*
- If not: switch to EXP3.P
  - *Tests should not be very infrequent (to have at most $\sqrt{T}$ regret at time of switch)*
- Guarantee: Stochastic pseudoregret of $\tilde{O}\left(\frac{K \cdot \log^2(T)}{\Delta}\right)$ and adversarial regret of $\tilde{O}\left(\sqrt{KT}\right)$

# Stochastic-based best of both worlds

## Stochastic and Adversarial Optimal (SAO) algorithm

[Bubeck & Slivkins, COLT'12]

- Run Successive Elimination
- For deactivated arms, randomly test if rewards are consistent with confidence interval
  - *Tests should not be very frequent (to maintain logarithmic guarantee)*
- If not: switch to EXP3.P
  - *Tests should not be very infrequent (to have at most $\sqrt{T}$ regret at time of switch)*
- Guarantee: Stochastic pseudoregret of $\tilde{O}\left(\frac{K \cdot \log^2(T)}{\Delta}\right)$ and adversarial regret of $\tilde{O}\left(\sqrt{KT}\right)$

## Stochastic and Adversarial PseudoOptimal (SAPO) algorithm

[Auer & Chiang, COLT'16]

- No algorithm can have $\mathrm{o}\left(\log^2(T)\right)$ stochastic pseudoregret and $\mathrm{o}(T)$ adversarial regret w.h.p.

# Stochastic-based best of both worlds

## Stochastic and Adversarial Optimal (SAO) algorithm

[Bubeck & Slivkins, COLT'12]

- Run Successive Elimination
- For deactivated arms, randomly test if rewards are consistent with confidence interval
  - *Tests should not be very frequent (to maintain logarithmic guarantee)*
- If not: switch to EXP3.P
  - *Tests should not be very infrequent (to have at most $\sqrt{T}$ regret at time of switch)*
- Guarantee: Stochastic pseudoregret of $\tilde{O}\left(\frac{K \cdot \log^2(T)}{\Delta}\right)$ and adversarial regret of $\tilde{O}\left(\sqrt{KT}\right)$

## Stochastic and Adversarial PseudoOptimal (SAPO) algorithm

[Auer & Chiang, COLT'16]

- No algorithm can have $o(\log^2(T))$ stochastic pseudoregret and $o(T)$ adversarial regret w.h.p.
- Guarantee: Stochastic pseudoregret of $\tilde{O}\left(\frac{K \cdot \log T}{\Delta}\right)$ and adversarial pseudoregret of $\tilde{O}\left(\sqrt{KT}\right)$
  - *Key idea: use past negative pseudoregret to allow for more infrequent tests*

# Adversarial-based best of both worlds

<u>EXP3++</u> [Seldin & Slivkins, COLT'14] [Seldin & Lugosi, COLT'17]

- Original version of EXP3 mixes with a uniform distribution $\gamma$
- Run EXP3 with arm-specific exploration probabilities $\gamma(a)$ that are inverse to empirical gap
- Leads to near-optimal stochastic and adversarial pseudoregret guarantees

# Adversarial-based best of both worlds

EXP3++ [Seldin & Slivkins, COLT'14] [Seldin & Lugosi, COLT'17]

- Original version of EXP3 mixes with a uniform distribution $\gamma$
- Run EXP3 with arm-specific exploration probabilities $\gamma(a)$ that are inverse to empirical gap
- Leads to near-optimal stochastic and adversarial pseudoregret guarantees

MD beyond Shannon entropy [Wei & Luo, COLT'18] [Zimmert & Seldin, JMLR'21]

- Run Mirror Descent with a stronger regularizer (log-barrier / Tsallis)
  - *No direct gap-driven exploration but probabilities of suboptimal arms decrease starkly*
- Analysis upper bounds regret via a unified "self-bounding term"
- Optimal stochastic and adversarial pseudoregret guarantees

# Adversarial-based best of both worlds

**EXP3++** [Seldin & Slivkins, COLT'14] [Seldin & Lugosi, COLT'17]

- Original version of EXP3 mixes with a uniform distribution $\gamma$
- Run EXP3 with arm-specific exploration probabilities $\gamma(a)$ that are inverse to empirical gap
- Leads to near-optimal stochastic and adversarial pseudoregret guarantees

**MD beyond Shannon entropy** [Wei & Luo, COLT'18] [Zimmert & Seldin, JMLR'21]

- Run Mirror Descent with a stronger regularizer (log-barrier / Tsallis)
  - *No direct gap-driven exploration but probabilities of suboptimal arms decrease starkly*
- Analysis upper bounds regret via a unified "self-bounding term"
- Optimal stochastic and adversarial pseudoregret guarantees

***Julian Zimmert will present this result in the September workshop***

# Hybrid stochastic-adversarial models

Challenges with most best of both worlds approaches:

- Stochastic-based approaches switch to EXP3.P if they detect non-stochasticity
- Until recently, adversarial-based approaches analyzed stochastic and adversarial separately
- In more complex learning settings, there is often no "adversarial" bandit algorithm

# Hybrid stochastic-adversarial models

Challenges with most best of both worlds approaches:

- Stochastic-based approaches switch to EXP3.P if they detect non-stochasticity
- Until recently, adversarial-based approaches analyzed stochastic and adversarial separately
- In more complex learning settings, there is often no "adversarial" bandit algorithm

*Q2 (Bridging the two worlds)*

*What are models that interpolate between the two worlds? What are design principles that adapt to the difficulty of such stochastic-adversarial models?*

*Q3 (Beyond multi-armed bandits)*

*How do these design principles extend beyond multi-armed bandits to more complex reward and feedback structures?*

# Stochastic bandits w/ adversarial corruptions

[*L*, Mirrokni, Paes Leme, STOC'18]

*Most of the data are i.i.d. but some rounds are adversarially corrupted*

Examples

- *Click fraud* in online advertising
- *Fake reviews* in recommender systems

# Model

<u>For $t = 1 \ldots T$:</u>

1. Learner selects a distribution $p(t)$ across arms
2. 
3. Each arm $a$ gets a reward $r_a(t)$

4. Learner (randomly) selects arm $A(t) \sim p(t)$
5. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
6. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

# Model

<u>For $t = 1 \dots T$:</u>

[*L*, Mirrokni, Paes Leme, STOC'18]

1. Learner selects a distribution $p(t)$ across arms
2. Adversary selects *latent* corruption $c(t) \in \{0,1\}$ as function of history $H_{1\dots t-1}$
3. Each arm $a$ gets a reward $r_a(t)$
   - *If* $c^t = 0$, $r_a(t) \coloneqq \tilde{r}_a(t) \sim F_a$ *else* $r_a(t) \coloneqq \bar{r}_a(t) \sim F_a(H_{1\dots t-1})$
4. Learner (randomly) selects arm $A(t) \sim p(t)$
5. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
6. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

# Model

For $t = 1 \ldots T$:

[*L*, Mirrokni, Paes Leme, STOC'18]

1. Learner selects a distribution $p(t)$ across arms
2. Adversary selects *latent* corruption $c(t) \in \{0,1\}$ as function of history $H_{1\ldots t-1}$
3. Each arm $a$ gets a reward $r_a(t)$
   - *If* $c^t = 0, r_a(t) \coloneqq \tilde{r}_a(t) \sim F_a$ *else* $r_a(t) \coloneqq \bar{r}_a(t) \sim F_a(H_{1\ldots t-1})$
4. Learner (randomly) selects arm $A(t) \sim p(t)$
5. **Reward earning:** Learner earns reward $r_{A(t)}(t)$
6. **Bandit feedback:** Learner observes reward $r_{A(t)}(t)$

***Goal:* Algorithm design principles that adapt to the number of corrupted rounds $C = \sum_t c(t)$**

Unknown number of corrupted rounds: $C = \sum_t c^t$

Number of arms: $K$

# Three main techniques

**Multi-layering Successive Elimination Race** [*L*, Mirrokni, Paes Leme, STOC'18]

With high probability: $$Regret \leq \sum_a \frac{\log^2(T) + CK \cdot \log(KT/\delta)}{\Delta(a)}$$

**BARBAR: Bad Arms get Recource** [Gupta, Koren, Talwar, COLT'19]

With high probability: $$Regret \leq CK + \sum_a \frac{\log^2(KT/\delta)}{\Delta(a)}$$

**Mirror Descent with Tsallis-INF** [Zimmert & Seldin, JMLR'21]

$$Pseudoregret \leq \sum_a \frac{\log(T)}{\Delta(a)} + \sqrt{C \sum_a \frac{\log(T)}{\Delta(a)}}$$

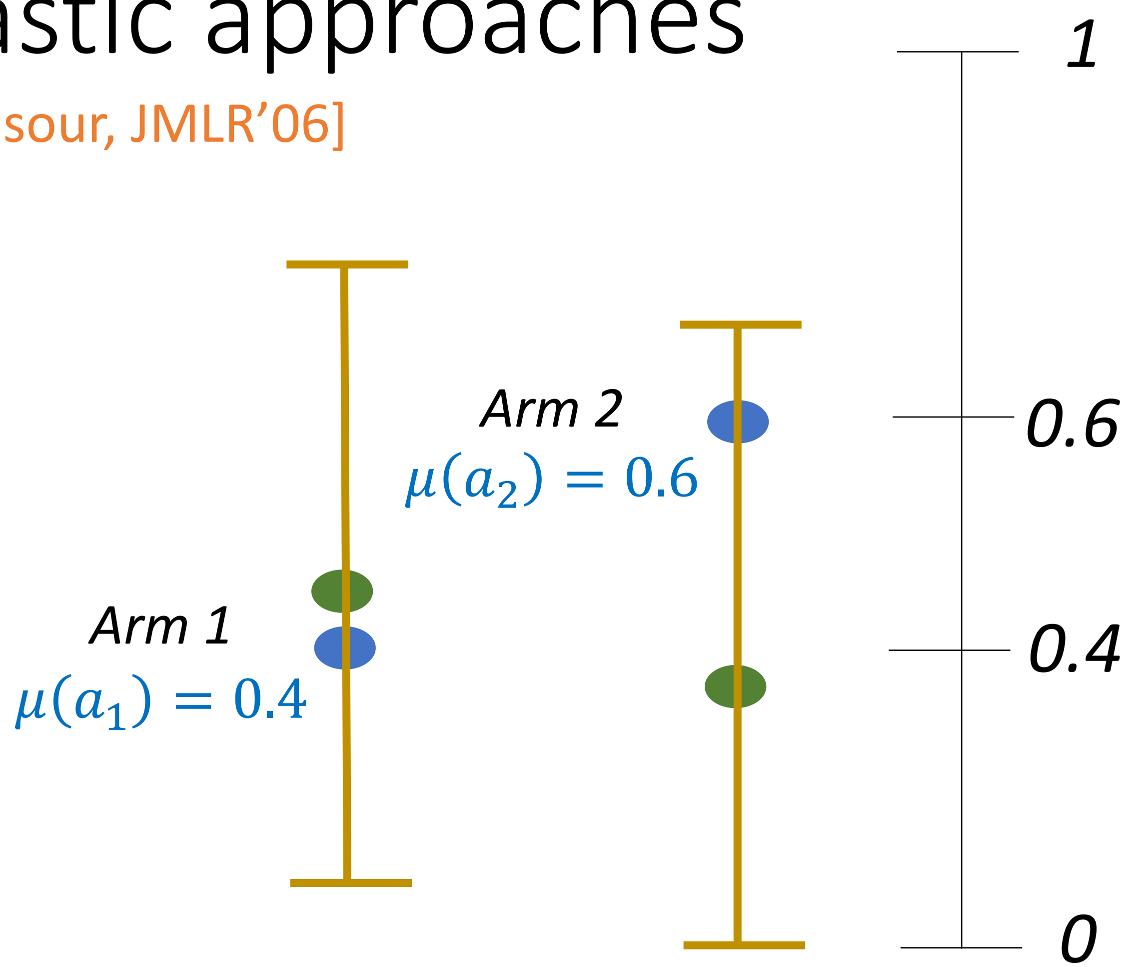
- *assumes uniqueness of optimal arm*

# Brittleness of stochastic approaches

<u>Successive Elimination</u> [Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another

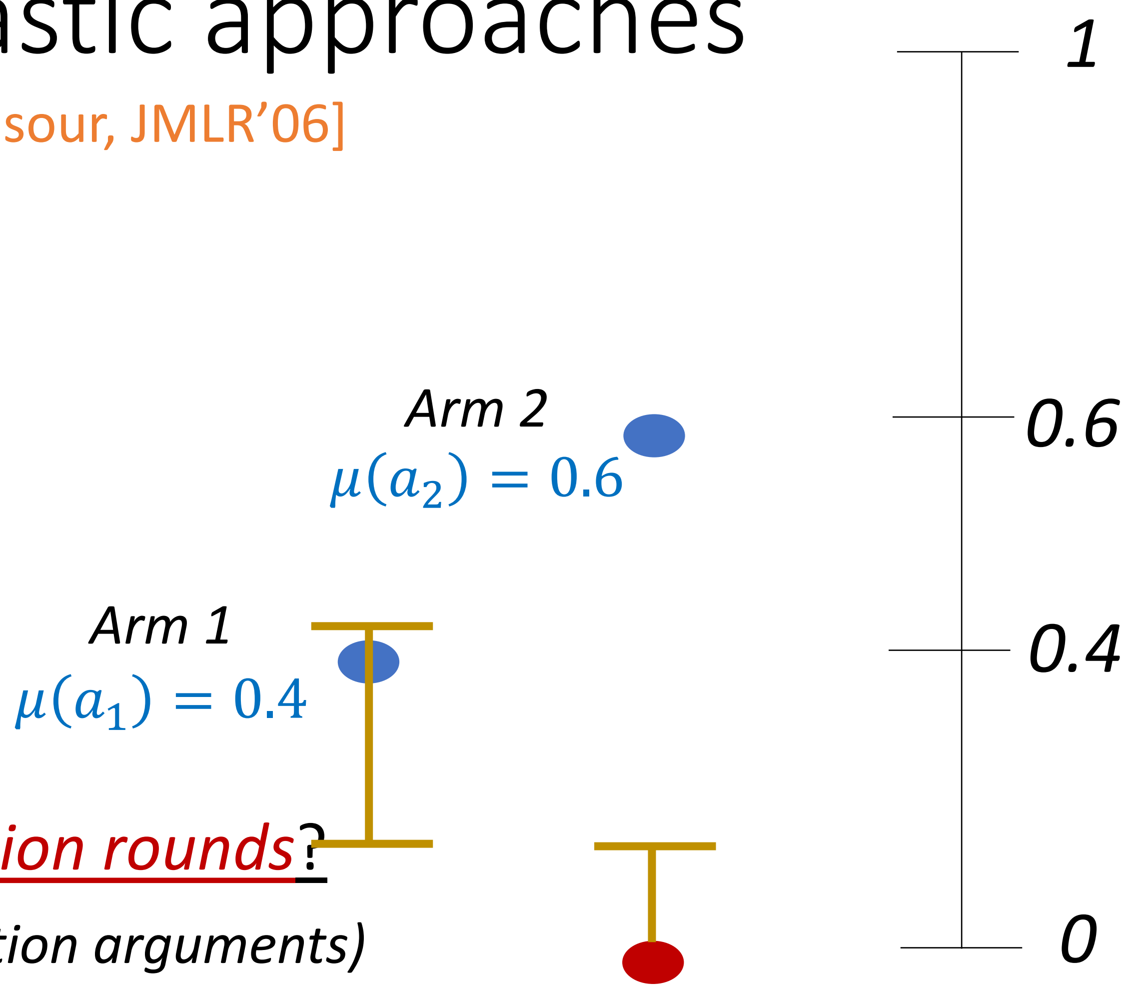


# Brittleness of stochastic approaches

**Successive Elimination** [Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another




**What breaks *if adversary corrupts the exploration rounds*?**

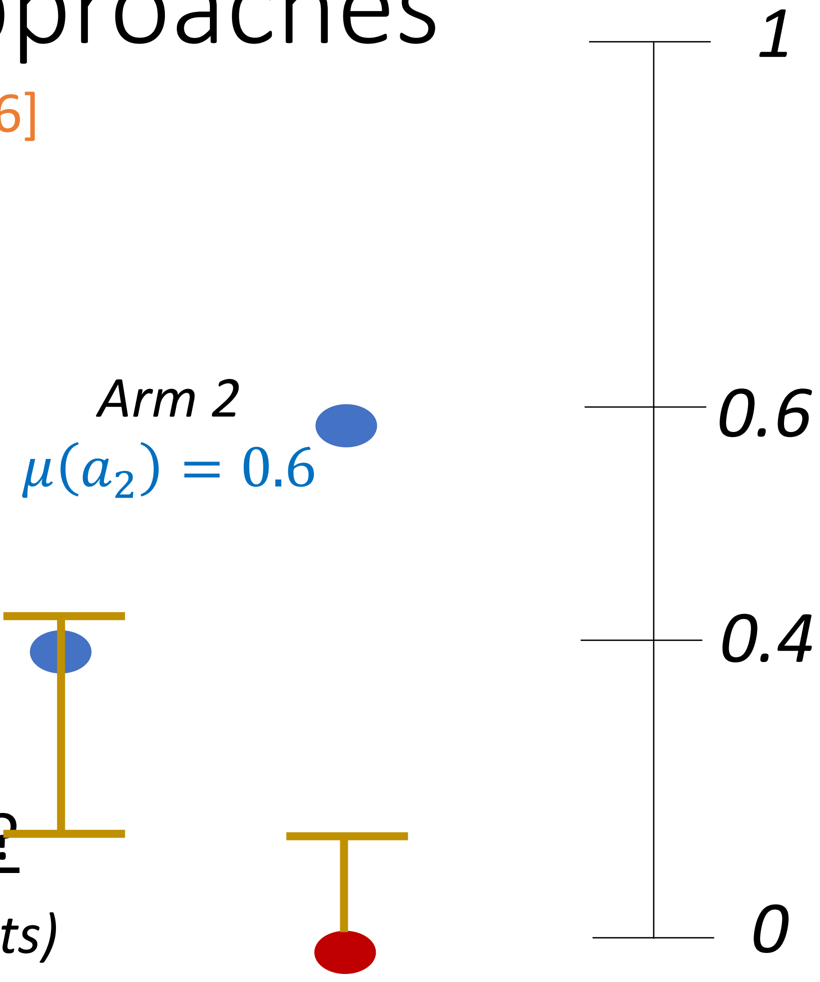
- *W.h.p. actual mean in confidence interval (concentration arguments)*

# Brittleness of stochastic approaches

<u>Successive Elimination</u> [Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another




<u>What breaks *if adversary corrupts the exploration rounds*?</u>

- *W.h.p. actual mean ~~in confidence interval~~ (concentration arguments)*

# Brittleness of stochastic approaches

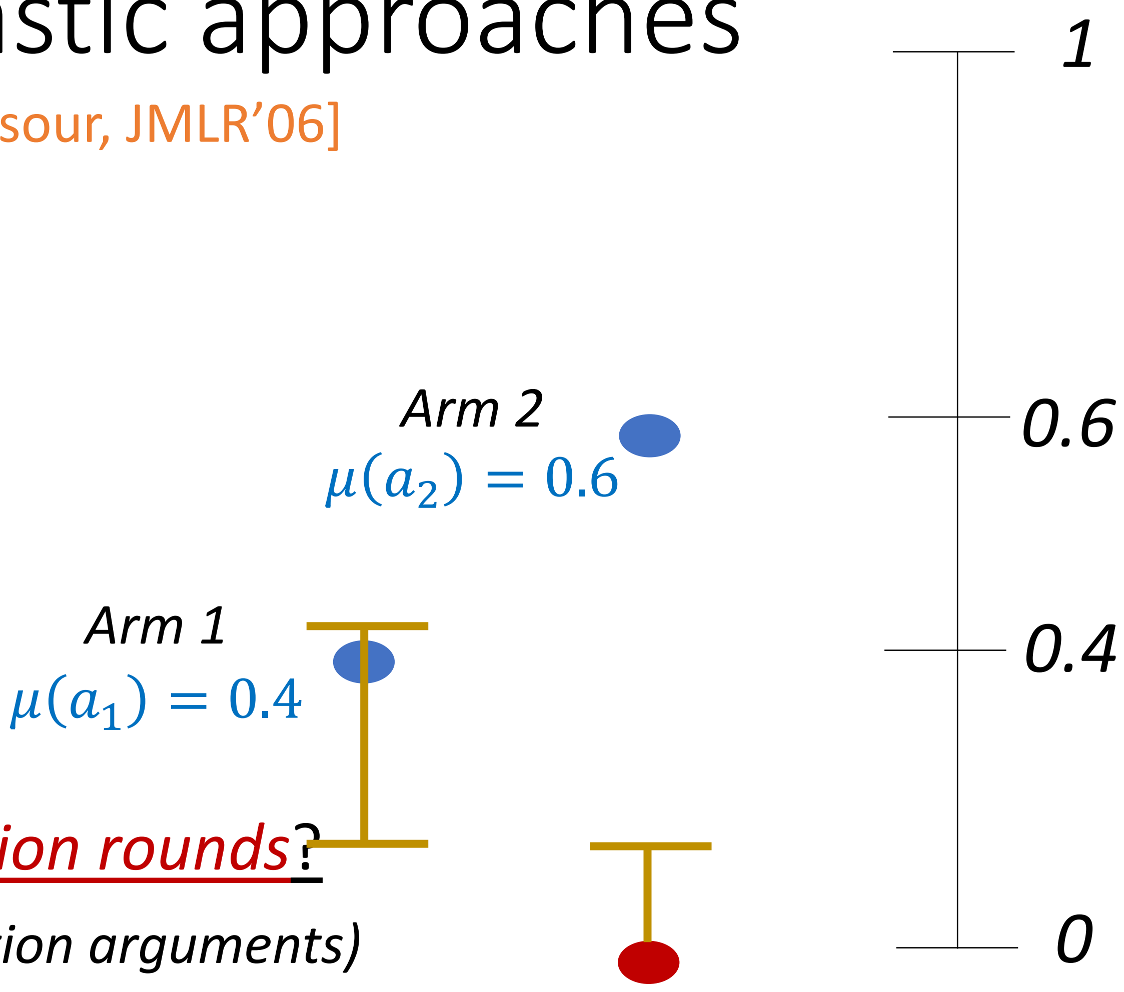
Successive Elimination [Even-Dar, Mannor, Mansour, JMLR’06]

- Each arm has a mean $\mu(a)$
- Keep a set of “active” arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
    - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an “active” arm uniformly at random
2. “Deactivate” any arm dominated by another

What breaks *if adversary corrupts the exploration rounds*?

- *W.h.p. ~~actual mean in confidence interval~~ (concentration arguments)*
- *Opimal arm $a$ is deactivated after $\log T$ rounds*

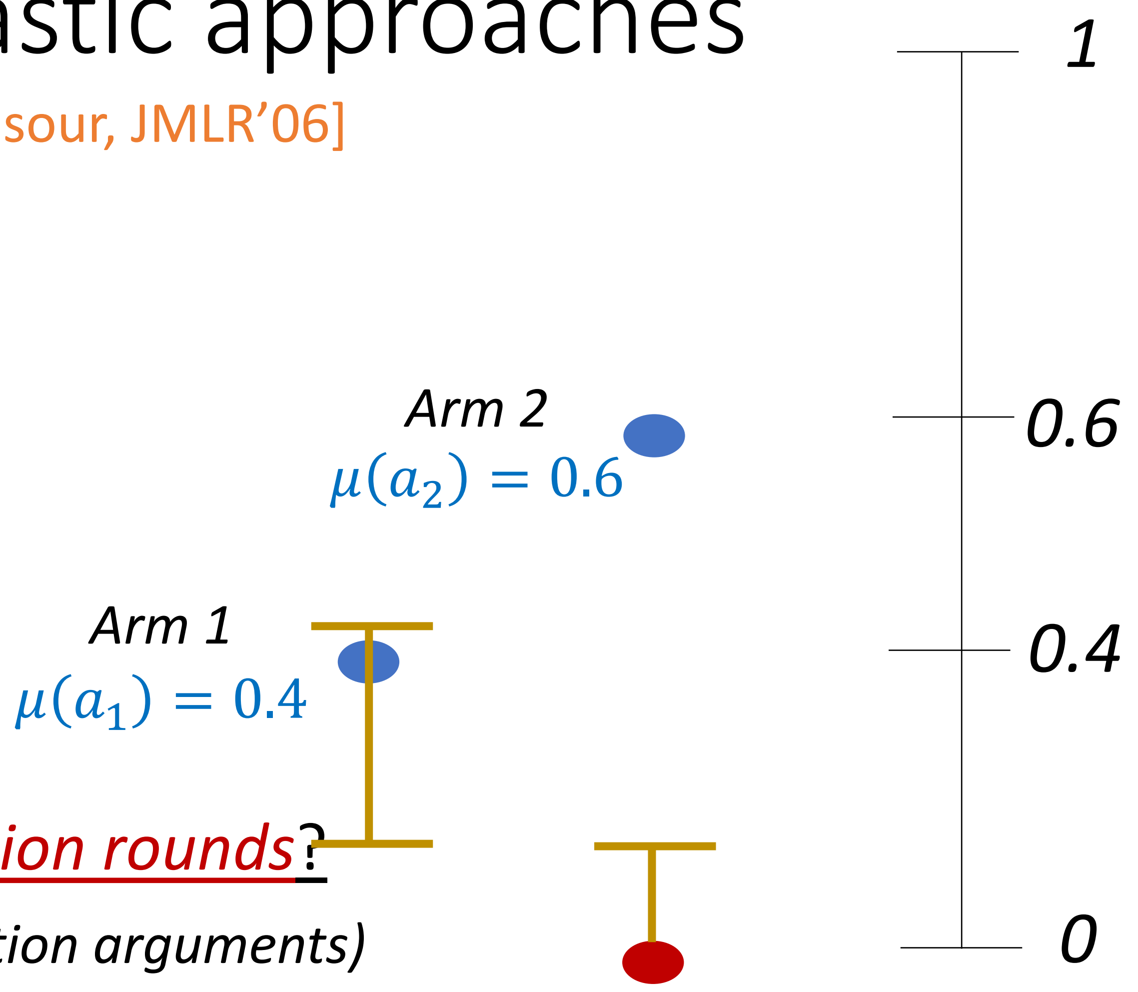


# Brittleness of stochastic approaches

<u>Successive Elimination</u> [Even-Dar, Mannor, Mansour, JMLR'06]

- Each arm has a mean $\mu(a)$
- Keep a set of "active" arms (initially all)
- Confidence interval = *Empirical mean* $\pm$ *Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}}$ *where* $N_a(t)$= *#trials*

1. Select an "active" arm uniformly at random
2. "Deactivate" any arm dominated by another




<u>What breaks *if adversary corrupts the exploration rounds*?</u>

- *W.h.p. ~~actual mean in confidence interval~~ (concentration arguments)*
- *Opimal arm $a$ is deactivated after $\log T$ rounds*
- ***Corruption then stops: linear regret with only logarithmic corruption!***

# Multi-layering Successive Elimination Race

[*L*, Mirrokni, Paes Leme, STOC'18]

If we knew that the number of corrupted rounds we encounter was $\bar{c} \leq \boldsymbol{log(KT/\delta)}$

# Multi-layering Successive Elimination Race

[L, Mirrokni, Paes Leme, STOC'18]

If we knew that the number of corrupted rounds we encounter was $\bar{c} \leq \boldsymbol{log(KT/\delta)}$

*We can account for it even if all corruption is going against us*

- Confidence interval = *Empirical mean* $\pm$ *Corruption Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}} + \frac{\bar{c}}{N_a(t)}$ *where* $N_a(t)$= *#trials*

# Multi-layering Successive Elimination Race

[*L*, Mirrokni, Paes Leme, STOC'18]

If we knew that the number of corrupted rounds we encounter was $\bar{c} \leq \boldsymbol{log(KT/\delta)}$

*We can account for it even if all corruption is going against us*

- Confidence interval = *Empirical mean* $\pm$ *Corruption Bonus*
  - *Bonus* = $\sqrt{\frac{\log(KT/\delta)}{N_a(t)}} + \frac{\bar{c}}{N_a(t)}$ *where* $N_a(t)$= *#trials*

<u>Successive Elimination analysis goes through</u>

- *W.h.p. actual mean in confidence interval*
- *Suboptimal arm* $a$ *is deactivated after* $\frac{\log(KT/\delta)+\bar{c}}{(\Delta_a)^2}$ *rounds w.h.p.*
  - *Contributes* $\frac{\log(KT/\delta)}{(\Delta_a)^2} \cdot \Delta_a = \frac{\log(KT/\delta)}{\Delta_a}$ *to regret*

# Multi-layering Successive Elimination Race

[*L*, Mirrokni, Paes Leme, STOC'18]

Idea: Create multiple independent copies of Successive Elimination (layers)

- Copy $\ell$ is responsible for corruption of $\approx 2^{\ell}$

# Multi-layering Successive Elimination Race

[*L*, Mirrokni, Paes Leme, STOC'18]

Idea: Create multiple independent copies of Successive Elimination (layers)

- Copy $\ell$ is responsible for corruption of $\approx 2^{\ell}$

***At every round: w.p. $\boldsymbol{2^{-\ell}}$ play according to copy $\boldsymbol{\ell = 1 \ldots log\, T}$***

- *Do not update estimates of any other copy*
- *Larger $\ell \geq \log C$ observe corruption at most $\bar{c} \leq$ $\boldsymbol{log(KT/\delta)}$ but slower to find $a^{\star}$*
- *Smaller $\ell$ faster but prone to corruption (similar as in Successive Elimination)*

# Multi-layering Successive Elimination Race

[*L*, Mirrokni, Paes Leme, STOC'18]

Idea: Create multiple independent copies of Successive Elimination (layers)

- Copy $\ell$ is responsible for corruption of $\approx 2^{\ell}$

***At every round: w.p. $2^{-\ell}$ play according to copy $\ell = 1 \ldots \log T$***

- *Do not update estimates of any other copy*
- *Larger $\ell \geq \log C$ observe corruption at most $\bar{c} \leq \log(KT/\delta)$ but slower to find $a^{\star}$*
- *Smaller $\ell$ faster but prone to corruption (similar as in Successive Elimination)*

**Challenge:** achieve a ***race across copies*** that combines learning speed with robustness

**Idea:** robust copies supervise faster ones *(nested eliminations of active arms)*

- *Number of rounds that a suboptimal arm survives: dictated by fastest robust copy $\ell^{\star} = \lceil \log C \rceil$*

***Regret of non−robust copies $\leq C \cdot$ Regret of fastest robust copy $\ell^{\star}$***

# Recipe for corruptions in multi-armed bandits

[*L*, Mirrokni, Paes Leme, STOC'18]

**Require:**

- Problem that can be solved by estimating **"ground truth"**

  $a^\star$ in multi-armed bandits

- An algorithm ***ALG*** that aggressively refines active confidence set containing "ground truth"

  ***ALG=Successive Elimination*** [Even-Dar, Mannor, Mansour, JMLR'06]

**Steps:**

1. *Robustness to **known amount** of corruption* $\bar{c} \approx \log T$ : ***ALG*** $\Rightarrow$ ***ROBUSTALG***$(\bar{\boldsymbol{c}})$
2. *Adapting to **unknown amount** of corruption* $C$:
   - *Run independent copies of **ROBUSTALG***$(\boldsymbol{log\ T})$ *in parallel*
   - *Each copy responsible for a different level of corruption*
   - *Robust versions supervise non-robust & correct errors via nested eliminations*

# Recipe for corruptions in contextual pricing

[Krishnamurthy, *L*, Podimata, Schapire, STOC'21 / OR'22]

**Require:**

- Problem that can be solved by estimating **"ground truth"**

  $\theta^\star$ in contextual pricing ---> value of customer is $\langle \theta^\star, x_t \rangle$ for adversarial context $x_t$

- An algorithm ***ALG*** that aggressively refines active confidence set containing "ground truth"

  ***ALG=Projected Volume*** [Lobel, Paes Leme, Vladu, EC'17 / OR'18]

**Steps:**

1. *Robustness to **known amount** of corruption $\bar{c} \approx \log T$ :* ***ALG*** $\Rightarrow$ ***ROBUSTALG***$(\bar{c})$

2. *Adapting to **unknown amount** of corruption $C$*

# Recipe for corruptions in contextual pricing

[Krishnamurthy, *L*, Podimata, Schapire, STOC'21 / OR'22]

**Require:**

- Problem that can be solved by estimating **"ground truth"**

  $\theta^\star$ in contextual pricing ---> value of customer is $\langle\theta^\star, x_t\rangle$ for adversarial context $x_t$

- An algorithm ***ALG*** that aggressively refines active confidence set containing "ground truth"

  ***ALG=Projected Volume*** [Lobel, Paes Leme, Vladu, EC'17 / OR'18]

**Steps:**

1. *Robustness to **known amount** of corruption* $\bar{c} \approx \log T$ : ***ALG*** $\Rightarrow$ ***ROBUSTALG***$(\bar{c})$
2. *Adapting to **unknown amount** of corruption* $C$

***Chara Podimata will present this result in the September workshop***

# Multi-layering race: a general recipe for corruptions

**Require:**

- Problem that can be solved by estimating **"ground truth"**

  $a^\star$ in multi-armed bandits $\theta^\star$ in contextual pricing

- An algorithm ***ALG*** that aggressively refines active confidence set containing "ground truth"

  ***ALG=Successive Elimination*** ***ALG=Projected Volume***

**Steps:**

1. *Robustness to **known amount** of corruption* $\bar{c} \approx \log T$ : ***ALG*** $\Rightarrow$ ***ROBUSTALG***$(\bar{c})$
2. *Adapting to **unknown amount** of corruption* $C$:
   - *Run independent copies of **ROBUSTALG***$(\log T)$ *in parallel*
   - *Each copy responsible for a different level of corruption*
   - *Robust versions supervise non-robust & correct errors via nested eliminations*

# Multi-layering race: a general recipe for corruptions

**Require:**

- Problem that can be solved by estimating **"ground truth"**

  $a^\star$ in multi-armed bandits $\theta^\star$ in contextual pricing

- An algorithm ***ALG*** that aggressively refines active confidence set containing "ground truth"

  ***ALG=Successive Elimination*** ***ALG=Projected Volume***

**Steps:**

1. *Robustness to **known amount** of corruption* $\bar{c} \approx \log T$ : ***ALG*** $\Rightarrow$ ***ROBUSTALG***$(\bar{\boldsymbol{c}})$
2. *Adapting to **unknown amount** of corruption* $C$:

**Other results via this recipe**

*Assortment optimization* [Chen, Krishnamurty, Wang'19] *via* [Agrawal, Avandhanula, Goyal, Zeevi, OR'19]

*Product rankings* [Golrezaei, Manshadi, Schneider, Sekar, EC'21] *via* [Derakhshan, Golrezaei, Manshadi, Mirrokni EC'20/MS'21]

# BARBAR

[Gupta, Koren, Talwar, COLT'19]

Multi-layering Successive Elimination Race runs copies in parallel

Robustness as slower copies are not selected too often: corruption subsampled

# BARBAR

[Gupta, Koren, Talwar, COLT'19]

Multi-layering Successive Elimination Race runs copies in parallel

Robustness as slower copies are not selected too often: corruption subsampled

Bandit Algorithms with Robustness: Bad Arms get Recourse (BARBAR)

- Works in geometrically increasing epochs: decisions always determined by previous epoch
- If input was stochastic, learn all arms with gap $2^{-\ell}$ by epoch $\ell$

# BARBAR

[Gupta, Koren, Talwar, COLT'19]

<u>Multi-layering Successive Elimination Race runs copies in parallel</u>

Robustness as slower copies are not selected too often: corruption subsampled

<u>Bandit Algorithms with Robustness: Bad Arms get Recourse (BARBAR)</u>

- Works in geometrically increasing epochs: decisions always determined by previous epoch
- If input was stochastic, learn all arms with gap $2^{-\ell}$ by epoch $\ell$
- Instead of eliminating "suboptimal" arms, BARBAR *selects them w.p. inverse to empirical gap*
- If $a^\star$ seems "bad" in an epoch, adversary needs much budget to corrupt it again
  - corruption subsampled automatically for any "bad arm"

# Tsallis-INF

[Zimmert & Seldin, JMLR’21]

- Analysis upper bounds regret via a unified ”self-bounding term”
- Optimal stochastic and adversarial pseudoregret guarantees
- Same analysis extends for pseudoregret in adversarial corruptions
- Dependence slightly strengthened subsequently [Massoudian & Seldin, COLT’21] [Ito, NeurIPS’21]

## Building block for regularizers that extend beyond multi-armed bandits

- combinatorial semi-bandits (routing) [Zimmert, Luo, Wei, ICML’19]
- reinforcement learning with unknown i.i.d. transitions [Jin, Huang, Luo, NeurIPS’21]

# Comparison of these techniques

Multi-layering successive elimination race [*L*, Mirrokni, Paes Leme, STOC'18]

+ applies to any setting with "confidence set" (binary feedback, no adversarial counterparts, etc)

+ high-probability guarantees

- multiplicative dependence on number of corrupted rounds $C$

# Comparison of these techniques

Multi-layering successive elimination race [*L*, Mirrokni, Paes Leme, STOC'18]

+ applies to any setting with "confidence set" (binary feedback, no adversarial counterparts, etc)

+ high-probability guarantees

- multiplicative dependence on number of corrupted rounds $C$

BARBAR [Gupta, Koren, Talwar, COLT'19]

+ elegant corruption subsampling => additive dependence on corrupted rounds $C$

+ high-probability guarantees

- requires some notion of "gap" to apply: less broadly applicable

# Comparison of these techniques

Multi-layering successive elimination race [*L*, Mirrokni, Paes Leme, STOC'18]

+ applies to any setting with "confidence set" (binary feedback, no adversarial counterparts, etc)

+ high-probability guarantees

- multiplicative dependence on number of corrupted rounds $C$

BARBAR [Gupta, Koren, Talwar, COLT'19]

+ elegant corruption subsampling => additive dependence on corrupted rounds $C$

+ high-probability guarantees

- requires some notion of "gap" to apply: less broadly applicable

Tsallis-INF [Zimmert & Seldin, JMLR'21]

+ achieves interpolation between two extremes

- requires some way to do IW: unclear how to go beyond bandit feedback & finite # policies

# Application to episodic RL

Building on multi-layering race [*L*, Simchowitz, Slivkins, Sun, COLT'21]

*+ applies to all settings with uncorrupted guarantees (tabular MDP, linear MDP, gap-based results)*

*- Multiplicative dependence on number of corrupted rounds $C$*

# Application to episodic RL

Building on multi-layering race [*L*, Simchowitz, Slivkins, Sun, COLT'21]

*+ applies to all settings with uncorrupted guarantees (tabular MDP, linear MDP, gap-based results)*

*- Multiplicative dependence on number of corrupted rounds $C$*

Building on BARBAR [Chen, Du, Jamieson, ICML'21]

*+ Additive dependence on number of corrupted rounds*

*- only applies to tabular MDP and gap-independent results*

# Application to episodic RL

## Building on multi-layering race [*L*, Simchowitz, Slivkins, Sun, COLT'21]

*+ applies to all settings with uncorrupted guarantees (tabular MDP, linear MDP, gap-based results)*

*- Multiplicative dependence on number of corrupted rounds $C$*

## Building on BARBAR [Chen, Du, Jamieson, ICML'21]

*+ Additive dependence on number of corrupted rounds*

*- only applies to tabular MDP and gap-independent results*

## Building on Tsallis-INF [Jin, Huang, Luo, NeurIPS'21]

*+ interpolation between the two extremes*

*- Requires transitions to not be corrupted => not clear how to do IW otherwise*

# Symbiosis of these techniques

[Chen & Wang, OR'22]

Recent work on learning and pricing with inventory constraints

- Binary search to identify right inventory level
- Multi-armed bandits to decide the most profitable price (arm)

# Symbiosis of these techniques

[Chen & Wang, OR'22]

Recent work on learning and pricing with inventory constraints

- Binary search to identify right inventory level
- Multi-armed bandits to decide the most profitable price (arm)

Need for symbiosis

- Tsallis-INF cannot work with binary feedback for the first task
- Multi-layering successive elimination race: suboptimal regret for the second task

# Symbiosis of these techniques

[Chen & Wang, OR'22]

Recent work on learning and pricing with inventory constraints

- Binary search to identify right inventory level
- Multi-armed bandits to decide the most profitable price (arm)

Need for symbiosis

- Tsallis-INF cannot work with binary feedback for the first task
- Multi-layering successive elimination race: suboptimal regret for the second task

**Algorithm combines the two techniques & achieves near-optimal regret**

# Model selection lens

Model selection: One way to view adversarial corruptions

- *Different layers in multi-layering race can be viewed as different models*

Recent work makes this connection for corrupted RL [Wei, Dann, Zimmert, ALT'22]

- Builds on model selection approach for non-stationary RL [Wei & Luo, COLT'21]

# Model selection lens

Model selection: One way to view adversarial corruptions

- *Different layers in multi-layering race can be viewed as different models*

Recent work makes this connection for corrupted RL [Wei, Dann, Zimmert, ALT'22]

- Builds on model selection approach for non-stationary RL [Wei & Luo, COLT'21]

***Chen-Yu Wei will present this line of work in the September workshop***

# Model selection lens

Model selection: One way to view adversarial corruptions

- *Different layers in multi-layering race can be viewed as different models*

Recent work makes this connection for corrupted RL [Wei, Dann, Zimmert, ALT'22]

- Builds on model selection approach for non-stationary RL [Wei & Luo, COLT'21]

***Chen-Yu Wei will present this line of work in the September workshop***

Another stochastic-adversarial interpolation via model selection

- *Memory of the adversary:* $r_a(t) \sim F_a(H_{t-M \ldots t-1})$
- *Some results for full information* [Muthukumar, Ray, Sahai, Bartlett, AISTATS'21]

# Model selection lens

Model selection: One way to view adversarial corruptions

- *Different layers in multi-layering race can be viewed as different models*

Recent work makes this connection for corrupted RL [Wei, Dann, Zimmert, ALT'22]

- Builds on model selection approach for non-stationary RL [Wei & Luo, COLT'21]

***Chen-Yu Wei will present this line of work in the September workshop***

Another stochastic-adversarial interpolation via model selection

- *Memory of the adversary:* $r_a(t) \sim F_a(H_{t-M\ldots t-1})$
- *Some results for full information* [Muthukumar, Ray, Sahai, Bartlett, AISTATS'21]

***Vidya Muthukumar will present this line of work in the September workshop***

# Agent-based learning

Stochastic model can often be thought as best response for an agent

- Pricing example: agent buys if value $\geq$ price

# Agent-based learning

Stochastic model can often be thought as best response for an agent

- Pricing example: agent buys if value ≥ price

Principal-agent or Stackelberg games capture this paradigm

- ***Principal commits on a (randomized) action*** $\boldsymbol{x_t}$
- ***Agent best responds according to their payoff matrix***

# Agent-based learning

Stochastic model can often be thought as best response for an agent

- Pricing example: agent buys if value ≥ price

Principal-agent or Stackelberg games capture this paradigm

- ***Principal commits on a (randomized) action $x_t$***
- ***Agent best responds according to their payoff matrix***

Learning in Stackelberg games: Principal does not know agent's payoff matrix

- Stackelberg Security Games [Blum, Haghtalab, Procaccia, NeurIPS'14] [Peng, Shen, Tang, Zuo, AAAI'19]
- Pricing with an unknown demand curve [Kleinberg & Leighton, FOCS'03] [Besbes & Zeevi, OR'09]
- Strategic classification [Dong, Roth, Schutzman, Waggoner, Wu, EC'18] [Chen, Liu, Podimata, NeurIPS'20]

# Agent-based learning

Stochastic model can often be thought as best response for an agent

- Pricing example: agent buys if value ≥ price

Principal-agent or Stackelberg games capture this paradigm

- ***Principal commits on a (randomized) action $x_t$***
- ***Agent best responds according to their payoff matrix***

Learning in Stackelberg games: Principal does not know agent's payoff matrix

- Stackelberg Security Games [Blum, Haghtalab, Procaccia, NeurIPS'14] [Peng, Shen, Tang, Zuo, AAAI'19]
- Pricing with an unknown demand curve [Kleinberg & Leighton, FOCS'03] [Besbes & Zeevi, OR'09]
- Strategic classification [Dong, Roth, Schutzman, Waggoner, Wu, EC'18] [Chen, Liu, Podimata, NeurIPS'20]

Crucial limitation of stochastic model: Agent is completely myopic (thus best responds)

- *Agent may want to sacrifice present payoff to affect principal's learning & get future utility*

# Learning with non-myopic agents

[Haghtalab, *L*, Nietert, Wei, EC'22]

Typical model for non-myopia: Agent is discounting the future

- At round $\tau$, agent selects action $y_\tau$ that (approx.) maximizes $\sum_{t \geq \tau} \gamma^{t-\tau} E[v_t(x_t, y_t)]$
- Interpolation between stochastic (best response) and adversarial (infinitely patient)

# Learning with non-myopic agents

[Haghtalab, *L*, Nietert, Wei, EC'22]

Typical model for non-myopia: Agent is discounting the future

- At round $\tau$, agent selects action $y_\tau$ that (approx.) maximizes $\sum_{t \geq \tau} \gamma^{t-\tau} E[v_t(x_t, y_t)]$
- Interpolation between stochastic (best response) and adversarial (infinitely patient)

Our approach:

- **Establish an information screen: slows down reacting to agent's responses**
  - *Delaying reaction decreases incentive for large deviations from best response*

# Learning with non-myopic agents

[Haghtalab, *L*, Nietert, Wei, EC'22]

Typical model for non-myopia: Agent is discounting the future

- At round $\tau$, agent selects action $y_\tau$ that (approx.) maximizes $\sum_{t \geq \tau} \gamma^{t-\tau} E[v_t(x_t, y_t)]$
- Interpolation between stochastic (best response) and adversarial (infinitely patient)

Our approach:

- **Establish an information screen: slows down reacting to agent's responses**
  - *Delaying reaction decreases incentive for large deviations from best response*
- Design minimally reactive algorithms that are robust to approximate best responses
  - *On the way, optimal algorithm for learning in Stackelberg Security Games with myopic agents*

# Learning with non-myopic agents

[Haghtalab, *L*, Nietert, Wei, EC'22]

Typical model for non-myopia: Agent is discounting the future

- At round $\tau$, agent selects action $y_\tau$ that (approx.) maximizes $\sum_{t\geq\tau} \gamma^{t-\tau} E[v_t(x_t, y_t)]$
- Interpolation between stochastic (best response) and adversarial (infinitely patient)

Our approach:

- **Establish an information screen: slows down reacting to agent's responses**
  - *Delaying reaction decreases incentive for large deviations from best response*
- Design minimally reactive algorithms that are robust to approximate best responses
  - *On the way, optimal algorithm for learning in Stackelberg Security Games with myopic agents*
- Apply the multi-layering race recipe to adapt to unknown discount factor of agent

# Learning with non-myopic agents

[Haghtalab, *L*, Nietert, Wei, EC'22]

Typical model for non-myopia: Agent is discounting the future

- At round $\tau$, agent selects action $y_\tau$ that (approx.) maximizes $\sum_{t \geq \tau} \gamma^{t-\tau} E[v_t(x_t, y_t)]$
- Interpolation between stochastic (best response) and adversarial (infinitely patient)

Our approach:

- **Establish an information screen: slows down reacting to agent's responses**
  - *Delaying reaction decreases incentive for large deviations from best response*
- Design minimally reactive algorithms that are robust to approximate best responses
  - *On the way, optimal algorithm for learning in Stackelberg Security Games with myopic agents*
- Apply the multi-layering race recipe to adapt to unknown discount factor of agent

***Sloan Nietert will likely present a poster on this work in the September workshop***

# Summary

*<u>Q1 (Best of both worlds)</u>*

*<u>Q2 (Bridging the two worlds)</u>*

*<u>Q3 (Beyond multi-armed bandits)</u>*

# Summary

*Q1 (Best of both worlds)*

- *Stochastic-based: Run stochastic,* test, *switch to adversarial if test fails*
- *Adversarial-based: Run adversarial, adapt exploration to empirical gap*

*Q2 (Bridging the two worlds)*

*Q3 (Beyond multi-armed bandits)*

# Summary



*Q1 (Best of both worlds)*

- *Stochastic-based: Run stochastic,* test, *switch to adversarial if test fails*
- *Adversarial-based: Run adversarial, adapt exploration to empirical gap*

*Q2 (Bridging the two worlds)*

- *Number of* ***adversarial corruptions***, ***memory of adversary***, ***discount factor of non-myopic agent***
- *For adversarial corruptions: Multi-layering race, BARBAR, Tsallis-INF*

*Q3 (Beyond multi-armed bandits)*

# Summary



***Q1 (Best of both worlds)***

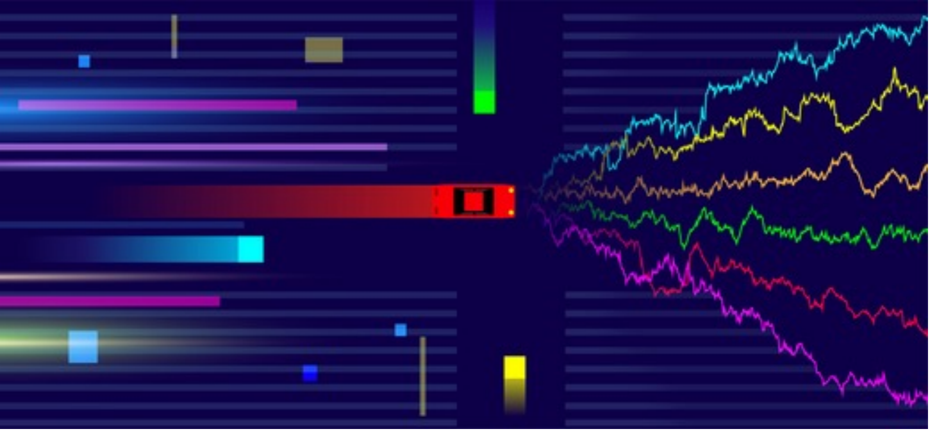
- *Stochastic-based: Run stochastic, test, switch to adversarial if test fails*
- *Adversarial-based: Run adversarial, adapt exploration to empirical gap*

***Q2 (Bridging the two worlds)***

- *Number of **adversarial corruptions**, **memory of adversary**, **discount factor of non-myopic agent***
- *For adversarial corruptions: Multi-layering race, BARBAR, Tsallis-INF*

***Q3 (Beyond multi-armed bandits)***

- *General recipe for multi-layering race (e.g., contextual pricing, non-myopic learning)*
- *Tsallis-INF extendable in settings where one can do Importance Weighted Sampling*
- *Sometimes symbiosis is useful*



# Thank you!

## Summary

Q1 (Best of both worlds)

- Stochastic-based: Run stochastic, test, switch to adversarial if test fails
- Adversarial-based: Run adversarial, adapt exploration to empirical gap

Q2 (Bridging the two worlds)

- Number of adversarial corruptions, memory of adversary, discount factor of non-myopic agent
- For adversarial corruptions: Multi-layering race, BARBAR, Tsallis-INF

Q3 (Beyond multi-armed bandits)

- General recipe for multi-layering race (e.g., contextual pricing, non-myopic learning)
- Tsallis-INF extendable in settings where one can do Importance Weighted Sampling
- Sometimes symbiosis is useful